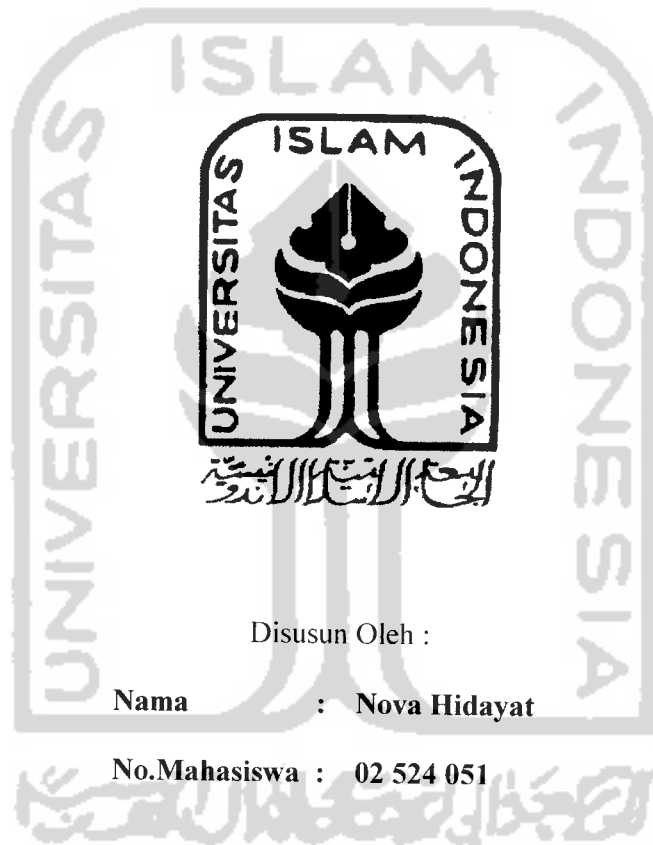


**PERANCANGAN
ENKRIPTOR – SIMETRIK SINYAL AUDIO
BERBASIS MIKROKONTROLER ATMEGA16**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Pada Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia**



Disusun Oleh :

Nama : Nova Hidayat

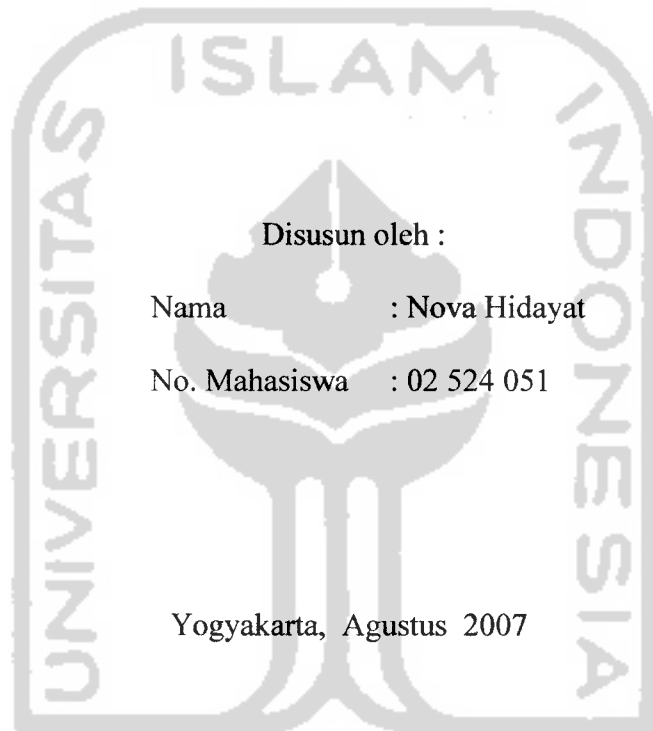
No.Mahasiswa : 02 524 051

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007**

LEMBAR PENGESAHAN PEMBIMBING

**PERANCANGAN ENKRIPTOR - SIMETRIK SINYAL AUDIO
BERBASIS MIKROKONTROLER ATMEGA16**

TUGAS AKHIR



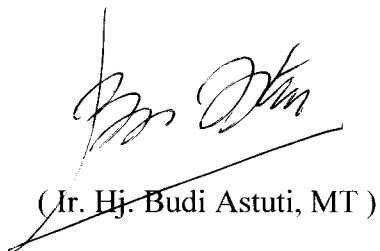
Disusun oleh :

Nama : Nova Hidayat

No. Mahasiswa : 02 524 051

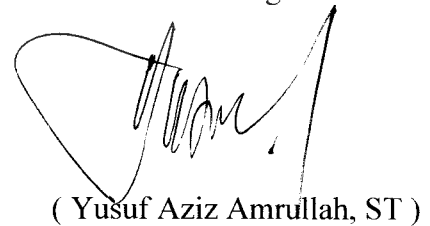
Yogyakarta, Agustus 2007

Pembimbing I

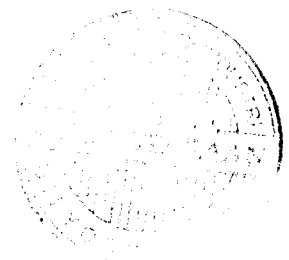


(Ir. Hj. Budi Astuti, MT)

Pembimbing II



(Yusuf Aziz Amrullah, ST)



LEMBAR PENGESAHAN PENGUJI

PERANCANGAN ENKRIPTOR-SIMETRIK SINYAL AUDIO BERBASIS MIKROKONTROLER ATMEGA16

Disusun oleh :

Nama : Nova Hidayat

No. Mahasiswa : 02 524 051

**Telah Dipertahankan di Depan Sidang Penguji sebagai Salah satu Syarat untuk
Memperoleh Gelar Sarjana Teknik Elektro Fakultas Teknologi Industri
Universitas Islam Indonesia**

Yogyakarta, 15 Agustus 2007

Tim Penguji,

Ir.Hj. Budi Astuti, MT

Ketua

Yusuf Aziz Amrullah, S.T

Anggota I

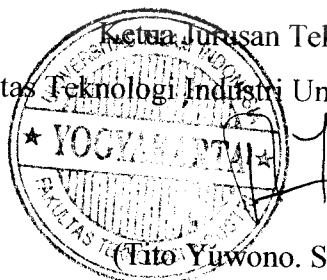
Dwi Ana Ratna Wati, S.T

Anggota II

Mengetahui,

Ketua Jurusan Teknik Elektro

Fakultas Teknologi Industri Universitas Islam Indonesia



(Tito Yuwono. S.T., M.Sc)

HALAMAN PERSEMBAHAN

Alhamdulillah atas segala nikmat & karunia dari Nya dan syukurku atas terselesaikannya karya ini, kupersembahkan dengan segala kerendahan hati kepadaMu ya Rabb, Penguasa Bumi dan Langit

Agamaku "Islam " yang memberikan ketenangan dalam menjalani kehidupan ini.

Ayahanda Achmad Basuni dan Ibuanda Dian Kartini "tercinta", "Terima kasih selalu memberikan yang terbaik untuk saya, dan tak henti-hentinya melimpahkan kasih sayang dan doanya".

Kakak2 dan adik2 ku "tersayang" Rahmi, Lea Ifan, De Selvi, De Rizki dan ponakanku Indra. "Eratkan tali persaudaraan untuk menjadi keluarga yang satu dan bahagia membahagiakan orang tua kita"

Teman-teman Asrama Mahasiswa Perantauan Selatan "Pangeran Sidiyotulahi" yang telah membantuku berprestasi dan duka. "Terima kasih sampai putus tali persahabatan kita"

Acil Rusli yang seperti orang tua semua, selalu memberikan arahan, nasehat dan selalu mendermikan urusannya saya.....

Teman-teman elektio angkatan 2009, teman2 Mancing Band CII, Teman2 KKA, Teman2 perantau dari kalsel, teman2 keluarga serta semua temanku yang tdk dapat disebutkan satu-persatu semoga menjadi kenangan.....

Seluruh keluarga besarku.....

"Terima Kasih atas segala dukungan dan doanya".

Guru dan Dosenku, " Semua yang kudapatkan sekarang karena kalian "

" Semua yang ada padaku sekarang karena pengabdian kalian".....

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji dan syukur ditunjukkan hanya bagi Allah SWT, sang Esa pemilik alam semesta. Semoga kesejahteraan diberikan bagi rasul-Nya, Muhammad SAW, sang mustofa. Atas rahmat dan taufik-Nya penulis dapat menyelesaikan tugas akhir dengan judul “ **Perancangan Enkriptor - Simetrik Sinyal Audio Berbasis Mikrokontroler ATmega16** ” dengan lancar.

Adapun maksud dan tujuan penyusunan tugas akhir ini adalah untuk melengkapi salah satu syarat dalam menempuh gelar sarjana pada Jurusan Teknik Elektro Universitas Islam Indonesia, Yogyakarta. Disamping itu untuk menambah pengetahuan terhadap ilmu yang telah dipelajari di bangku perkuliahan untuk diterapkan di masyarakat.

Pada kesempatan ini dengan segala rasa syukur dan kerendahan hati penulis sampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Bpk Fathul Wahid, ST. M.Sc. Selaku Dekan Fakultas Teknologi Industri (FTI) Universitas Islam Indonesia (UII).
2. Bpk Tito Yuwono, ST . MT selaku Kepala Jurusan Teknik Elektro.
3. Ibu Ir. Hj. Budi Astuti, MT. Selaku dosen pembimbing I atas masukan dan bimbingannya.

ABSTRAK

Pada garis besarnya, masalah keamanan jaringan dapat dibagi menjadi empat bidang yang saling berhubungan: kerahasiaan, keaslian, pengakuan dan kontrol integritas. Untuk mengantisipasi berbagai masalah kebocoran informasi rahasia, maka diperlukan sebuah sistem yang real time, untuk mengamankan informasi yang dikirimkan tersebut. Enkripsi data audio merupakan suatu perancangan pengamanan data audio yang banyak dirancang untuk tujuan penelitian, industri maupun kompetensi. Sesuai dengan namanya, tugas dari enkripsi data audio adalah mampu mengamankan data audio dengan proses mengacak data audio sebelum dikirim dengan tingkat presisi tertentu. Hal ini akan sangat bermanfaat digunakan untuk melakukan pengamanan data audio dengan tingkat keamanan dan kerahasiaan yang tinggi tanpa dibatasi ruang dan waktu. Pengolahnya berupa mikrokontroler ATmega16 dan frekuensi yang diolah antara 20 Hz sampai 20 KHz. Proses enkripsi diawali dengan pengolahan sinyal yang masuk melalui internal ADC menjadi sinyal audio digital. Dalam pengacakan ini menggunakan metode aritmatika berupa gerbang XOR yaitu untuk mode 1, data masuk di XOR dengan data sebelumnya dan untuk mode 2, data masuk di XOR dengan data 2 kali sebelumnya. Hasil pengolahan dari sistem tersebut akan menghasilkan chipper digital audio, yaitu sinyal audio rahasia, yang telah diamankan, sehingga orang lain yang “menyadap” sinyal tersebut pada frekuensi transmitter yang sama dan mendengarkannya, maka tak ada informasi yang dapat mereka peroleh, kecuali mereka mampu memecahkan kode rahasianya dan menggunakan *descriptor*. Keluaran dari perancangan ini ada dua keadaan yaitu tanpa terenkripsi atau normal dan terenkripsi dengan dua mode, yaitu mode 1 dan mode 2, yang masing-masing sinyal audio yang di-enskripsi hasilnya berubah dan tidak seperti sinyal audio aslinya, baik berupa audionya, bentuk sinyalnya dan memiliki tingkat keamanan dan variasi enkripsi yang berbeda.

Kata kunci adalah kriptografi, chipper, Analog Digital Converter, titik sampling.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING	ii
HALAMAN PENGESAHAN PENGUJI	iii
HALAMAN PERSEMBAHAN.....	iv
MOTTO	v
KATA PENGANTAR	vi
ABSTRAK	viii
DAFTAR ISI.....	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I. PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian	3
1.5. Sistematika Penulisan.....	3
BAB II. LANDASAN TEORI	
2.1. Enkripsi Data	5
2.1.1. Standarisasi Enkripsi	6
2.1.2. Model-model enkripsi.....	7

2.1.3. Manfaat dan kerugian menggunakan enkripsi	9
2.2. Mikrokontroler ATmega16	
2.2.1. Gambaran umum	10
2.2.2. Arsitektur ATmega16.....	11
2.2.3. Fitur ATmega16.....	12
2.2.4. Konfigurasi pin ATmega16	13
2.2.5. Status register	15
2.2.6. Interupsi	17
2.2.7. ADC (Perubah analog ke digital)	20
2.3. Gerbang logika	24
2.4. Konsep dasar audio	28
2.4.1. Suara	28
2.4.1.1. Frekuensi	29
2.4.1.2. Amplitudo.....	30
2.4.1.3. Velocity	30
2.4.2. Representasi suara.....	31
2.4.3. Teori sampling.....	31

BAB III. PERANCANGAN SISTEM

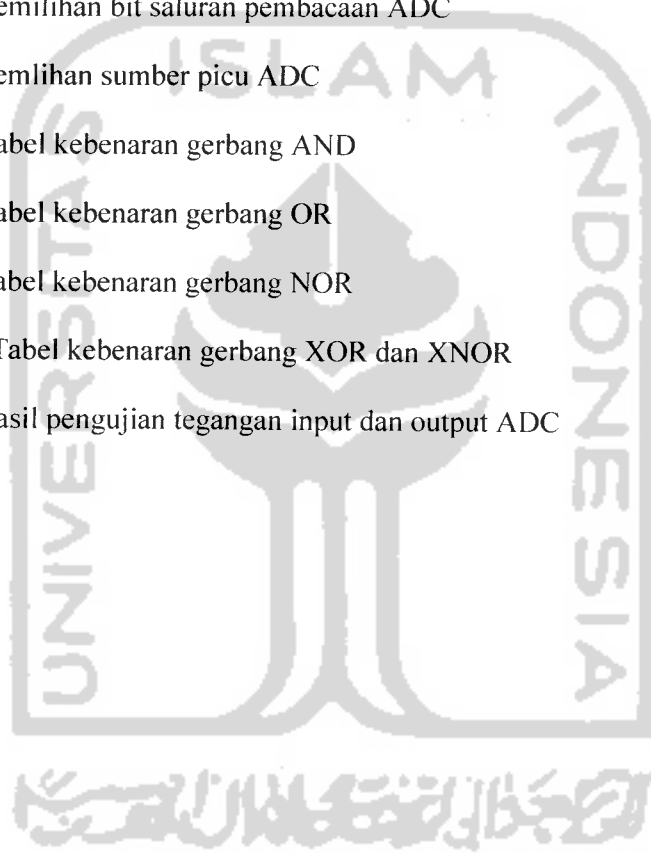
3.1. Perancangan perangkat keras.....	37
3.1.1. Rangkaian pengoperasian mikrokontroler	37
3.1.2. Perancangan power supply.....	42
3.2. Perancangan perangkat lunak	43

BAB IV. ANALISA DAN PEMBAHASAN

4.1. Pengujian rangkaian mikrokontroler.....	49
4.1.1. Pengujian tegangan input dan output ADC.....	50
4.1.2 Pengujian metode enkripsi.....	51
4.2. Pengujian power suplai	51
4.3. Pengujian sistem keseluruhan.....	52
4.3.1. Pengujian sinyal.....	52
4.3.2. Pengujian sinyal normal dan enkripsi.....	53
4.3.2.1. Pengujian input sinyal normal dengan normal	53
4.3.2.2. Pengujian input sinyal normal dengan mode 1.....	54
4.3.2.3. Pengujian input sinyal normal dengan mode 2.....	56
4.3.2.4. Pengujian input sinyal audio dengan mode.....	67
BAB V. PENUTUP	
5.1. Kesimpulan	70
5.2. Saran.....	71
DAFTAR PUSTAKA.....	72
LAMPIRAN	

DAFTAR TABEL

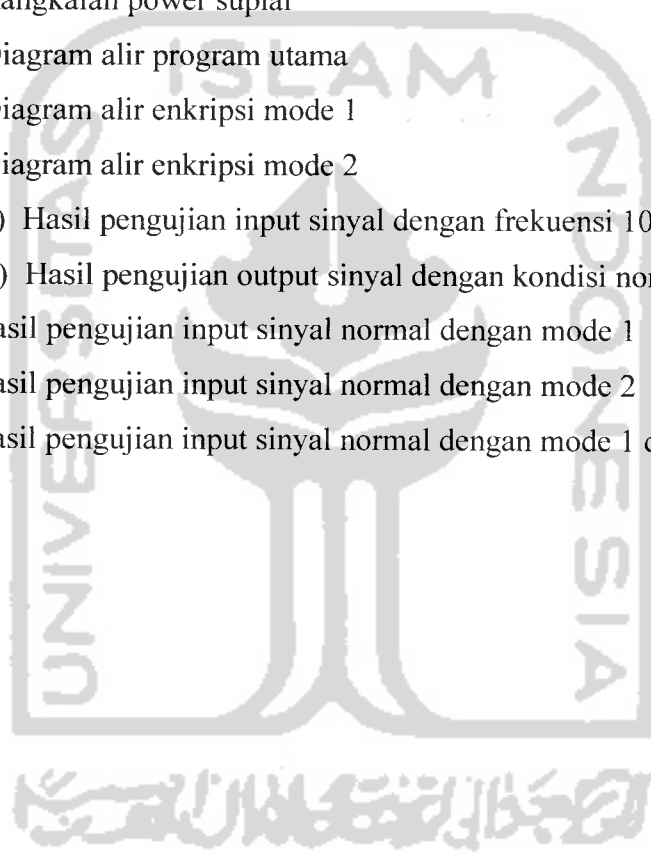
Tabel 2.1. Setting kondisi menyebabkan interupsi eksternal 1	17
Tabel 2.2. Setting kondisi menyebabkan interupsi eksternal 0	18
Tabel 2.3. Macam sumber interupsi pada ATmega16	19
Tabel 2.4. Pemilihan mode tegangan referensi ADC	21
Tabel 2.5. Pemilihan bit saluran pembacaan ADC	22
Tabel 2.6. Pemilihan sumber picu ADC	24
Tabel 2.7. Tabel kebenaran gerbang AND	25
Tabel 2.8. Tabel kebenaran gerbang OR	26
Tabel 2.9. Tabel kebenaran gerbang NOR	27
Tabel 2.10. Tabel kebenaran gerbang XOR dan XNOR	28
Tabel 4.1. Hasil pengujian tegangan input dan output ADC	50



DAFTAR GAMBAR

Gambar 2.1	Enkripsi data	6
Gambar 2.2	AES (Advance Encrytion Standart)	7
Gambar 2.3.	Enkripsi simetrik	8
Gambar 2.4.	Enkripsi Asimetrik	9
Gambar 2.5	Blok diagram Fungsional ATmega16	11
Gambar 2.6.	Konfigurasi pin mikrokontroler ATmega16	15
Gambar 2.7.	Status register ATmega16	15
Gambar 2.8.	Register MCUCR	17
Gambar 2.9.	Register seluruh kontrol Interupsi	18
Gambar 2.10.	Blok skematik ADC	20
Gambar 2.11.	Register ADMUX	21
Gambar 2.12.	Format data ADC dengan ADLAR = 0	21
Gambar 2.13.	Format data ADC dengan ADLAR = 1	22
Gambar 2.14.	Register ADCSRA	23
Gambar 2.15.	Register SFIOR	24
Gambar 2.16.	Gerbang AND	25
Gambar 2.17.	Gerbang OR	26
Gambar 2.18.	Gerbang NOR	26
Gambar 2.19.	Gerbang NOT	27
Gambar 2.20.	Gerbang XOR	27
Gambar 2.21.	Gerbang XNOR	27
Gambar 2.22.	Proses suara ke pendengar	28
Gambar 2.23.	Gelombang suara	29
Gambar 2.24.	Bentuk sinyal	30
Gambar 2.25.	Bentuk titik sampling pada sinyal	31
Gambar 3.1.	Diagam blok sistem enkripsi sinyal audio	34

Gambar 3.2. Rangkaian Input data	38
Gambar 3.3. Rangkaian LED mode	39
Gambar 3.4. Rangkaian Koneksi SPI	39
Gambar 3.5. Rangkaian tombol saklar mode	40
Gambar 3.6. Rangkaian output data	40
Gambar 3.7. Rangkaian osilator dan reset	41
Gambar 3.8. Rangkaian lengkap skematik	42
Gambar 3.9. Rangkaian power suplai	43
Gambar 3.10. Diagram alir program utama	45
Gambar 3.11. Diagram alir enkripsi mode 1	47
Gambar 3.12. Diagram alir enkripsi mode 2	48
Gambar 4.1. (a) Hasil pengujian input sinyal dengan frekuensi 1000 Hz (b) Hasil pengujian output sinyal dengan kondisi normal	53
Gambar 4.2. Hasil pengujian input sinyal normal dengan mode 1	54
Gambar 4.3. Hasil pengujian input sinyal normal dengan mode 2	56
Gambar 4.4. Hasil pengujian input sinyal normal dengan mode 1 dan 2	57



BAB.I

PENDAHULUAN

1.1. Latar Belakang

Masalah keamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu informasi. Dalam hal ini sangat terkait dengan betapa pentingnya informasi tersebut dikirim dan diterima oleh orang yang berkepentingan. Informasi akan tidak berguna lagi apabila di tengah jalan informasi itu disadap atau dibajak oleh orang yang tidak berhak. Pada garis besarnya, masalah keamanan jaringan dapat dibagi menjadi empat bidang yang saling berhubungan: kerahasiaan, keaslian, pengakuan dan kontrol integritas.

Untuk mengantisipasi berbagai masalah kebocoran informasi rahasia, maka diperlukan sebuah sistem yang *real time*, untuk mengamankan informasi yang dikirimkan tersebut. Untuk keperluan pengamanan data tersebut dapat dilakukan sebuah metode yang dikenal sebagai metode enkripsi data. Enkripsi adalah sebuah proses yang melakukan perubahan suatu kode dari yang bisa dimengerti menjadi sebuah kode yang tidak dimengerti atau tidak terbaca. Untuk menjalankan metode tersebut, maka diperlukan enskriptor dan deskriptor.

Enkripsi dapat diartikan sebagai kode atau *chipper*. Sebuah sistem pengkodean dengan menggunakan suatu tabel atau kamus yang telah didefinisikan untuk mengganti kata dan informasi atau yang merupakan bagian dari informasi yang dikirim, dengan menggunakan suatu algoritma yang dapat mengkodekan

semua aliran data (*stream*) bit dari sebuah pesan menjadi *cryptogram* yang tidak dimengerti.

Karena teknik *chipper* merupakan suatu sistem yang telah siap untuk di automasi, maka teknik ini digunakan dalam sistem keamanan jaringan komunikasi. Beberapa contoh aplikasi sistem enkripsi data yaitu enkripsi data pada sistem komunikasi satelit, komunikasi pada ATM, komunikasi pada kepentingan militer, komunikasi kepentingan negara, atau komunikasi rahasia perusahaan tertentu, dan tentunya masih banyak aplikasi lainnya.

Seperti pengalaman negara maju yang ada, dengan berjalannya waktu, maka diharapkan akan meningkat pula sistem pengamanan yang ada. Mengingat pentingnya sistem ini untuk digunakan, baik sekarang maupun kelak dikemudian hari, maka akan sangat baik apabila segera dapat direalisasikan metode enkripsi yang ada. Selanjutnya, penelitian ini diharapkan akan dapat memberikan inspirasi pada penelitian berikutnya. Nantinya diharapkan akan dapat dikembangkan menjadi penelitian yang berkelanjutan menuju kesempurnaan sistem.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat dirumuskan masalah sebagai berikut, yaitu : Bagaimanakah rancangan enkriptor-simetrik sinyal audio berbasis mikrokontroler ATmega16.

1.3. Batasan Masalah

Dalam penelitian ini diberikan beberapa batasan masalah sebagai berikut :

BAB.II

LANDASAN TEORI

2.1. ENKRIPSI DATA

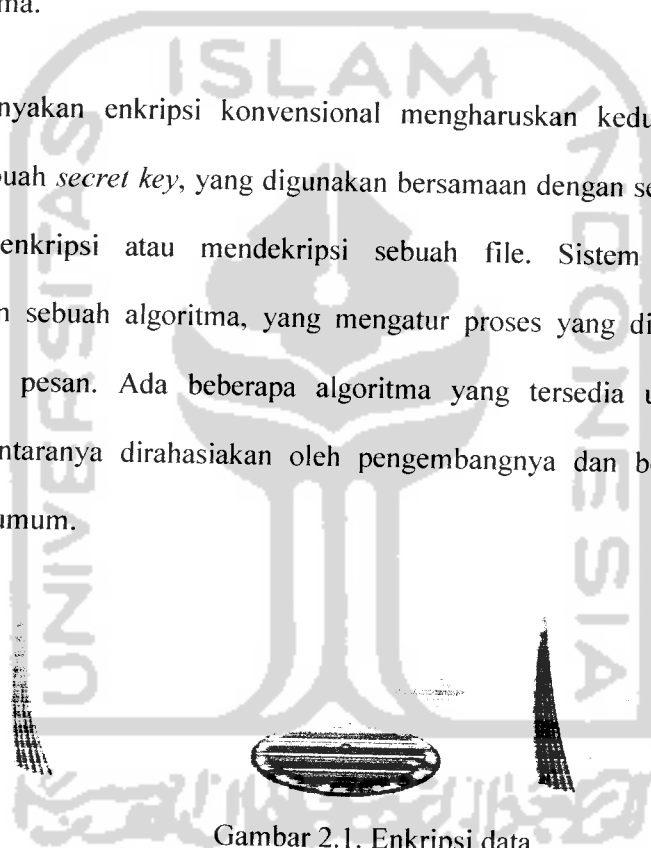
Menurut kamus istilah komputer, enkripsi adalah penerjemahan data menjadi kode rahasia. Enkripsi adalah metode yang efektif untuk keperluan pengamanan data. Untuk membaca data yang telah di-enskripsi, diperlukan akses kunci rahasia atau *password* yang dapat mengizinkan sistem untuk memberikan ijin untuk proses di-enskripsi. Data yang telah di-enskripsi dinamakan data *chipper*.

Kriptografi adalah seni memproteksi informasi dengan meng-enskripsi informasi ke dalam format yang tak dapat dibaca, yang dikenal dengan istilah *chipper data*. Informasi yang telah di-enskripsi kadang-kadang dapat dipecahkan oleh seorang Cryptanalysis, yang juga disebut dengan pemecah sandi, meskipun teknik kriptografi modern sebetulnya sulit, atau bahkan tak dapat dipecahkan.

Cippher yang digunakan dalam bidang militer meningkat sejak abad ke 18, dan metode yang digunakan semakin rumit. Sebagai tambahan dari metode substitusi, digunakan cara-cara lain seperti menyembunyikan pesan di dalam partitur musik. Perang sering kali menjadi waktu untuk inovasi kriptografi karena pentingnya menyimpan informasi dari jangkauan tangan musuh, dan militer tetap menjadi pemakai utama dari enkripsi. Sejak Perang Dunia II, metode mekanis telah umum dibandingkan sistem manual. Dari sistem seperti mesin

Enigma yang berasal dari Jerman, sampai *software packages* untuk komunikasi berbasis komputer seperti e-mail, atau *device* untuk mengenkripsi komunikasi suara. Enkripsi suara semakin rumit karena enkripsi biasanya beroperasi pada aliran data digital dan suara biasanya berupa sinyal analog – sehingga gelombang suara harus didigitalkan dan kemudian dienkripsi sebelum melakukan transmisi dan kemudian di dekripsi dan diubah kembali ke bentuk gelombang sinyal analog untuk penerima.

Kebanyakan enkripsi konvensional mengharuskan kedua pihak untuk membagi sebuah *secret key*, yang digunakan bersamaan dengan sebuah algoritma untuk mengenkripsi atau mendekripsi sebuah file. Sistem enkripsi juga membutuhkan sebuah algoritma, yang mengatur proses yang digunakan untuk mengenkripsi pesan. Ada beberapa algoritma yang tersedia untuk enkripsi, beberapa diantaranya dirahasiakan oleh pengembangnya dan beberapa dibuat terbuka buat umum.

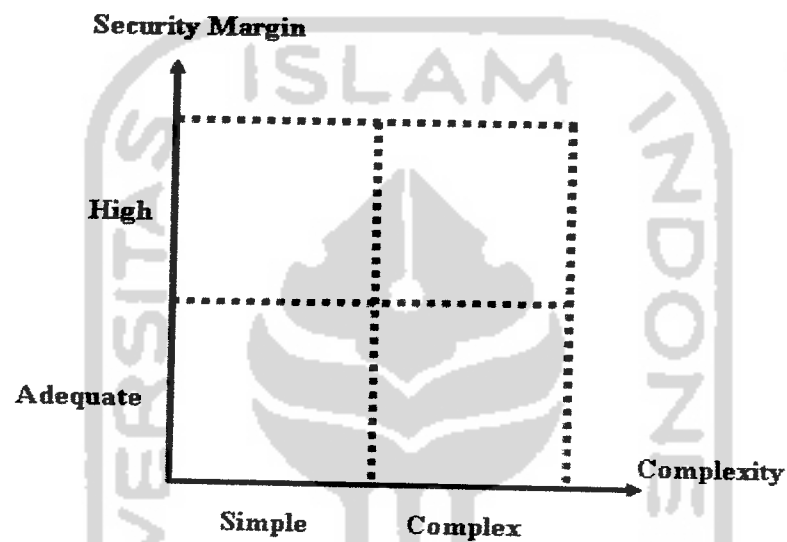


Gambar 2.1. Enkripsi data

2.1.1 Standarisasi Enkripsi

Pada awal tahun 1970, algoritma Standar enkripsi data diperkenalkan, yang menggunakan kunci 56 bit untuk mengenkripsi dan mendekripsikan informasi. DES membagi setiap pesan dalam blok-blok dan meng-*encode* setiap

blok satu pada setiap waktu. DES diadopsi sebagai algoritma yang diakui untuk penggunaan US Federal, tetapi tidak lagi dianggap cukup aman karena sebuah kunci 56-bit bisa dibuka secara paksa dalam waktu yang relatif cepat. DES kemudian diganti oleh *Advanced Encryption Standard* (AES), menggunakan algoritma Rijndael. AES beroperasi dengan kunci 128,192, atau 256 bit.



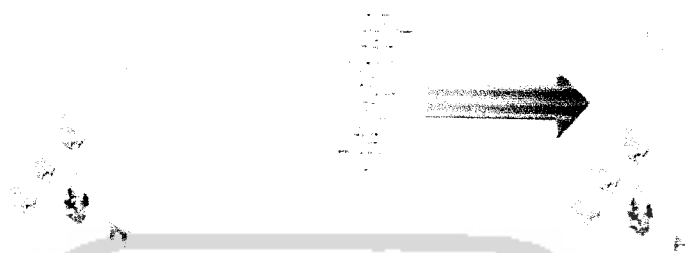
Gambar 2.2. AES (Advanced Encryption Standard)

Setiap algoritma yang paling aman karena memiliki kesederhanaan, keseimbangan antara keamanan dan fleksibilitas dan kecepatan tinggi dalam berbagai platform software dan hardware.

2.1.2 Model - Model Enkripsi

Dalam membahas model-model enkripsi beserta algoritma yang akan dipakai untuk setiap enkripsi ada 2, yaitu enkripsi dengan kunci pribadi dan enkripsi dengan kunci publik.

- a. **Enkripsi dengan kunci pribadi** atau enkripsi simetrik merupakan enkripsi yang nilai *password*-nya sama.



Gambar 2.3. Enkripsi Simetris

Sebuah proses enkripsi dan deskripsi data menggunakan kunci yang sama, artinya kunci yang digunakan untuk melakukan enkripsi sebuah pesan sama dengan kunci yang digunakan untuk deskripsi pesan tersebut. Kekuatan dari enkripsi jenis ini terletak pada ukurannya atau panjang kuncinya dan keamanan kunci enkripsi yang digunakan. Kunci simetris paling banyak digunakan karena kecepatan dan kemudahan algoritma yang digunakan.

Enkripsi yang sering digunakan: Substitution Cipher, Caesar Cipher, Transposition Cipher, Vigenere Cipher, Enigma Cipher, Data Encryption Standard (DES), Triple DES, Rivest Code 2 (RC2) dan Rivest Code 4 (RC4), Internasional Data Encryption Algorithm (IDEA), Skipjack, Gost Block Cipher.

- b. **Enkripsi dengan kunci publik** atau enkripsi asimetrik merupakan tipe enkripsi yang *password* enkripsi dengan *password* de-enskripsi-nya berbeda.

Figure 2.4. Asymmetric Encryption

Gambar 2.4. Enkripsi Asimetrik

Disebut kunci publik karena kunci yang digunakan untuk melakukan enkripsi dapat dipublikasikan atau disebar luaskan tanpa harus memperdulikan keamanan datanya karena untuk melakukan deskripsinya memerlukan kunci privat. Sehingga hanya orang yang memegang kunci privat (pasangan kunci publik) yang dapat melakukan deskripsi terhadap data tersebut. Jadi yang perlu dijaga keamanannya adalah kunci privat. Karena kunci privat merupakan kunci yang digunakan untuk membuka pesan yang telah disandikan menggunakan kunci publik. Enkripsi yang sering digunakan: Sistem Diffie Hellman, RSA (Ron Rivest, Adi Shamir dan Len Adleman), PGP (Pretty Good Privacy)

2.1.3 Manfaat dan kerugian menggunakan enkripsi

Beberapa manfaat yang bisa didapatkan dari enkripsi ini adalah :

- a. Kerahasiaan suatu informasi terjamin
- b. Menyediakan authentication dan perlindungan integritas pada algoritma checksum/hash
- c. Menanggulangi penyadapan telpon dan email

- d. Untuk digital signature. Digital signature adalah menambahkan suatu baris statemen pada suatu elektronik copy dan mengenkripsi statemen tersebut dengan kunci yang kita miliki dan hanya pihak yang memiliki kunci dekripsinya saja yang bisa membukanya.
- e. Untuk digital cash

Penyalahgunaan dan kerugian dari enkripsi adalah:

- a. Penyandian rencana teroris
- b. Penyembunyian rekaman kriminal oleh seorang penjahat
- c. Pesan tidak bisa dibaca bila penerima pesan lupa atau kehilangan kunci (decryptor).

2.2 MIKROKONTROLLER ATMEL AVR ATMEGA 16

2.2.1 Gambaran Umum

AVR (*Alf and Vegard's Rise processor*) secara umum dapat di kelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, Keluarga AT90Sxx, Keluarga ATMega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, bisa dikatakan hampir sama. Mikrokontroler ATMEL AVR ATMega16 adalah mikrokontroler buatan ATMEL yang merupakan keluarga dari seri AVR. Mikrokontroler ini merupakan Mikrokontroler ATMEL Generasi yang lebih baru dari MCS 51 yang berteknologi CISC (*Complex Instruction Set Computing*).

4. CPU terdiri atas 32 buah register.
5. Memori *Flash* sebesar 16 Kbyte dengan kemampuan Read While Write
6. EEPROM sebesar 512 byte.
7. Whatchdog Timer dengan osilator internal
8. Unit interupsi internal dan eksternal
9. Port antarmuka SPI
10. Antar muka komparator analog
11. Port USART untuk komunikasi serial
12. Tiga buah Timer/Counter dengan kemampuan perbandingan.

2.2.3. Fitur ATmega 16

Kapabilitas detail dari ATmega16 adalah sebagai berikut :

1. Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 8 MIPS.
2. Kapabilitas memori flash 8 KB, SRAM sebesar 1 Kbyte, dan EEPROM (*Elektrically Erasable Programmable Read Only Memory*) sebesar 512 byte.
3. ADC internal dengan fidelitas 10 bit sebanyak 8 channel.
4. Portal Komunikasi Serial (USART) dengan kecepatan maksimal 2,5 Mbp

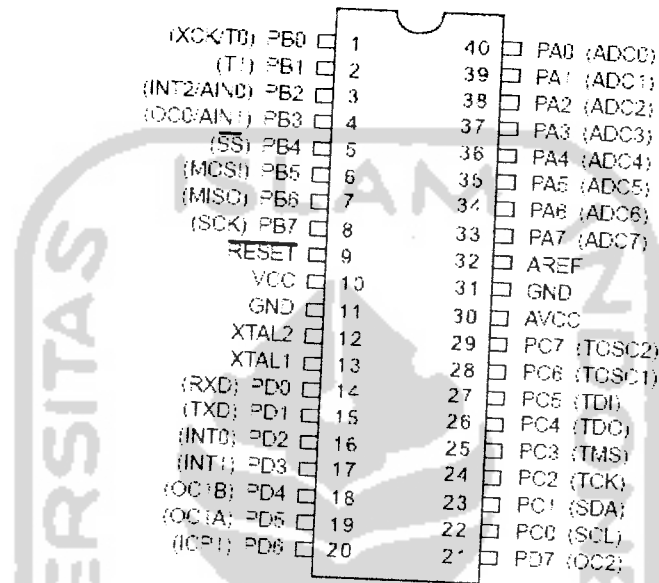
- eksternal dialiri daya rendah akan menjadi sumber arus jika internal *pull-up* register aktif. Pin pada kondisi *tri-state* ketika terjadi reset.
6. Port D (PD0-PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial. Port ini merupakan kelompok 8-bit *bi-directional I/O Port* dengan internal *pull-up* register. Dapat diberi *pull-up* secara internal. INT 0 – INT 1 disediakan untuk operasi interupsi secara eksternal. Sebagai input, port D yang secara eksternal dialiri daya rendah akan menjadi sumber arus apabila *pull-up* register aktif. Pin ini pada kondisi *tri-state* ketika terjadi reset.
 7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler. Reset digunakan untuk input reset. Pin ini dapat diaktifkan dengan *logic low* selama 50 ms. Dengan syarat itu, maka mikrokontroler dapat dalam kondisi reset.
 8. XTAL1 dan XTAL2 merupakan pin masukan klock eksternal. XTAL 1 merupakan input dari *inverting* osilator dan XTAL 2 merupakan output dari *inverting* osilator A/D dengan Vcc.
 9. AVCC merupakan pin masukan tegangan untuk ADC
AVCC digunakan sebagai suplai tegangan untuk port A dan A/D *converter*. Apabila ADC tidak digunakan pin ini harus dikoneksikan dengan Vcc. Jika ADC digunakan, pin ini harus dikoneksikan ke Vcc melalui *low-pass filter*.

2.2.4. Konfigurasi Pin ATMEGA16

Konfigurasi pin ATmega16 dapat dijelaskan secara fungsional sebagai berikut :

1. VCC merupakan pin yang berfungsi sebagai masukan catu daya
2. GND merupakan pin ground.
3. Port A (PA0-PA7) merupakan pin I/O dua arah dan pin masukan ADC. Port ini merupakan satu kelompok 8-bit *bi-directional I/O Port*. Jika *A/D Converter* tidak dapat digunakan, pin ini pada kondisi *tri-state* ketika terjadi reset, dapat diberi *pull-up* secara internal. Ketika pin PA0 hingga pin PA7 digunakan sebagai input dan secara eksternal dialiri daya rendah, maka akan menjadi sumber arus jika internal *pull-up* register aktif.
4. Port B (PB0-PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/ Counter, Komparator analog, dan SPI. Port ini merupakan satu kelompok 8-bit *bi-directional I/O Port* dengan internal *pull-up* register (dipilih untuk tiap bit). Sebagai input port B yang secara eksternal dialiri daya rendah akan menjadi sumber arus jika internal *pull-up* register aktif. Pin pada kondisi *tri-state* ketika terjadi reset.
5. Port C (PC0-PC7) merupakan pin I/O dua arah dan pin fungsi khusus yaitu TWI, Komparator analog, dan *Timer Oscillator*. Port ini merupakan satu kelompok 8-bit *bi-directional I/O Port* dengan internal *pull-up* register. Sebagai input port C yang secara

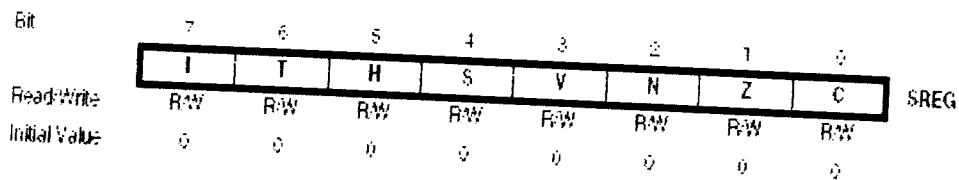
10. AREF merupakan pin masukan tegangan referensi ADC atau input referensi analog untuk A/D converter.
11. AGND analog ground. Jika board memiliki bidang ground analog terpisah, pin ini sebaiknya dikoneksikan ke dalam bidang ground ini.



Gambar 2.6. Pin mikrokontroler ATmega16

2.2.5 Status Register (SREG)

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



Gambar 2.7. Status Register ATmega16

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

Bit harus di set untuk mengaktifkan interupsi. Setelah itu, dapat mengaktifkan interupsi mana yang akan digunakan dengan cara mengaktifkan bit kontrol register yang bersangkutan secara individu. Bit akan di bersihkan apabila terjadi suatu interupsi yang dipicu oleh *hardware*, dan bit tidak akan mengizinkan terjadinya interupsi, serta akan diset kembali oleh instruksi RETI.

b. Bit 6 – T : *Bit Copy Storage*

Intruksi BLD dan BST menggunakan bit-T sebagai sumber atau tujuan dalam operasi bit. Suatu bit dalam sebuah register GPR dapat disalin ke bit T menggunakan instruksi BST, dan sebaliknya bit-T dapat disalin kembali ke suatu bit dalam register GPR menggunakan instruksi BLD.

c. Bit 5 – H : *Half Carry Flag*

d. Bit 4 – S : *Sign Bit*. Bit-S merupakan hasil operasi EOR antara flag-N (negatif) dan flag V(komplemen dua overflow).

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

Bit berguna untuk mendukung operasi aritmatika

f. Bit 2 – N : *Negative Flag*

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

g. Bit 1 – Z : *Zero Flag*

Bit akan diset bila hasil operasi yang diperoleh adalah nol.

h. Bit 0 – C : *Carry Flag* Apabila suatu operasi menghasilkan carry, maka bit akan diset.

2.2.6. Interupsi

Interupsi adalah suatu kejadian yang akan menghentikan sementara jalannya program yang sedang berjalan untuk menjalankan suatu subrutin interupsi tertentu. Setelah selesai dikerjakan maka program yang dihentikan tadi akan dilanjutkan kembali secara normal.

Pada AVR terdapat 3 pin untuk interupsi eksternal, yaitu INT0, INT1 dan INT2. Interupsi eksternal dapat dibangkitkan apabila terdapat perubahan logika atau logika 0 pada pin interupsi. Pengaturan kondisi keadaan yang menyebabkan terjadinya interupsi eksternal diatur oleh register MCUCR (MCU Control Register) yang terlihat pada gambar dibawah ini :

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	"	"	"	0	"	0	0	0	

Gambar 2.8. Register MCUCR

Bit penyusunnya dapat dijelaskan berikut :

- Bit ISC11 dan ISC10 bersama-sama menentukan kondisi yang dapat menyebabkan interupsi eksternal pada pin INT1. Keadaan selengkapnya dapat dilihat pada tabel 2.1.berikut .

Tabel 2.1. Setting kondisi menyebabkan interupsi eksternal 1

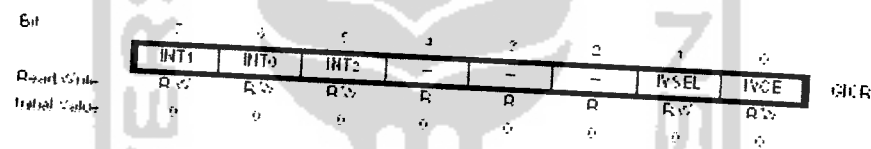
ISC11	ISC10	Deskripsi
0	0	Logika 0 pada pin INT1 menyebabkan interupsi
0	1	Perubahan logika pada pin INT1 menyebabkan interupsi
1	0	Perubahan kondisi 1 ke 0 pada pin INT1 menyebabkan interupsi
1	1	Perubahan kondisi 0 ke 1 pada pin INT1 menyebabkan interupsi

- b. Bit ISC01 dan ISC00 bersama-sama menentukan kondisi yang dapat menyebabkan interupsi eksternal pada pin INT0. Keadaan selengkapnya dapat dilihat pada tabel 2.2. berikut .

Tabel 2.2. Setting kondisi menyebabkan interupsi eksternal 0

ISC01	ISC00	Deskripsi
0	0	Logika 0 pada pin INT0 menyebabkan interupsi
0	1	Perubahan logika pada pin INT0 menyebabkan interupsi
1	0	Perubahan kondisi 1 ke 0 pada pin INT0 menyebabkan interupsi
1	1	Perubahan kondisi 0 ke 1 pada pin INT0 menyebabkan interupsi

Pemilihan pengaktifan interupsi eksternal diatur oleh register GICR (General Interrupt Control Register) yang terlihat seperti gambar berikut :



Gambar 2.9. General Interrupt Control Register

Bit penyusun dapat dijelaskan sebagai berikut :

- Bit INT1 adalah bit yang mengaktifkan interupsi eksternal 1. Apabila bit tersebut diberi logika 1 dan bit-I pada SREG (status register) juga satu, maka interupsi eksternal 1 akan aktif.
- Bit INT0 adalah bit untuk mengaktifkan interupsi eksternal 0. Apabila bit tersebut diberi logika 1 dan bit-I pada SREG (status register) juga satu, maka interupsi eksternal 0 akan aktif.

- c. Bit INT2 adalah bit untuk mengaktifkan interupsi eksternal 2. Apabila bit tersebut diberi logika 1 dan bit-I pada SREG (status register) juga satu, maka interupsi eksternal 2 akan aktif.

Program interupsi dari masing-masing jenis interupsi eksternal akan dimulai dari vektor interupsi pada masing-masing jenis. Alamatnya dapat dilihat pada tabel 2.3. seperti berikut ini.

Tabel 2.3. Sumber Interupsi pada ATmega16

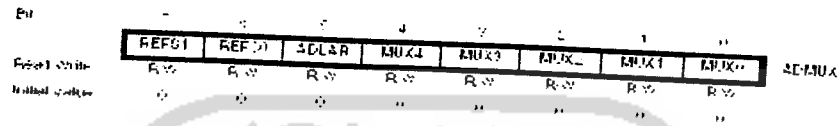
Vector No.	Program Address ¹²	Source	Interrupt Definition
1	\$0001	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$0002	INT0	External Interrupt Request 0
3	\$0004	INT1	External Interrupt Request 1
4	\$0006	TIMER2 COMP	Timer Counter2 Compare Match
5	\$0006	TIMER2 OVF	Timer Counter2 Overflow
6	\$000A	TIMER1 CAPT	Timer Counter1 Capture Event
7	\$000C	TIMER1 COMFA	Timer Counter1 Compare Match A
8	\$000E	TIMER1 COMPB	Timer Counter1 Compare Match B
9	\$0010	TIMER1 OVF	Timer Counter1 Overflow
10	\$0012	TIMER0 OVF	Timer Counter0 Overflow
11	\$0014	SPI STC	Serial Transfer Complete
12	\$0016	USART, RXC	USART, Rx Complete
13	\$0016	USART, UDRE	USART Data Register Empty
14	\$001A	USART, TXC	USART, Tx Complete
15	\$001C	ADC	ADC Conversion Complete
16	\$001E	EE_RDY	EEPROM Ready
17	\$0020	ANA_COMP	Analog Comparator
18	\$0022	TWI	Two-wire Serial Interface
19	\$0024	INT2	External Interrupt Request 2
20	\$0026	TIMER0 COMP	Timer Counter0 Compare Match
21	\$0028	SPM_RDY	Store Program Memory Ready

2.2.7. ADC (Pengubah Analog ke Digital)

ATmega16 merupakan tipe AVR yang telah dilengkapi dengan 8 saluran ADC internal dengan kebenaran 10 bit. Selain itu ADC ATmega16 memiliki

nilainya adalah ADMUX (*ADC Multiplexer Selection Register*), ADCSRA (*ADC Control and Status Register A*), SFIOR (*Special Function IO Register*).

ADMUX merupakan register 8 bit yang berfungsi menentukan tegangan referensi ADC, format data output, dan saluran ADC yang digunakan. Konfigurasinya seperti gambar di bawah ini.



Gambar 2.11. Register ADMUX

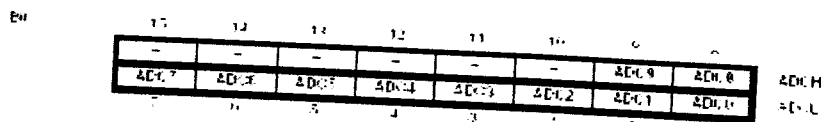
Bit penyusunnya adalah sebagai berikut :

- a. REFS[1-0] merupakan bit pengatur tegangan referensi ADC ATmega16. Memiliki nilai awal 00 sehingga referensi tegangan berasal dari pin ARIF. Detail nilai yang lain dapat dilihat pada tabel 2.4 berikut :

Tabel 2.4. Pemilihan mode tegangan referensi ADC

REFS[1..0]	Mode tegangan referensi
00	Berasal dari pin AREF
01	Berasal dari pin AVCC
10	Tidak dipergunakan
11	Berasal dari tegangan referensi internal sebesar 2,56 V

- b. ADLAR merupakan bit pemilih mode data keluaran ADC. Bernilai awal 0 sehingga 2 bit tertinggi data hasil koversinya berada di register ADCH dan 8 bit sisanya berada di register ADCL.



Gambar 2.12. Format data ADC dengan ADLAR = 0

	15	14	13	12	11	10	9	8	AD[7:0]
AD[9]	AD[8]	AD[7]	AD[6]	AD[5]	AD[4]	AD[3]	AD[2]	AD[1]	AD[0]
AD[9]	AD[8]	-	-	-	-	-	-	-	-
	7	6	5	4	3	2	1	0	

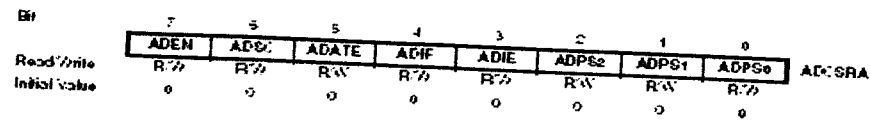
Gambar 2.13 . Format data ADC dengan ADLAR = 1

- c. MUX[4-0] merupakan bit pemilih saluran pembacaan ADC. Bernilai awal 00000. Untuk mode *single ended* input, MUX[4..0] bernilai dari 00000 – 00111. Konfigurasi bit MUX dapat dilihat pada tabel 2.5 berikut.

Tabel 2.5. Pemilihan Bit Saluran Pembacaan ADC

MUX[4:0]	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
00000	AD[0]	NA		
00001	AD[1]			
00010	AD[2]			
00011	AD[3]			
00100	AD[4]			
00101	AD[5]			
00110	AD[6]			
00111	AD[7]			
01000		AD[0]	AD[0]	1/2
01001		AD[1]	AD[0]	1/2
01010 ⁰¹		AD[0]	AD[1]	2/3
01011 ⁰¹		AD[1]	AD[0]	2/3
01100		AD[2]	AD[2]	1/2
01101		AD[2]	AD[2]	1/2
01110 ⁰¹		AD[2]	AD[2]	1/2
01111 ⁰¹		AD[3]	AD[2]	2/3
10000		AD[0]	AD[1]	2/3
10001		AD[1]	AD[1]	1/2
10010	NA	AD[2]	AD[1]	1/2
10011		AD[2]	AD[1]	1/2
10100		AD[4]	AD[1]	1/2
10101		AD[5]	AD[1]	1/2
10110		AD[6]	AD[1]	1/2
10111		AD[7]	AD[1]	1/2
11000		AD[0]	AD[2]	1/2
11001		AD[1]	AD[2]	1/2
11010		AD[2]	AD[2]	1/2
11011		AD[3]	AD[2]	1/2
11100		AD[4]	AD[2]	1/2

ADCSRA merupakan register 8 bit yang berfungsi melakukan manajemen sinyal sinyal kontrol dan status dari ADC.

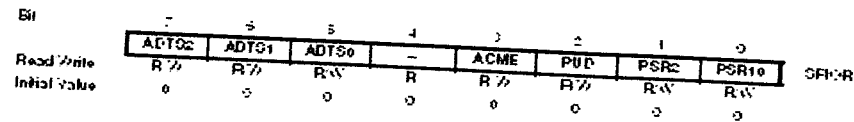


Gambar 2.14. Register ADCSRA

Bit penyusunnya adalah :

- ADEN merupakan bit pengatur aktivasi ADC. Bernilai awal 0. Jika bernilai 1, maka ADC aktif.
- ADSC merupakan bit penanda mulainya konversi ADC. Bernilai awal 0 selama konversi ADC akan bernilai 1, sedangkan jika konversi telah selesai, akan bernilai 0.
- ADIF merupakan bit pengatur aktivasi picu otomatis operasi ADC. Bernilai awal 0. Jika bernilai 1, operasi konversi ADC akan dimulai pada saat transisi positif dari sinyal picu yang dipilih. Pemilihan sinyal picu menggunakan bit ADTS pada register SFIOR.
- ADIE merupakan bit penanda akhir suatu konversi ADC. Bernilai awal 0. Jika bernilai 1, maka konversi ADC pada suatu saluran telah selesai dan data siap diakses.
- ADIE merupakan bit pengatur aktivasi interupsi yang berhubungan dengan akhir konversi ADC. Bernilai awal 0. Jika bernilai 1 dan jika sebuah konversi ADC telah selesai, sebuah interupsi akan dieksekusi.
- ADPS[2-0] merupakan bit pengatur clock ADC.

SFIOR merupakan register 8 bit pengatur sumber picu konversi ADC, apakah dari picu eksternal atau dari picu internal .



Gambar 2.15. Register SFIOR

ADTS[2-0] merupakan bit pengatur picu eksternal operasi ADC. Hanya berfungsi jika bit ADATE pada register ADCSRA bernilai 1. Bernilai awal 000 sehingga ADC bekerja pada mode free running dan tidak ada interupsi yang akan dihasilkan. Detail nilai ADTS[2-0] dapat dilihat ada tabel 2.6 berikut :

Tabel 2.6. Pemilihan sumber Picu ADC

ADTS[2..0]	Sumber picu
000	Mode free running
001	Komparator analog
010	Interupsi eksternal
011	Timer Counter 0 Compare Match
100	Timer Counter 0 Overflow
101	Timer Counter 0 Compare Match B
110	Timer Counter 1 Overflow
111	Timer Counter 1 Capture Event

Untuk operasi ADC, bit ACME, PUD, PSR2, dan PSR10 tidak diaktifkan.

2.3 GERBANG LOGIKA

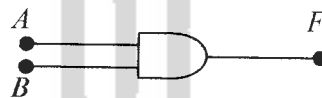
Istilah “logika” biasanya digunakan untuk menyatakan suatu proses pengambilan keputusan. Maka suatu gerbang logika merupakan suatu rangkaian yang dapat memutuskan untuk berkata ya atau tidak pada keluaran berdasarkan masukan.

Logika sangat penting dalam aspek kehidupan manusia. Dengan menggunakan logika lebih efektif dalam mengenal dan menghindari kesalahan penalaran. Adapun aplikasi logika salah satunya di bidang teknologi informasi dimana di bidang ini mengenal perangkat komputer serta bahasa pemrogramannya. Perangkat komputer tersebut biasanya disebut *hardware* dan bahasa pemrograman biasa disebut *software*, dan baik *software* ataupun *hardware* banyak menggunakan prinsip-prinsip logika.

a. Gerbang AND

Gerbang AND dapat diwakili oleh sejumlah saklar yang dihubungkan secara seri. Indikator bernilai ("1" secara logis) jika A dan B tertutup, yaitu keduanya "1" secara logis. Tabel kebenaran memperlihatkan keadaan output untuk semua kemungkinan kombinasi input. Tabel kebenaran ini akan menyatakan persamaan Boole.

$$\text{Output } (F) = A.B$$



Gambar 2.16. Gerbang AND

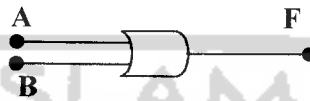
Tabel 2.7. Tabel kebenaran gerbang AND

Input		output
A	B	(F)
0	0	0
0	1	0
1	0	0
1	1	1

b. Gerbang OR

Gerbang OR dapat dinyatakan oleh sejumlah saklar yang di hubungkan secara paralel. Indikator bernilai ("1" secara logis) jika A dan B atau keduanya tertutup, yaitu "1" secara logis. Simbol logika untuk rangkaian dua buah input dan tabel kebenaran ditunjukkan sebagai berikut :

Persamaan Boole-nya ialah Output (F) = $A + B$



Gambar 2.17. Gerbang OR

Tabel 2.8. Tabel kebenaran gerbang OR

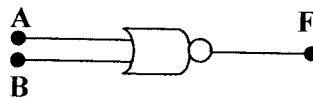
Input		output
A	B	(F)
0	0	0
0	1	1
1	0	1
1	1	1

c. Gerbang NOR

Gerbang ini sama dengan NOT-OR dan dapat dibuat dari sebuah gerbang OR yang diikuti oleh gerbang NOT. Simbol dan tabel kebenaran gerbang NOR diperlihatkan sebagai berikut :

Persamaan Boole-nya ialah

$$\text{Output } (F) = \overline{A + B}$$



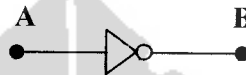
Gambar 2.18. Gerbang NOR

Tabel 2.9. Tabel kebenaran gerbang NOR

Input		output
A	B	(F)
0	0	1
0	1	0
1	0	0
1	1	0

d. Gerbang NOT

Biasanya gerbang ini disebut inverter (penginversi), negater, atau hanya pengubah tanda. Lingkaran kecil yang berarti berubah tandanya. Simbol gerbang NOT diperlihatkan sebagai berikut :

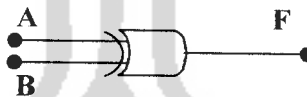


Gambar 2.19. Gerbang NOT

e. Gerbang OR Eklusif dan NOR Eklusif

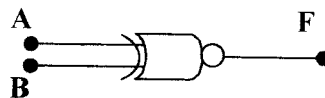
Persamaan Boole-nya ialah

$$\text{Output Xor } (F) = A \oplus B$$



Gambar 2.20. Gerbang XOR

$$\text{Output XNor } (F) = \overline{A \oplus B}$$



Gambar 2.21. Gerbang XNOR

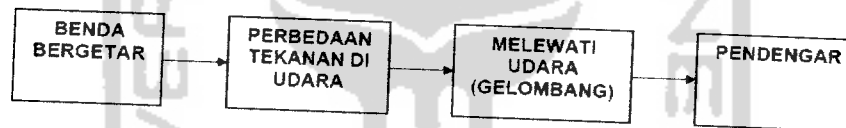
Tabel 2.10. Tabel kebenaran gerbang XOR dan XNOR

Input		output XOR	Output XNOR
A	B		
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

2.4 KONSEP DASAR AUDIO

2.4.1 Suara

Suara adalah fenomena fisik yang dihasilkan oleh getaran benda dan getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu.

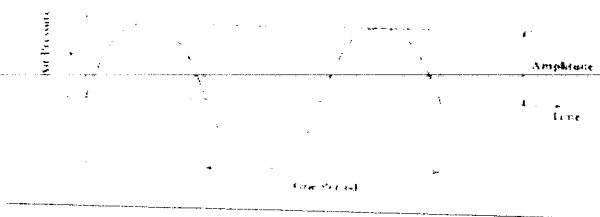


Gambar. 2.22 Proses suara ke pendengar

Suara berhubungan erat dengan rasa “mendengar” suara/bunyi biasanya merambat melalui udara. Suara/bunyi tidak bisa merambat melalui ruang hampa. Suara dihasilkan oleh getaran suatu benda. Selama bergetar perbedaan tekanan terjadi di udara sekitarnya. Pola osilasi yang terjadi dinamakan sebagai gelombang. Periode merupakan gelombang mempunyai pola sama yang berulang pada interval tertentu.

Contoh suara periodik : instrumen musik, nyanyian burung.

Contoh suara nonperiodik : batuk, percikan ombak.



Gambar 2.23. Gelombang suara

Hal – hal yang berkaitan erat dengan suara :

2.4.1.1 Frekuensi

Frekuensi adalah banyaknya periode dalam 1 detik. Satuan : *Hertz (Hz)* atau *cycles per second (cps)*

Berdasarkan frekuensi, suara yang ada di muka bumi dibagi menjadi:

- a. Infrasonik 0Hz – 20 Hz
- b. Pendengaran manusia 20Hz – 20 KHz
- c. Ultrasonik 20KHz – 1 GHz
- d. Hipersonik 1GHz – 10 THz

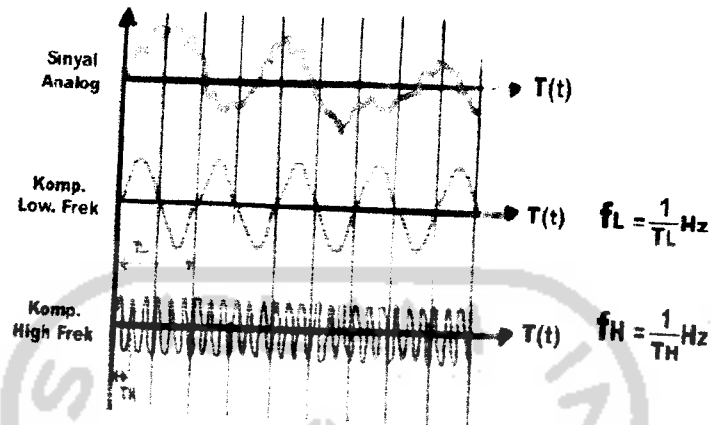
Pendengaran manusia dibagi berdasarkan sumber frekuensi, yaitu :

- a. Sinyal suara manusia berkisar antara frekuensi 50Hz – 10KHz
- b. Sinyal suara musik berkisar antara frekuensi 20Hz – 20KHz .

Sistem multimedia menggunakan suara yang berada dalam range pendengaran manusia.

Suara yang berada pada range pendengaran manusia sebagai audio, dan gelombangnya sebagai *accoustic* sinyal. Suara diluar range pendengaran manusia

dapat dikatakan sebagai noise (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi, tidak dapat didengar manusia).



Gambar 2.24. Bentuk sinyal

Analisis Fourier merupakan suatu sinyal analog terdiri dari sebuah frekuensi sinusoidal dimana amplitudonya serta fasenya berubah secara “relatif” antara satu dengan lainnya.

2.4.1.2 Amplitudo

Keras lemahnya bunyi atau tinggi rendahnya gelombang. Satuan amplitudo adalah decibel (db). Bunyi mulai dapat merusak telinga jika tingkat volumenya lebih besar dari 85 dB dan pada ukuran 130 dB akan mampu membuat hancur gendang telinga.

2.4.1.3 Velocity

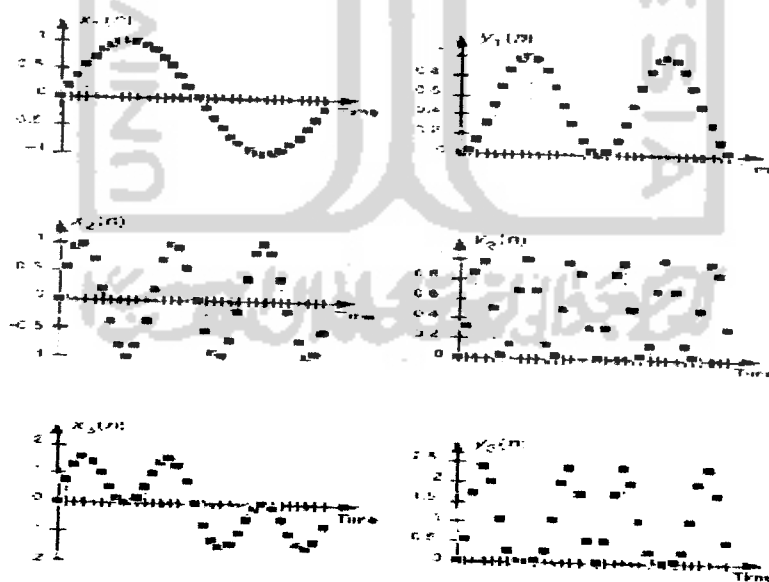
Kecepatan perambatan gelombang bunyi sampai ke telinga pendengar. Satuan yang digunakan : m/s. Pada udara kering dengan suhu 20 °C (68 °F) kecepatan rambat suara sekitar 343 m/s.

2.4.2 Representasi suara

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan sampel.

2.4.3 Teori sampling

Teori sampling merupakan teori yang penting dan mendasar pada komunikasi digital. Pada komunikasi yang mengirimkan sinyal hasil pencuplikan, pada sisi penerima rekonstruksi sinyal informasi dapat digunakan dengan menggunakan filter (LPF). Secara matematis akan dapat dipahami bahwa proses rekonstruksi dengan kecatatan yang minimum hanya memungkinkan bila frekuensi sampling (pencuplikan). Teknik sampling merupakan teknik memungkinkan sinyal dirubah dan diproses sehingga menjadi bit-bit digital.



Gambar 2.25. Bentuk titik sampling pada sinyal

ADC (Pengubah Analog ke Digital) merupakan proses mengubah amplitudo gelombang bunyi ke dalam waktu interval tertentu (disebut juga sampling), sehingga menghasilkan representasi digital dari suara.

Frekuensi sampling (F_s) atau *Nyquist Sampling Rate* merupakan untuk memperoleh representasi akurat dari suatu sinyal analog secara lossless, amplitudonya harus diambil sampel-nya setidaknya pada kecepatan (rate) sama atau lebih besar dari 2 kali lipat komponen frekuensi maksimum sinyal analog yang akan dicuplik.

Beberapa hal yang berhubungan dengan ADC (Pengubah analog ke digital) sebagai berikut :

- a. Sampling audio sinyal merupakan memilih data yang spesifik instant disuatu waktu.
- b. *Alliasing* merupakan hasil yang tidak tepat dengan hasilnya saat sinyal merekonstruksikan ketika sampling tersebut tidak tepat.
- c. *Sampling rate* merupakan beberapa gelombang yang diambil dalam satu detik. Minimum sampling rate, sama seperti 2x frekuensi tertinggi yang terakhir digunakan.

Jadi: $F_s \geq 2f_{maks}$(2.2)

- d. *Time sampling* merupakan waktu sampling dalam satu periode. Satuannya adalah detik (s) .

Waktu cuplik (T_s) = $\frac{1}{F_s}$(2.3)

Dimana : F_s = Frekuensi sampling (Hz)

- e. *Quantization Noise* merupakan perbandingan antara range dengan range centre.

Pemrosesan sinyal digital merupakan solusi yang pantas untuk seluruh masalah pemrosesan sinyal. Banyak sinyal dengan lebar pita (bandwidth) yang sangat lebar, pemrosesan waktu real sangat dibutuhkan. Rangkaian digital tidak hanya menghasilkan sistem yang lebih murah dan lebih dipercaya untuk pemrosesan sinyal, tetapi untuk keuntungan-keuntungan lain.

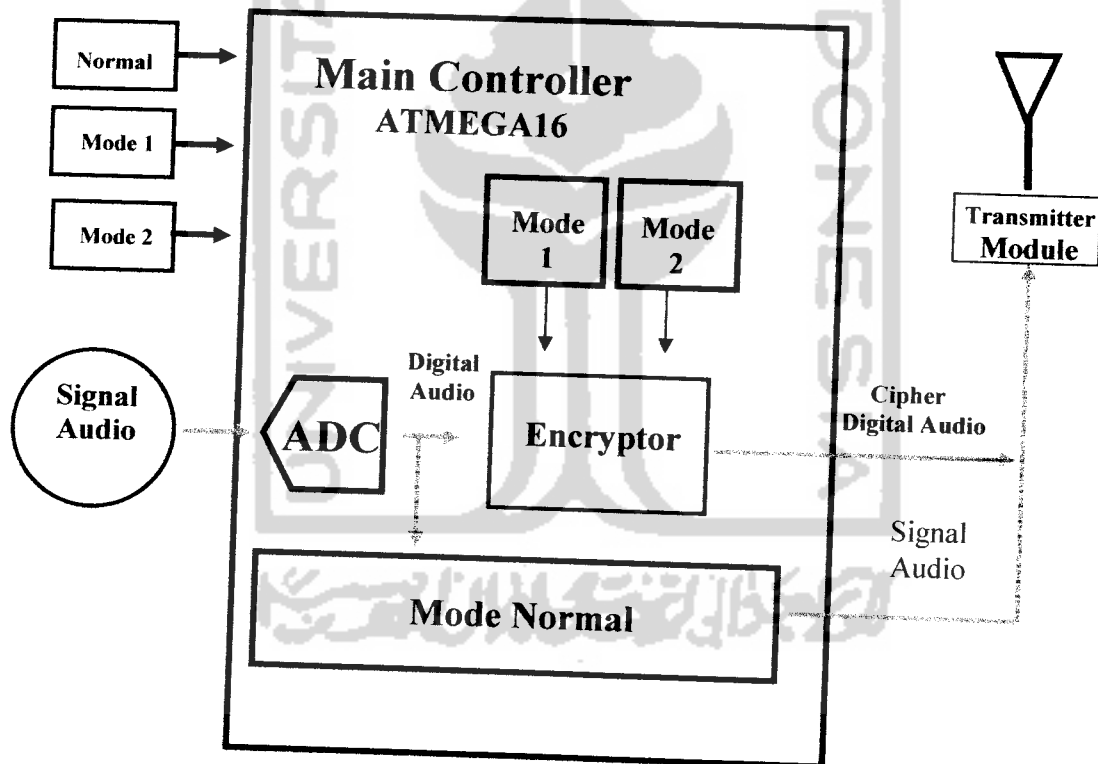
Suatu sinyal didefinisikan sebagai besaran fisik yang berubah-ubah menurut waktu, ruang atau variabel bebas atau variabel-variabel lainnya.

Suatu sistem dapat didefinisikan sebagai sebuah alat fisik dan perangkat lunak yang melakukan operasi pada suatu sinyal. Bila melewati sinyal melalui sebuah sistem, seperti dalam pentapisan, maka dapat dikatakan telah memproses sinyal tersebut. Pemrosesan sinyal meliputi pentapisan noise dan interferensi dari sinyal yang diinginkan. Umumnya sistem dicirikan dengan jenis operasi yang bekerja pada sinyal sebagai contoh, jika operasi tersebut linear, sistem ini dinamakan linear. Operasi seperti itu biasanya dinamakan sebagai pemrosesan sinyal.

BAB.III

PERANCANGAN SISTEM

Dalam bab III akan dibahas mengenai perancangan sistem yang didalamnya terdapat perancangan rangkaian elektronik enkripsi simetrik sinyal audio berbasis Mikrokontroler berdasarkan teori-teori yang dibahas pada bab sebelumnya. Untuk lebih memudahkan pemahaman cara kerja dari sistem yang akan dibuat, Gambar 3.1 memperlihatkan blok diagram dari perancangan sistem.



Gambar 3.1. Diagram blok sistem enkripsi sinyal audio



Berdasarkan diagram blok di atas, sistem dapat dibagi dalam beberapa bagian, yaitu :

1. Mikrokontroler

Mikrokontroler sebagai pusat pemrosesan sinyal audio memerlukan kecepatan yang tinggi serta memiliki fungsi-fungsi kendali. Hal ini tentu saja akan ikut mempengaruhi kecepatan akses data untuk menuju terwujudnya sistem yang *real time*.

Namun demikian, dalam sistem ini digunakan mikrokontroler Atmel AVR seri ATmega16, dengan pertimbangan bahwa mikrokontroler 8-bit ini memiliki pemrosesan data dengan proses internal 16-bit. Sehingga mikrokontroler ini sudah akan memiliki kecepatan pemrosesan 2 kali dari kecepatan pemrosesan yang dimiliki mikrokontroler seri 89C51. ATmega16 dilengkapi dengan ADC internal, yang diharapkan akan menghemat tempat perangkat keras. Selain itu diharapkan pula akan meningkatkan kecepatan akses, karena jarak ADC dengan prosesor akan lebih dekat.

Proses enkripsi akan dilakukan secara *software* dengan cara mengolah data digital yang dihasilkan dari ADC internal. Proses enkripsi menggunakan *fixed password* yang ditentukan nilainya. Pengguna cukup menekan tombol mode yang dikehendaki, yaitu tombol Mode 1 atau Mode 2. Dengan mode 1, maka enkripsi akan dilakukan dengan *fixed password 1*. Dengan mode 2, maka enkripsi akan dilakukan dengan *fixed password 2*. Mode Normal juga disediakan dalam sistem ini untuk mengantisipasi penggunaan sistem tanpa enkripsi. Pada mode normal, signal audio tidak akan dienkripsi.

2. Mode

Mode adalah tombol yang digunakan untuk proses enkripsi. Tombol ini berisi tiga buah mode yaitu tombol normal, mode 1 dan mode 2. Tombol Normal adalah tombol untuk menghentikan proses enkripsi, sehingga sinyal yang masuk diloloskan langsung menuju *transmitter* tanpa dilakukan enkripsi. Tombol Mode 1 adalah tombol perintah enkripsi dengan menggunakan *fixed password 1*. Dengan penekanan tombol ini, maka sistem akan mengenkripsi sinyal yang masuk. Tombol Mode 2 adalah tombol perintah enkripsi dengan menggunakan *fixed password 2*. Dengan penekanan tombol ini, maka sistem akan mengenkripsi sinyal yang masuk.

3. Transmitter Module

Modul ini digunakan untuk pengiriman data, sebagai aplikasi penggunaan sistem enkripsi yang ada. Modul *transmitter* dapat dipilih sesuai kebutuhan yang ada.

Prinsip kerja sistem dapat ditunjukkan dengan urutan kerja sebagai berikut:

1. Sistem akan melakukan inisialisasi / *setting* awal,
2. Apabila tombol Mode ditekan 1 kali, maka enkripsi akan dilakukan dengan *fixed password 1*.
3. Apabila tombol Mode ditekan 2 kali, maka enkripsi akan dilakukan dengan menggunakan *fixed password 2*.

4. Selanjutnya, apabila akan dilakukan penghentian proses enkripsi, maka dapat dilakukan dengan menekan tombol normal.
5. Proses enkripsi diawali dengan pengolahan sinyal yang masuk melalui internal ADC menjadi sinyal audio digital. sinyal digital tersebut, selanjutnya diolah oleh sistem *software* dengan model enkripsi simetrik dengan menggunakan *input* berupa sinyal audio digital dan *fixed password*. Hasil pengolahan dari sistem tersebut, akan menghasilkan chipper digital Audio, yaitu sinyal audio rahasia, yang telah diamankan, sehingga orang lain yang “menyadap” sinyal tersebut pada frekuensi transmitter yang sama dan mendengarkannya, maka tak ada informasi yang dapat mereka peroleh, kecuali mereka mampu memecahkan kode rahasianya dan menggunakan *Descriptor*.

3.1 Perancangan Perangkat keras

Pada penelitian ini perangkat keras yang dipergunakan berupa sebuah mikrokontroler ATmega16, yang didukung dengan perlengkapan-perengkapan seperti, tombol saklar password, tombol saklar reset , jack phone dan transmitter modul.

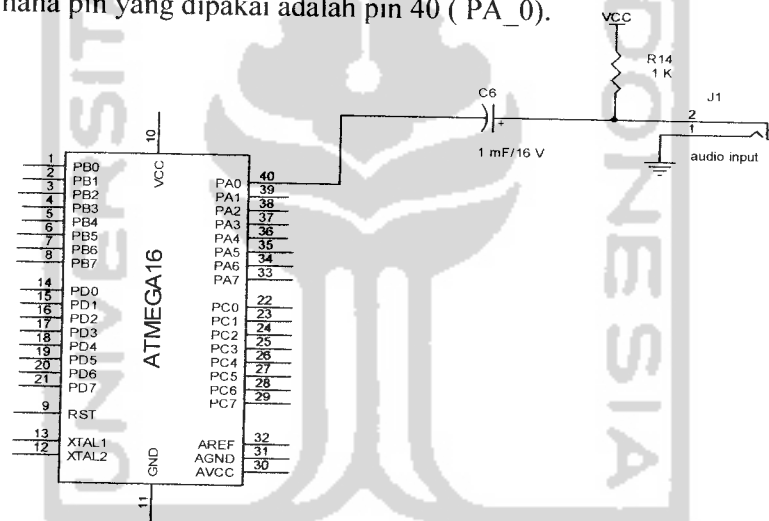
3.1.1 Rangkaian Pengoperasian Mikrokontroler

Mikrokontroler yang dipergunakan dalam perancangan alat ini adalah mikrokontroler tipe ATmega16, mikrokontroler ini termasuk dalam keluarga MCS-51, sehingga perintah-perintah dan fungsi dari pin-pinnya sama dengan keluarga MCS-51.

Adapun pemanfaatan dari pin-pin mikrokontroler ATmega16 pada alat tugas akhir ini. Menghubungkan pin 10 ke Vcc dan menghubungkan pin 11 ke ground, sedangkan pin 12 (XTAL1) dan pin 13 (XTAL2) diparalel dengan suatu kristal untuk memberikan eksternal clock untuk siklus kerja mikrokontroler. Kelebihan Mikrokontroler ATMega16 ini adalah memiliki ADC internal maka pin 30 untuk Vcc dan pin 31 untuk ground ADC nya. Pin 9 (RST) diberikan untuk reset.

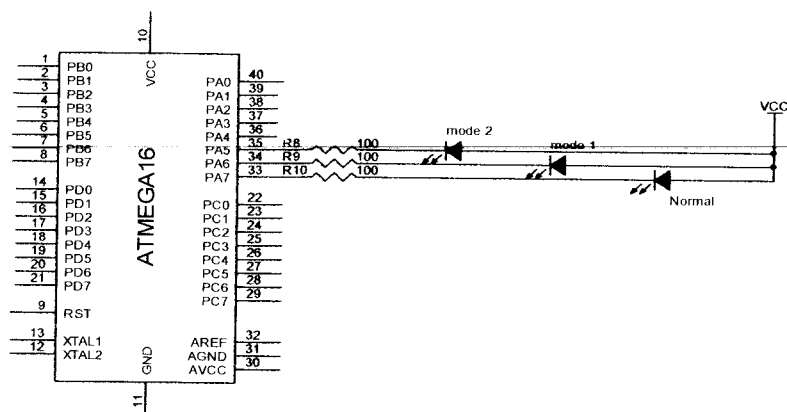
Port A Mikrokontroler ATMega16

Port A pada mikrokontroler ATMega16 digunakan sebagai input frekuensi audio. Dimana pin yang dipakai adalah pin 40 (PA_0).



Gambar 3.2. Rangkaian Input data

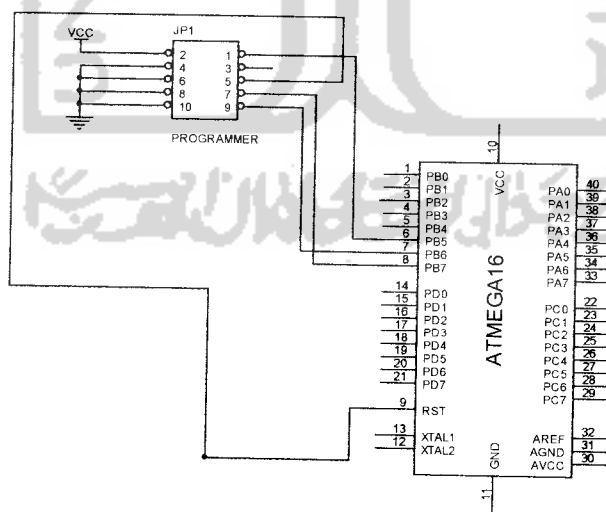
Port A juga sebagai masukan LED indikator mode-mode / password program, berfungsi untuk memberikan keterangan mode, dimana pin 35 (PA_5) dihubungkan dengan LED 1 (normal), pin 34 (PA_6) diubungkan dengan LED 2 (mode1), pin 33 (PA_7) dihubungkan dengan LED 3 (mode2).



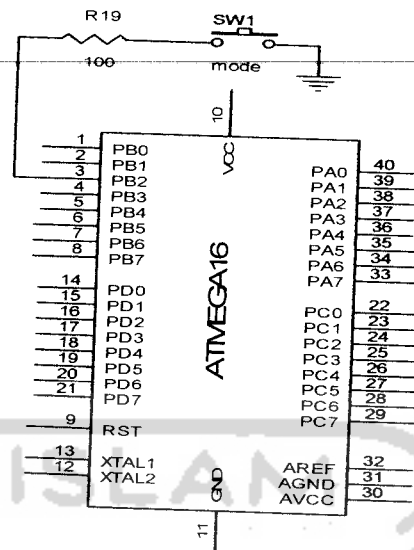
Gambar 3.3. Rangkaian LED mode

Port B Mikrokontroler ATMEGA16

Port B pada mikrokontroler ATMEGA16 digunakan untuk saluran *Serial Peripheral Interface* (SPI) atau saluran pemrograman mikrokontroler dan rangkaian tombol saklar mode. Dimana pin 8 SCK (PB_7) dihubungkan dengan pin 7 SPI, pin 7 MISO (PB_6) dihubungkan dengan pin 9 SPI, dan pin 6 MOSI (PB_5) dihubungkan dengan pin 1 SPI. pin 3 INT2 (PB_2) dihubungkan dengan tombol saklar mode. Pada rangkaian SPI, pin 3 SPI dihubungkan dengan pin 9 (RST) mikrokontroler ATMEGA16



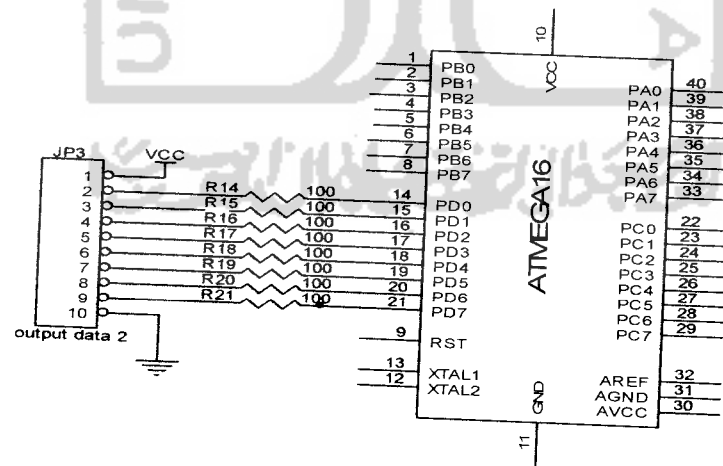
Gambar 3.4 . Rangkaian koneksi SPI



Gambar 3.5 . Rangkaian tombol saklar mode

Port D Mikrokontroler ATMEGA16

Port D pada mikrokontroler ATMEGA16 digunakan sebagai keluaran sinyal cipher digital audio yang terhubung ke modul transmiter. Dimana pin yang digunakan adalah pin 14 (PD_0), pin 15 (PD_1), pin 16 (PD_2), pin 17 (PD_3), pin 18 (PD_4), pin 19 (PD_5), pin 20 (PD_6), pin 21 (PD_7).

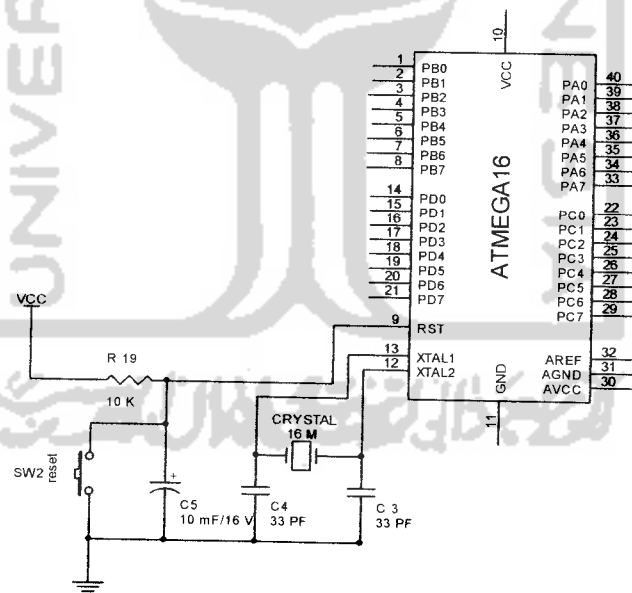


Gambar 3.6. Rangkaian output data

Port Osilator dan Reset Mikrokontroler ATmega16

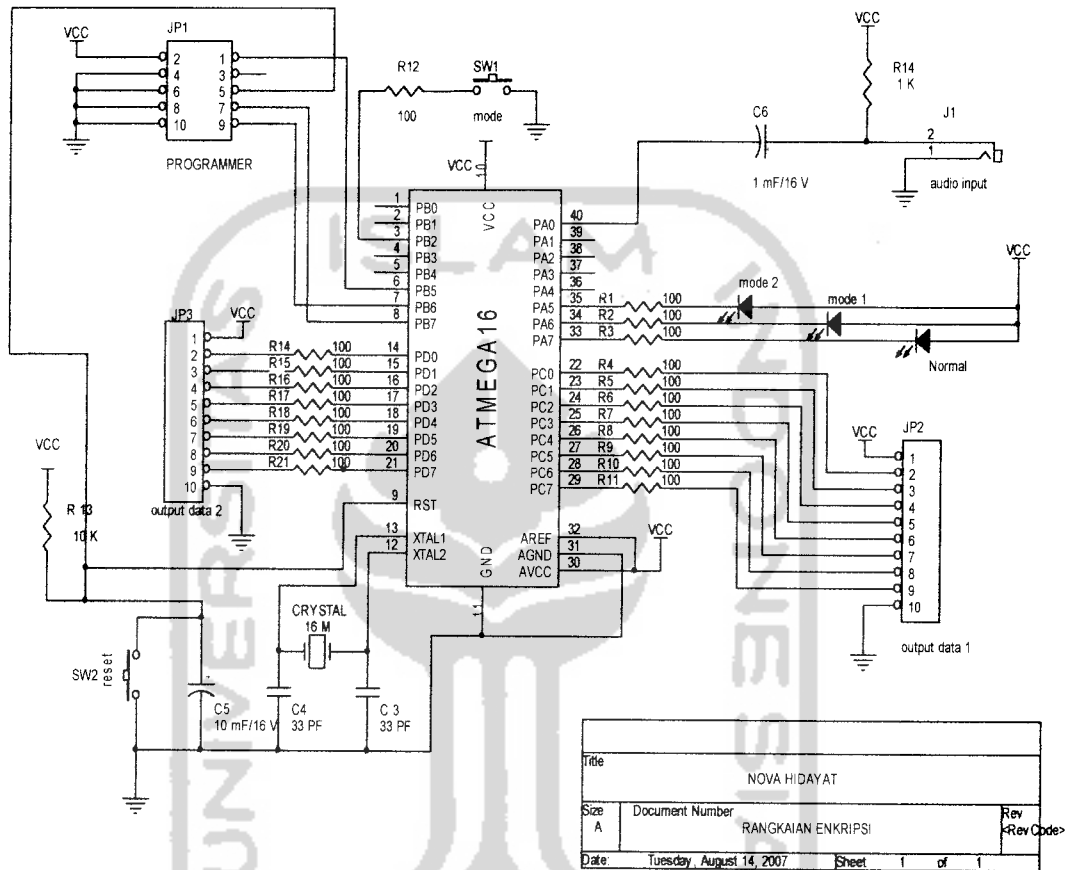
Pada rangkaian *osilator* ini menggunakan pin 12 (XTAL1) dan pin 13 (XTAL2) diparalel, digunakan kristal 16 MHz seperti ditunjukkan Gambar 3.7. Menurut *datasheet* kristal yang bisa digunakan untuk mikrokontroler ATmega16 adalah 0 – 33 MHz dan dua kapasitor 30 ± 10 pF.

Pin 9 (RST) merupakan rangkaian *reset*, yang mana digunakan untuk menghentikan kerja mikrokontroler dengan kembali ke alamat 0000/reset. Rangkaian *reset* dapat dilihat pada Gambar 3.7. Untuk *mereset* mikrokontroler ATmega16 yaitu dengan memberikan logika tinggi pada pin *reset* (pin 9) mikrokontroler ATmega16, logika tinggi ini dibuat minimal dua *machine cycles* (24 *Oscilator periode*). Jika menggunakan XTAL 16 MHz maka logika tinggi minimal selama 2 μ s.



Gambar 3.7. Rangkaian osilator dan reset

Jika seluruh rangkaian rancangan jika dihubungkan akan ditunjukkan pada gambar dibawah ini.



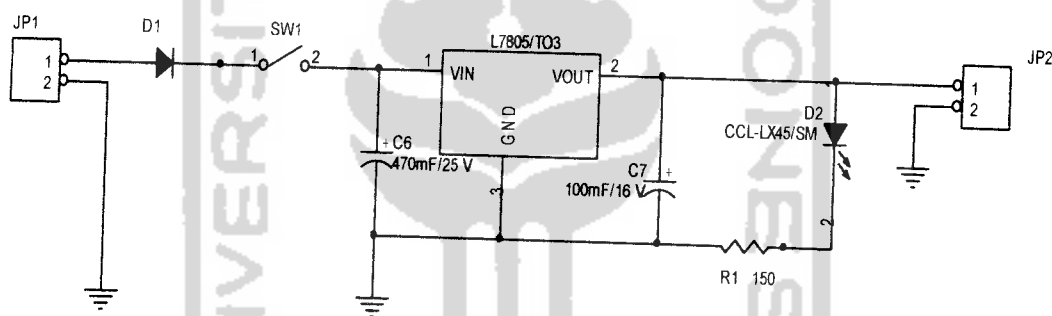
Gambar 3.8. Rangkaian lengkap skematik

3.2.2 Rangkaian power suplai

Rangkaian power supply merupakan bagian yang sangat penting pada rangkaian karena tanpa power suplai alat ini tidak dapat bekerja. Alat ini memerlukan catu daya yang dapat memberikan tegangan sebesar 5 Volt DC, yang

mana tegangan 5 Volt DC digunakan oleh sistem minimum mikrokontroler ATMEga16.

Mempertahankan suatu level tegangan yang konstan sangat diperlukan dalam rangkaian catu daya ini, dengan demikian rangkaian catu daya menggunakan regulator tegangan (*voltage regulator*) yang berbentuk IC (*integrated circuit*) yang mengandung sejumlah rangkaian untuk tegangan referensi, alat pengontrol, penguat komparator, dan pelindung tegangan berlebih (*overload protection*). Pada alat ini pelindung tegangan berlebih digunakan regulator jenis positif regulator dengan tipe LM7805 untuk penstabil tegangan 5 Volt DC.



Gambar 3.9. Rangkaian power suplai.

3.2. Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak dibahas tentang perancangan enkripsi data dan implementasi diagram alir yang meliputi inisialisasi program. Bahasa pemrograman yang biasa digunakan adalah bahasa *assembly*, kemudian dikembangkan kompailer untuk bahasa tingkat tinggi. Untuk mikrokontroler

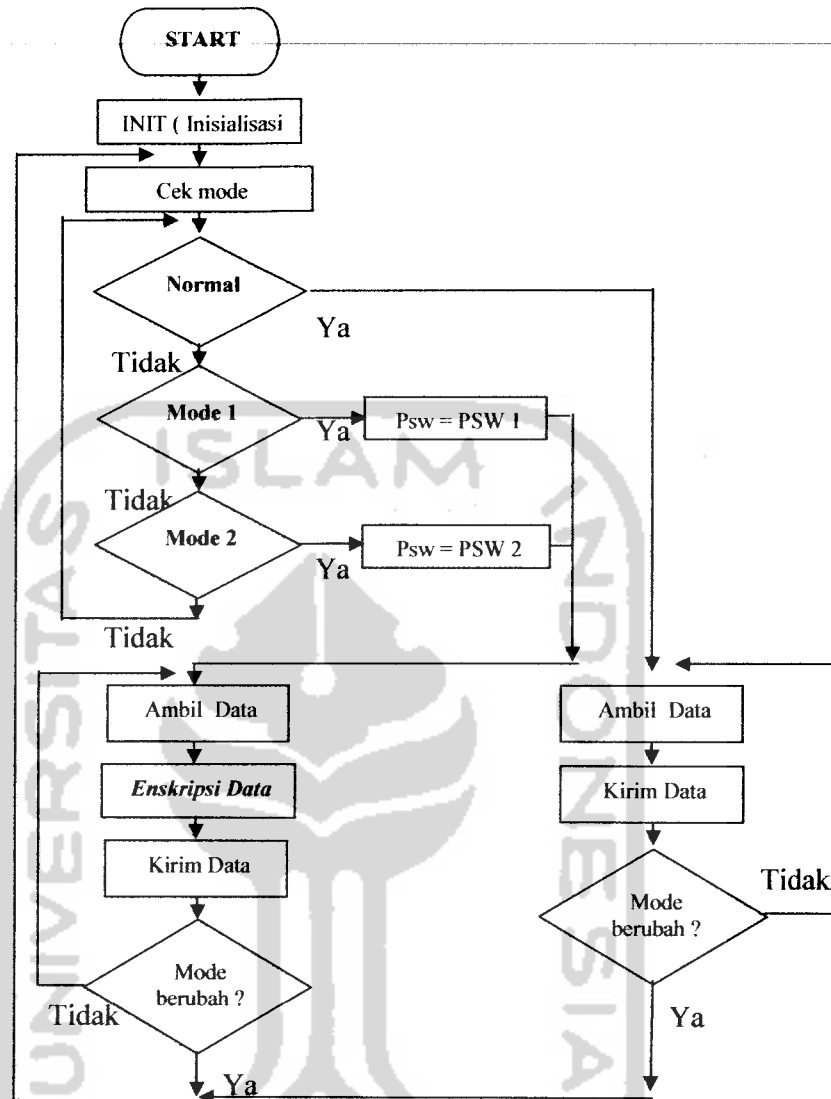
keluarga MCS-51 bahasa tingkat tinggi yang dikembangkan antara lain, *Basic*, *Pascal*, dan *C*. Bahasa *C* paling banyak dikembangkan semisal *Keil compiler* dan *Small Device C Compiler (SDCC)*. Pemrograman yang digunakan dalam rangkaian ini menggunakan bahasa *C*. Software yang digunakan *code vision*.

Dalam perancangan metode enkripsi ini, data input yang berupa sinyal analog dirubah menjadi data digital oleh mikrokontroler *ATMega16* yang memiliki ADC internal dalam bilangan biner 8-bit. Jadi setiap data tersebut menghasilkan suatu informasi.

Dalam perancangan ini dirancang tiga keadaan atau tiga mode yaitu normal, mode enkripsi 1 dan mode enkripsi 2 yang mana setiap mode pengacakan dengan tingkat peng-enskripsian yang berbeda untuk menghasilkan enkripsi data yang diinginkan yaitu data keluaran yang teracak dan terjaga keamanannya.

Perancangan program utama.

Program utama meliputi proses dari tiap rangkaian. Dan mekanismenya adalah yang pertama di mulai, program akan menginisialisasi data, kemudian dilanjutkan dengan pengecekan tombol atau mode, jika mode = kondisi normal maka data langsung diambil tanpa ada proses enkripsi, kemudian data langsung dikirim, jika tombol tidak ditekan atau tidak ada perubahan maka program akan berjalan terus - menerus pada mode normal. Berikut adalah diagram alir dari program utama..



Gambar. 3.10 Diagram alir program utama

Jika tombol ditekan maka program akan berhenti selama 50 ms, kemudian program berjalan lagi untuk langsung mengecek ke mode 1, selanjutnya mode 1 langsung mengambil data, kemudian data di proses secara enskripsi, setelah diproses data langsung dikirim, selanjutnya program akan berjalan terus menerus hingga mode 2 ditekan, apabila tombol ditekan maka program akan berhenti

selama 50 ms, kemudian program berjalan lagi untuk mengecek ke mode 2, selanjutnya mode 2 langsung mengamibil data kemudian data langsung diproses secara enkripsi, kemudian langsung dikirim. Program akan terus-menerus mengalami pengulangan.

Perancangan program enkripsi mode 1

Pada sub rutin enkripsi, program pertama kali akan menginisialisasi keluaran dari ADC sebagai input, selanjutnya program akan mengecek apakah input tidak sama dengan input sebelumnya, jika input sama dengan input sebelumnya maka data tidak diproses sebaliknya jika input tidak sama dengan input sebelumnya maka data akan langsung diproses, kemudian setelah data diproses maka data akan dikirim di port D, sehingga input sebelumnya sama dengan input, selanjutnya program akan terus-menerus mengalami pengulangan.

Perancangan enkripsi yang digunakan sebagai mode 1 atau password 1 ini menggunakan sebagai berikut.

$$\text{Chiper 1} = I(n) \oplus Y(n-1) \quad ; \text{untuk password 1} \dots \dots \dots (3.1)$$

Dimana :

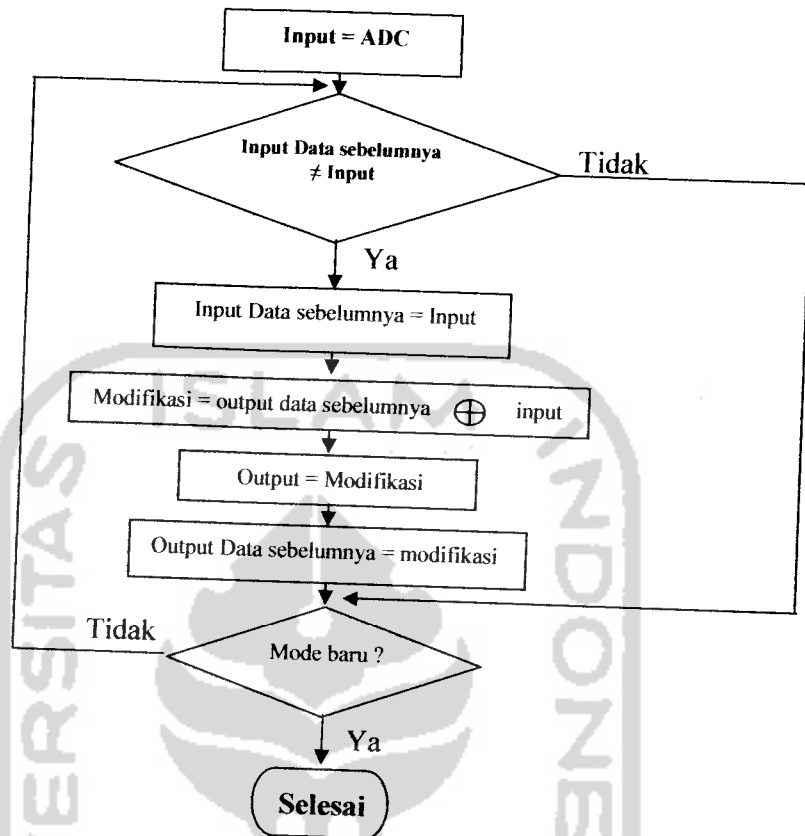
Chiper 1 = Data keluaran

I (n) = Data masukan

Y(n-1) = Data keluaran n -1

Dalam perancangan mode 1 ini input data yang masuk di XOR dengan data keluaran 1 kali sebelumnya.

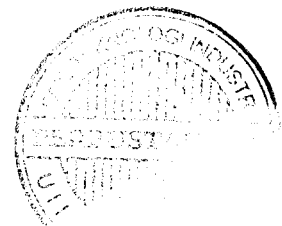
Berikut adalah diagram alir dari program enkripsi mode 1.



Gambar.3.11 Diagram alir enkripsi mode 1

Perancangan program enkripsi mode 2

Pada proses enkripsi, program pertama kali akan menginisialisasi masukan dari ADC sebagai input, selanjutnya program akan mengecek apakah input tidak sama dengan input sebelumnya jika input sama dengan input sebelumnya maka data tidak diproses. Jika sebaliknya input tidak sama dengan input sebelumnya maka data akan langsung diproses.



Perancangan enkripsi yang digunakan sebagai mode 2 atau password 2 ini menggunakan sebagai berikut:

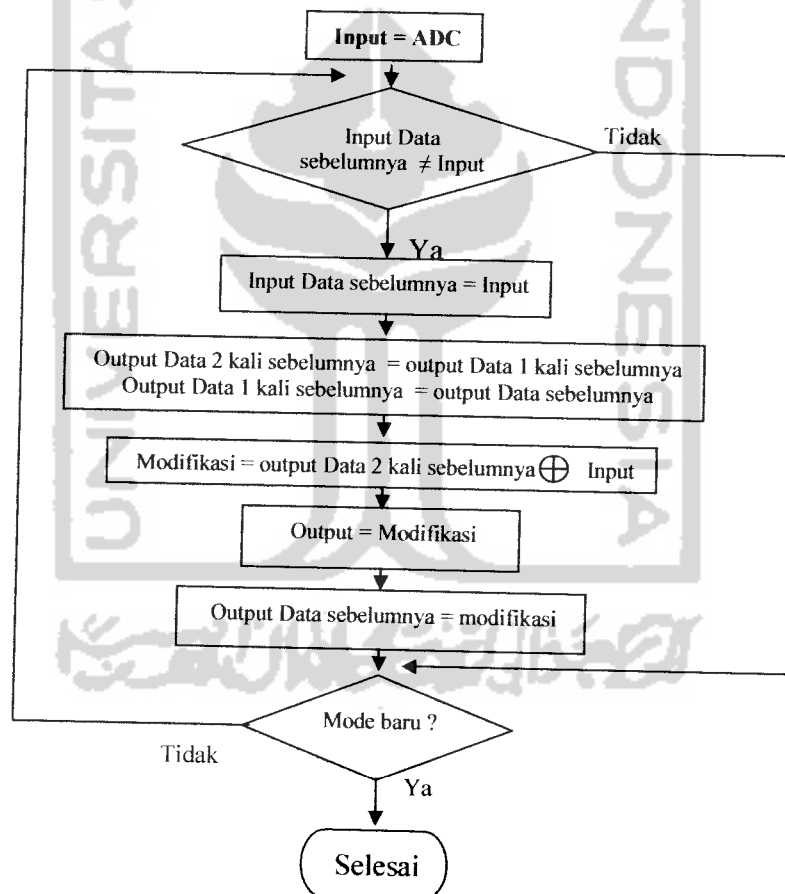
$$\text{Cipher 2} = I(n) \oplus Y(n-2) \quad ; \text{untuk password 2} \dots \dots \dots (3.2)$$

Dimana : Cipher 2 = Data keluaran

 I (n) = Data masukan

 Y(n - 2) = Data keluaran n - 2

Dalam perancangan mode 2 ini input data yang masuk di XOR dengan data masukan 2 kali sebelumnya. Berikut adalah diagram alir dari program enkripsi mode 2..



Gambar 3.12. Diagram alir enkripsi mode 2

BAB.IV

ANALISA DAN PEMBAHASAN

Data hasil pengujian didapatkan dengan melakukan pengujian langsung pada tiap bagian rangkaian dan sistem secara keseluruhan. Pengujian dilakukan berdasarkan fungsi operasional dan beberapa parameter yang ingin diketahui dari sistem tersebut. Data yang diperoleh dari pengujian tersebut dapat memberikan informasi yang cukup untuk keperluan penyempurnaan sistem dan sekaligus digunakan dalam pengembangan lebih lanjut. Pada pengujian alat ini dilakukan dengan menggunakan alat bantu, yaitu osiloskop, AFG (*Audio Frekuensi Generator*), generator variabel dan multimeter.

Berikut ini adalah proses pengujian dari rangkaian pembagian yang meliputi rangkaian mikrokontroler dan rangkaian power suplai.

4.1 Pengujian Rangkaian Mikrokontroler

Mikrokontroler merupakan pemproses utama dalam perancangan alat ini. Pengujian rangkaian mikrokontroler dilakukan dengan cara memberikan logika pada keempat port masukan dan keluarannya, kemudian membaca kondisi logika pada masing-masing port. Dari hasil pengujian, pada semua port mempunyai logika sesuai dengan yang diprogram, sehingga rangkaian mikrokontroler ini dapat digunakan sebagai pemproses utama.

4.1.1 Pengujian tegangan input dan output ADC

Pada pengujian rangkaian ADC ini pengujianya dilakukan dengan cara memberikan tegangan masukan yang berbeda dengan menggunakan adaptor variabel dan mengamati hasil keluarannya.

Tabel 4.1 Pengujian tegangan input dan output ADC

Input tegangan (volt)	Output data (Kondisi LED)								Output bilangan Heksa
	D7	D6	D5	D4	D3	D2	D1	D0	
5,0	1	1	1	1	1	1	1	1	FF
4,8	1	1	1	1	0	1	0	1	F5
4,6	1	1	1	0	1	1	0	0	EC
4,4	1	1	1	0	0	0	1	0	E2
4,2	1	1	0	1	0	1	1	1	D7
4,0	1	1	0	0	1	1	0	1	CD
3,8	1	1	0	0	0	0	1	1	C3
3,6	1	0	1	1	1	0	0	1	B9
3,4	1	0	1	0	1	1	1	0	AE
3,2	1	0	1	0	0	1	0	1	A5
3,0	1	0	0	1	1	0	1	0	9A
2,8	1	0	0	1	0	0	0	1	91
2,6	1	0	0	0	0	1	1	0	88
2,4	0	1	1	1	1	0	1	1	7B
2,2	0	1	1	1	0	0	0	1	71
2,0	0	1	1	0	0	1	1	0	66
1,8	0	1	0	1	1	1	0	0	59
1,6	0	1	0	1	0	0	1	0	52
1,4	0	1	0	0	1	0	0	0	48
1,2	0	0	1	1	1	1	1	0	3E
1,0	0	0	1	1	0	0	1	1	33
0,8	0	0	1	0	1	0	0	1	29
0,6	0	0	0	1	1	1	1	1	1F
0,4	0	0	0	1	0	1	0	0	14
0,2	0	0	0	0	1	0	0	1	09
0	0	0	0	0	0	0	0	0	00

Hasil keluaran dari pengujian sistem ADC adalah kondisi LED, dimana jika "0" mewakili kondisi menyala sebaliknya jika "1" mewakili kondisi mati. Jangkauan

tegangan masukan untuk sistem ADC yaitu antara 0 – 5 volt, dengan skala perubahan kenaikan tegangan adalah 0,2 volt. Dari pengujian dapat dilihat bahwa nilai output tertinggi adalah FF dalam bilangan heksa. Dalam 26 langkah tersebut terdapat 26 keadaan yang berbeda setiap outputnya berupa kondisi LED.

Pada perancangan ini menggunakan ADC 8 bit karena pertimbangan kecepatan. Jadi resolusi ADC = $\frac{\text{Tegangan referensi}}{2^{n \text{ bit}}}$

$$= \frac{5}{256} = 0.0195313 \text{ Volt} \approx 0.02 \text{ Volt}$$

Jadi resolusi perubahan data per bit ADC adalah 0.02 volt. Jadi dengan tegangan referensi 5 volt dan resolusi data 0.02 volt diperoleh 250 perubahan.

4.1.2 Pengujian metode enkripsi

Dalam perancangan enkripsi sinyal audio yang dilakukan menggunakan metode gerbang digital XOR. Setiap mode memiliki password yang berbeda dan menghasilkan output yang berbeda pula. Sinyal audio yang berupa data digital dengan bilang biner akan lebih mudah untuk pengujian. Sistem enkripsi ini yaitu merubah data-data digital yang masuk secara langsung atau realtime terus menerus. Perubahan sinyal audio yang ter-enkripsi akan menyebabkan sinyal audio tersebut tidak bisa dimengerti. Perubahan itu juga terlihat jelas pada bentuk sinyal yang terlihat di osiloskop bahwa sinyal tersebut pecah, acak dan tidak berbentuk suatu sinyal yang jelas.

4.2 Pengujian Power Suplai

Tegangan DC untuk sumber daya bagi seluruh rangkaian yaitu ATmega16 berasal dari baterai 9 Volt.

Hasil pengukuran pada semua komponen setelah keluaran dari pin 3 LM7805 adalah 4.99 Volt.

4.3 Pengujian Sistem Keseluruhan

Pengujian sistem keseluruhan adalah pengujian alat enkripsi simetrik sinyal audio yang merupakan integrasi dari seluruh rangkaian yang dihubungkan sesuai dengan fungsinya. Pengujian sistem keseluruhan dilakukan dengan mengoperasikan mikrokontroler ATMegal6. Pada pengujian ini mengamati apakah alat dapat beroperasi dengan baik, yaitu apakah alat dapat menerima input, memproses dan menghasilkan baik berupa bentuk sinyal ataupun berupa output audio normal dan terenskripsi. Proses pengujian ini terjawab dengan cara proses mengoperasikan dan memberikan input pada alat berupa sinyal dengan frekuensi tetap dan sinyal audio analog.

4.3.1 Pengujian sinyal

Pengujian ini dilakukan untuk mengetahui bentuk sinyal masukan dan keluaran, dalam arti bentuk kedua sinyal harus sama, karena sinyal masih dalam keadaan murni dan belum mengalami proses enkripsi, dalam pengujian ini menggunakan AFG (Audio Frekuensi Generator) sebagai input yang besarnya antara frekuensi 1000 Hz, 2000 Hz, 3000 Hz dan keluaran akan terlihat pada osiloskop. Dalam pengujian ini jika sinyal masukan tidak mengalami proses enkripsi maka sinyal keluaran akan terlihat mirip seperti bentuk sinyal aslinya yaitu bentuk sinyal sinus tetapi dalam hal ini sinyal keluarannya akan terlihat berupa titik sampling.

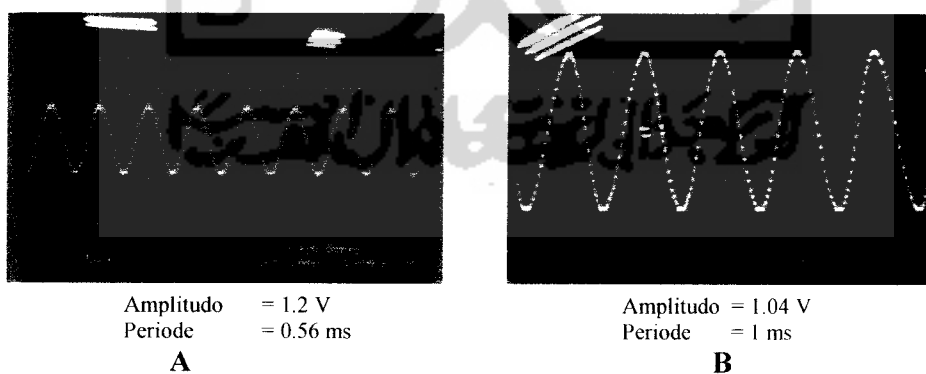
Titik titik sampling tersebut terjadi karena terjadinya proses mengubah amplitudo gelombang bunyi ke dalam waktu interval tertentu (disebut juga sampling), sehingga menghasilkan representasi digital dari suara. Sinyal dirubah dan diproses sehingga menjadi bit-bit digital. Proses tersebut dilakukan oleh ADC internal mikrokontroller ATmega16. Jadi setiap titik sampling tersebut mewakili dan membawa suatu data atau informasi. Semakin rendah frekuensi berarti semakin banyak titik samplingnya dan semakin tinggi frekuensinya, semakin sedikit samplingnya. Sehingga pada pengujian ini kita akan mengamati titik sampling per 1 gelombangnya.

4.3.2 Pengujian sinyal normal dan enkripsi

Pengujian ini dilakukan untuk mengamati sinyal normal dan setelah di enkripsi, dengan menggunakan AFG (Audio Frekuensi Generator) sebagai input yang besarnya antara 1000 Hz – 3000 Hz dan input suara asli manusia.

4.3.2.1 Pengujian input sinyal normal dengan mode normal.

Pada pengujian ini dilakukan dengan menggunakan sinyal masukan antara 1000 Hz sampai 3000 Hz.



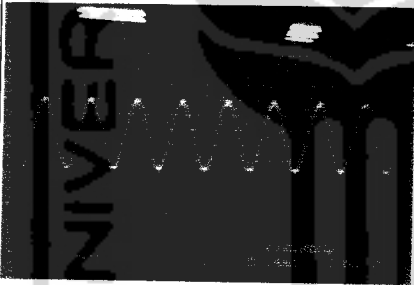
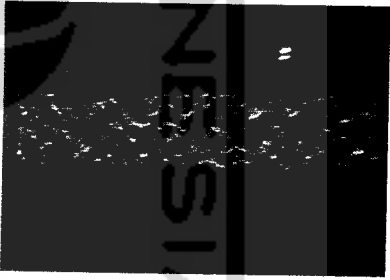

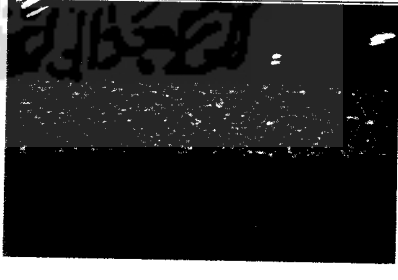
Gambar.4.1 (A) Kondisi input sinyal dengan frekuensi 1000 Hz

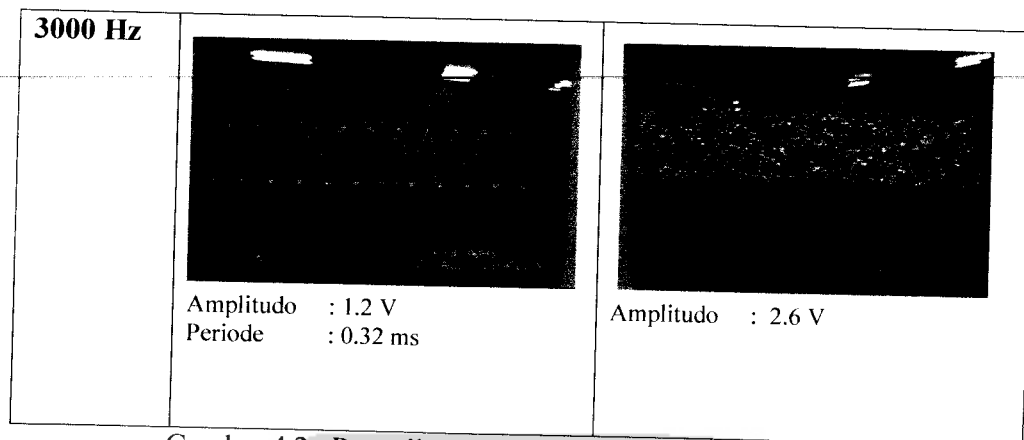
(B) Kondisi sinyal output dengan mode normal

Pada gambar diatas sinyal tampak jelas, karena pada proses ini sinyal tidak mengalami enkripsi maka bentuk sinyal keluaran mirip dengan sinyal masukan aslinya. Pada pengujian ini terlihat besarnya titik sampling per 1 gelombang .

4.3.2.2 Pengujian input sinyal normal dengan mode 1

Dalam pengujian ini input sinyal berupa frekuensi 1000 Hz, 2000 Hz, 3000 Hz. Kemudian sinyal ini di enkripsi sehingga hasil keluaranya akan terlihat pada gambar 4.2 berikut :

Frekuensi	Kondisi Normal	Enskripsi mode 1
1000 Hz	 <p>Amplitudo : 1.2 V Periode : 0.56 ms</p>	 <p>Amplitudo : 2.6 V</p>
2000 Hz	 <p>Amplitudo : 1.2 V Periode : 0.4 ms</p>	 <p>Amplitudo : 2.6 V</p>



Gambar.4.2 Pengujian input sinyal normal dengan mode 1

Pada gambar diatas terlihat sinyal keluaran tampak acak, karena sinyal telah di enkripsi, sehingga untuk menghitung titik samplingnya harus mendekripsikan sinyal tersebut agar bisa menjadi bentuk sinyal aslinya.

Untuk mendekripsikan sinyal tersebut tentunya harus mengetahui kode enkripsinya.

Dengan Chiper 1 = $I(n) \oplus Y(n-1)$; untuk password 1

Dimana :

Chiper 1 = Data keluaran

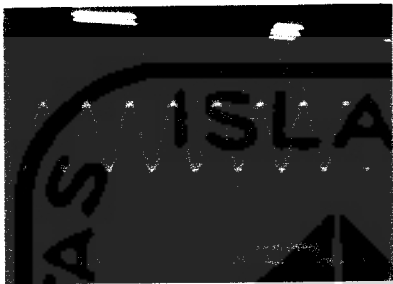
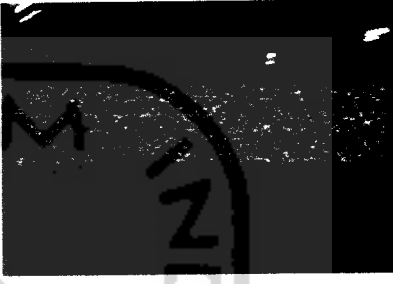
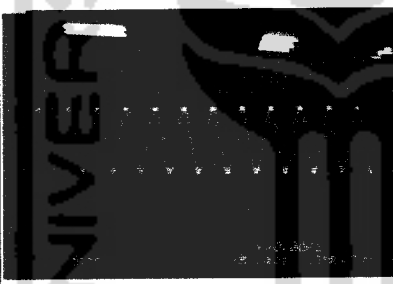



I (n) = Data masukan

Y(n-1) = Data keluaran n-1

Jadi setiap data yang masuk yaitu berupa sampling-sampling, yang berisi data-data digital berupa bit-bit tersebut langsung diproses dengan di Xor dengan data sebelumnya. Sehingga nilai-nilai data-data tersebut berubah dan menyebabkan perubahan pada sinyalnya dan suaranya.

4.3.2.3 Pengujian input sinyal normal dengan mode 2.

Pada pengujian ini input sinyal masukan berupa frekuensi 1000Hz, 2000 Hz, 3000 Hz . Kemudian sinyal ini dienkripsi dengan mode 2, setelah di enkripsi sinyal akan terlihat pada gambar 4.3 berikut :

Frekuensi	Kondisi Normal	Enskripsi mode 2
1000 Hz	 Amplitudo : 1.2 V Periode : 0.56 ms	 Amplitudo : 2.6 V
2000 Hz	 Amplitudo : 1.2 V Periode : 0.4 ms	 Amplitudo : 2.6 V
3000 Hz	 Amplitudo : 1.2 V Periode : 0.32 ms	 Amplitudo : 2.6 V

Gambar 4.3. Pengujian input sinyal normal dengan mode 2.

Pada gambar diatas terlihat sinyal keluaran menjadi acak, ini disebabkan karena sinyal telah dienkripsi, disamping itu juga terlihat perbedaan antara enkripsi dengan mode 1 dan mode 2, perbedaan tersebut terletak pada bentuk sinyal mode 2 yang semakin acak .

Dengan Chiper 2 = $I(n) \oplus Y(n-2)$; untuk password 2

Dimana :

Chiper 2 = Data keluaran

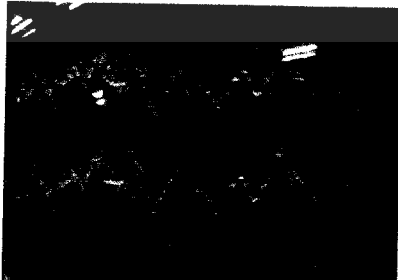
$I(n)$ = Data masukan



$Y(n-2)$ = Data keluaran n-2

Jadi setiap data yang masuk yaitu berupa sampling-sampling, yang berisi data-data digital berupa bit-bit tersebut langsung diproses dengan di Xor dengan data 2 kali sebelumnya. Sehingga nilai-nilai data-data tersebut berubah dan menyebabkan perubahan pada sinyalnya dan suaranya.

4.3.2.4 Pengujian input sinyal audio dengan mode

Pada pengujian ini input sinyal masukan berupa frekuensi audio. Kemudian sinyal ini dienkripsi dengan kondisi normal, mode 1 dan mode 2, setelah di enkripsi sinyal akan terlihat pada gambar 4.4 berikut :

Kondisi	Gambar	Keterangan
Normal		Input Amplitudo : 1.2 V Periode : 2 ms Output amplitudo : 1.2 V

Mode 1		Input Amplitudo : 1.4 V Time / Div : 2 ms Output Amplitudo : 2.6 V
Mode 2		Input Amplitudo : 1.2 V Time / Div : 2 ms Output Amplitudo : 2.4 V

Gambar 4.4. Pengujian input sinyal audio dengan mode

Pada gambar diatas terlihat bawa sinyal input yang berupa analog terlihat jelas bentuk sinyalnya dan sinyal keluaran menjadi acak, ini disebabkan karena sinyal telah dienkripsi, disamping itu juga terlihat perbedaan antara enkripsi dengan mode 1 dan mode 2, perbedaan tersebut terletak pada bentuk sinyal mode 2 yang semakin acak.

BAB.V

PENUTUP

5.1. Kesimpulan

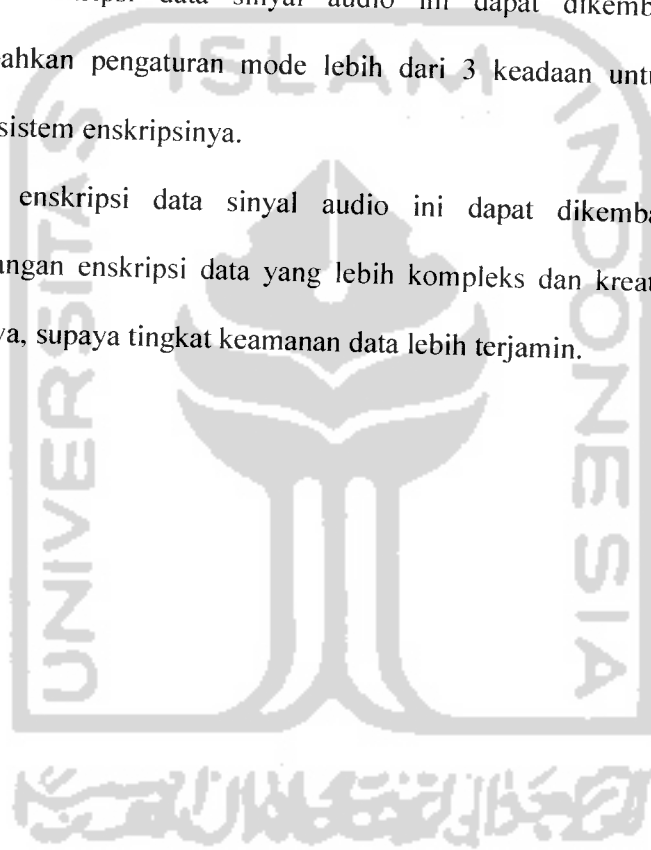
Berdasarkan dari proses perancangan, pembuatan, pengamatan dan hasil uji sistem yang dibuat, maka dapat diambil beberapa kesimpulan dan saran untuk kemajuan, perbaikan dan pengembangan dari aplikasi sistem sebagai berikut :

1. Unjuk kerja dari enkripsi data pada sinyal audio berbasis mikrokontroler ATmega16 ini dapat mengamankan data yang berupa sinyal audio berdasarkan mode yang dikehendaki.
2. Perancangan sistem dari enkripsi data audio ini menggunakan tiga mode, yaitu: Mode normal, Mode 1 (Enkripsi 1), Mode 2 (Enkripsi 2).
3. Untuk perancangan ini, proses enkripsi data memerlukan kecepatan dalam memproses data masuk dan mengirimkan hasil enkripsinya secara *real time*.
4. Perubahan dari analog ke digital dengan ADC merupakan mengubah amplitudo gelombang bunyi ke dalam interval waktu tertentu, sehingga menghasilkan representasi digital dari suara yaitu berupa sinyal yang tersampling.
5. Frekuensi sampling (f_s) pada perancangan ini adalah minimal 40 KHz, hal ini karena $f_s \geq 2 f$ maksimal dengan frekuensi maksimal 20 KHz.

5.2. Saran

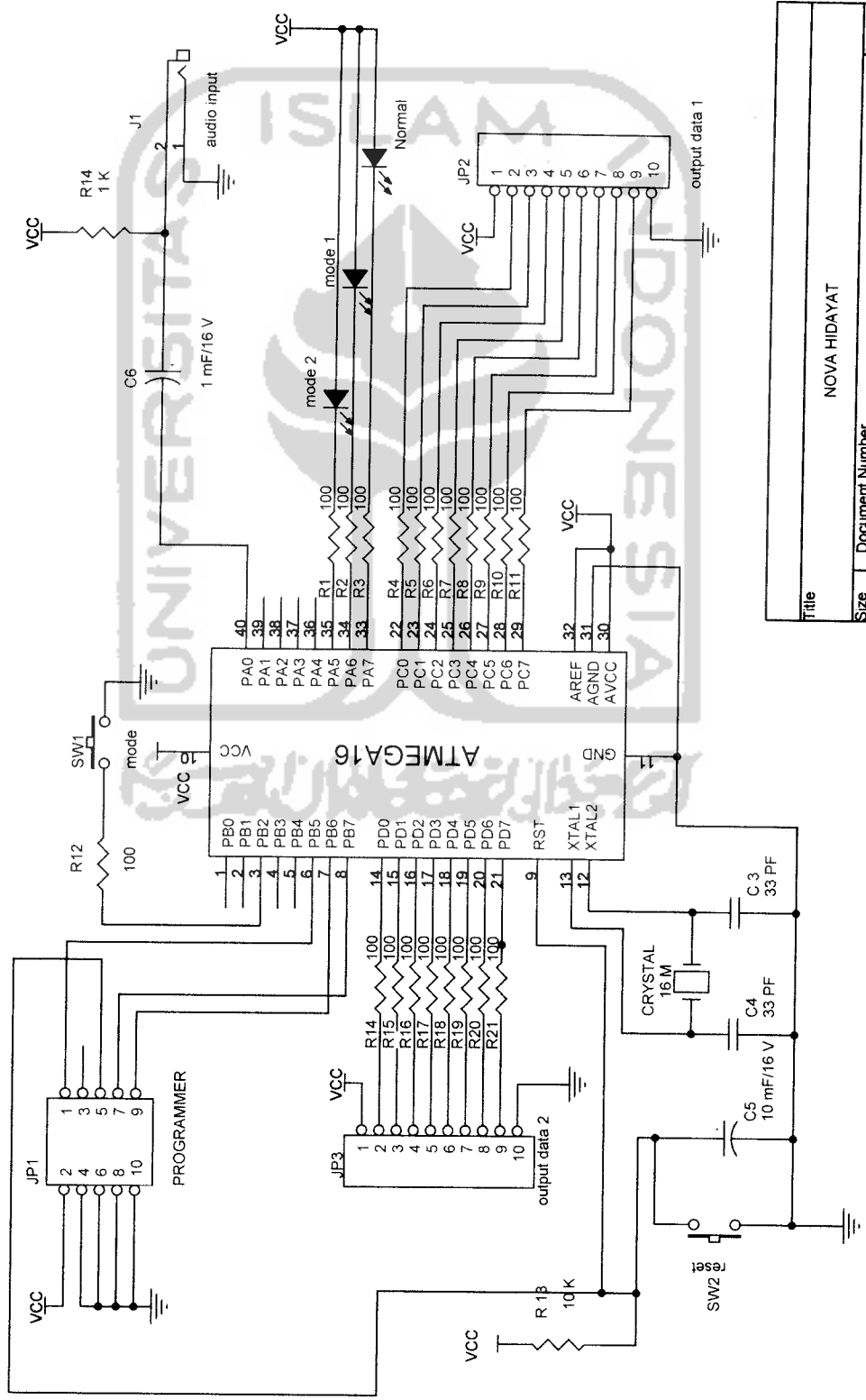
Untuk pengembangan enkripsi data sinyal audio selanjutnya, penulis menyarankan:

1. Dalam merancang hardware, alat ukur dan perangkat pengamatan yang digunakan harus dipilih yang mempunyai kualitas bagus untuk memperkecil kesalahan.
2. Sistem enkripsi data sinyal audio ini dapat dikembangkan dengan ditambahkan pengaturan mode lebih dari 3 keadaan untuk lebih variasi dalam sistem enkripsinya.
3. Sistem enkripsi data sinyal audio ini dapat dikembangkan dengan perancangan enkripsi data yang lebih kompleks dan kreatif dalam setiap modenya, supaya tingkat keamanan data lebih terjamin.

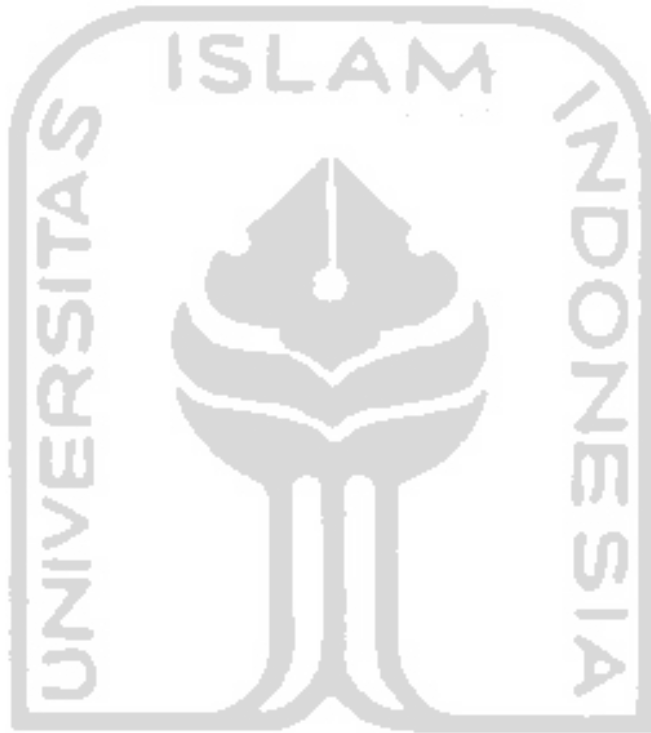
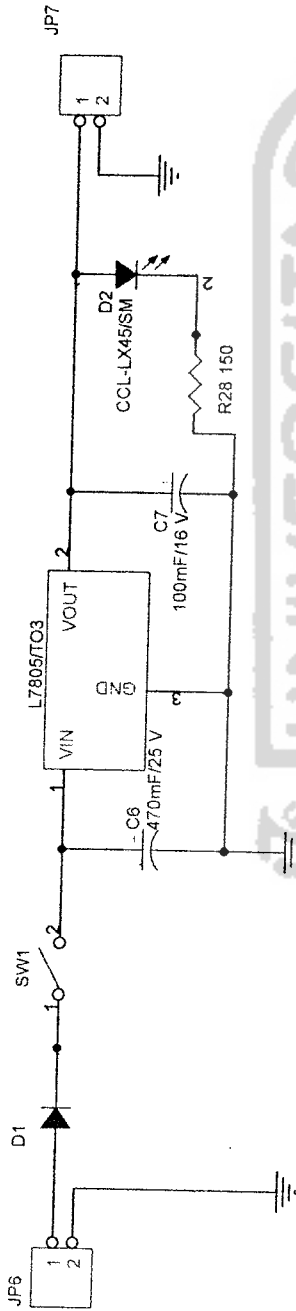


DAFTAR PUSTAKA

- Andri, Kristanto, 2003. *Keamanan Data pada Jaringan Komputer*. Yogyakarta. Penerbit : Gava Media.
- Data sheet available ATmega16. www.atmel.com/
- Proakis, J.G., dan Manolakis, D. G. 1997. *Pemrosesan Sinyal Digital : Prinsip, Algoritma dan Aplikasi*. Jakarta . Penerbit : PT.Prenhallindo.
- Rudyanto ,Arif, 2006. *Modul Belajar Keamanan komputer*. Yogyakarta . STIMIK AMIKOM Yogyakarta.
- Teguh Wahyono. 2003. *Prinsip Dasar dan Teknologi Komunikasi Data*. Graha Ilmu. Yogyakarta
- Tokheim, Roger, L, 1995. *Elektroika Digital* . Jakarta. Penerbit : Erlangga
- Wahyono, T, 2003. *Prinsip Dasar dan Teknologi Komunikasi Data*. Yogyakarta. Penerbit Graha Ilmu.
- Wardhana, Lingga., 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware dan Aplikasi*, Yogyakarta : Penerbit Andi.
- Woollard, Barry, G, 1999. *Elektronika Praktis*. Jakarta. Penerbit : PT Pradnya Paramita.



Title		NOVA HIDAYAT	
Size	Document Number	RANGKAIAN ENKRIPSI	
A			
Date:	Tuesday, August 14, 2007	Sheet	1 of 1
			2



Title		NOVA HIDAYAT	
Size	Document Number	Rev	<RevCode
A	RANGKAIAN CATU DAYA		
Date:	Wednesday, February 28, 2007	Sheet	1 of 1

```
/******
```

```
Program      : Audio Encrytion
Version      : 23
Date         : 6 / 14 / 2007
Programmer   : Nova Hidayat
NIM          : 02 524 051
University   : Univ. Islam Indonesia

Mikrokontroler : ATmega 16
X-TAL         : 16.000000 MHz
```

```
*****/
```

```
#include <mega16.h>
#include <delay.h>

#define Normal 0

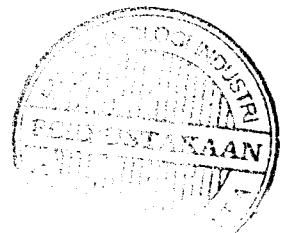
unsigned char OutputBefore, OutputBefore1, OutputBefore2, InputBefore;
unsigned char Modify, Mode, Input;
bit ModeBaru;

//External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
    ModeBaru = 1;
}

#define ADC_VREF_TYPE 0x60

// Setting ADC
Unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

void main(void)
{
```



```
// Inisialisasi Input/Output Ports
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=In Func3=In Func2=In
  Func1=In  Func0=In
// State7=0 State6=0 State5=0 State4=T State3=T State2=T State1=T
  State0=T
PORTA=0x00;
DDRA=0xE0;

// Inisialisasi Port B
// Semua sebagai masukan
// State7=T State6=T State5=T State4=T State3=T State2=P State1=P
  State0=T
PORTB=0x06;
DDRB=0x00;

// Inisialisasi Port C
// Semua sebagai Output
// Nilai Awal 0
PORTC=0xFF;
DDRC=0xFF;

// Inisialisasi Port D
// Semua sebagai Output
// Semua diberi data awalan 0
PORTD=0x00;
DDRD=0xFF;

// Inisialisasi Timer/Counter 0
// Clock source: System Clock
// Clock value: Time 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Inisialisasi Timer/Counter 1
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
```

```
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Inisialisasi Timer/Counter 2
// Clock source: System clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Inisialisasi External Interrupt(s)
// INT0: Off
// INT1: Off
// INT2: On
// INT2 Mode: Rising Edge
GICR|=0x20;
MCUCR=0x00;
MCUCSR=0x40;
GIFR=0x20;

// Inisialisasi Timer(s)/Counter(s) Interrupt(s)
TIMSK=0x00;

// Inisialisasi Analog Comparator
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// Inisialisasi ADC
// ADC Clock frequency: 1.000000 KHz
// ADC Voltage Reference: AVCC pin
```

```
// ADC Auto Trigger Source: None
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x84;
```

```
// Global enable interrupts
#asm("sei")
Mode= Normal;
while (1)
{
    // Pengecekan Mode
    if(ModeBaru == 1){
        While(1){
            delay_ms(50);
            Mode=Mode + 1;
            If( PINB.2 ==1) Break;
        }
    }
    if (Mode == 3) Mode = Normal;
    ModeBaru = 0;
```

```
    //Penampilan Mode
    if(Mode == Normal) {
        PORTA.7 = 0; // On;
        PORTA.6 = 1; // Off;
        PORTA.5 = 1; // Off;
    }
    else if(Mode == 1) {
        PORTA.7 = 1; // Off;
        PORTA.6 = 0; // On;
        PORTA.5 = 1; // Off;
    }
    else if(Mode == 2) {
        PORTA.7 = 1; //Off;
        PORTA.6 = 0; //On;
        PORTA.5 = 0; //On;
    }
}
```

```
// Proses Enskripsi Data
```

```
If(Mode == 1){
    // OutputBefore/Password = 01;
    While(1){
        Input = read_adc(0);
        //Input = Input ^ 0xAA;
    }
}
```



```
// CodeVisionAVR C Compiler
```

```
// I/O registers definitions for the ATmega16
```

```
#ifndef _MEGA16_INCLUDED_  
#define _MEGA16_INCLUDED_
```

```
#pragma used+  
sfrb TWBR=0;  
sfrb TWSR=1;  
sfrb TWAR=2;  
sfrb TWDR=3;  
sfrb ADCL=4;  
sfrb ADCH=5;  
sfrw ADCW=4; // 16 bit access  
sfrb ADCSRA=6;  
sfrb ADMUX=7;  
sfrb ACSR=8;  
sfrb UBRRL=9;  
sfrb UCSRB=0xa;  
sfrb UCSRA=0xb;  
sfrb UDR=0xc;  
sfrb SPCR=0xd;  
sfrb SPSR=0xe;  
sfrb SPDR=0xf;  
sfrb PIND=0x10;  
sfrb DDRD=0x11;  
sfrb PORTD=0x12;  
sfrb PINC=0x13;  
sfrb DDRC=0x14;  
sfrb PORTC=0x15;  
sfrb PINB=0x16;  
sfrb DDRB=0x17;  
sfrb PORTB=0x18;  
sfrb PINA=0x19;  
sfrb DDRA=0x1a;  
sfrb PORTA=0x1b;  
sfrb EECR=0x1c;  
sfrb EEDR=0x1d;  
sfrb EEARL=0x1e;  
sfrb EEARH=0x1f;  
sfrw EEAR=0x1e; // 16 bit access  
sfrb UBRRH=0x20;
```

```

sfrb UCSRC=0x20;
sfrb WDTCSR=0x21;
sfrb ASSR=0x22;
sfrb OCR2=0x23;
sfrb TCNT2=0x24;
sfrb TCCR2=0x25;
sfrb ICR1L=0x26;
sfrb ICR1H=0x27;
sfrb OCR1BL=0x28;
sfrb OCR1BH=0x29;
sfrw OCR1B=0x28; // 16 bit access
sfrb OCR1AL=0x2a;
sfrb OCR1AH=0x2b;
sfrw OCR1A=0x2a; // 16 bit access
sfrb TCNT1L=0x2c;
sfrb TCNT1H=0x2d;
sfrw TCNT1=0x2c; // 16 bit access
sfrb TCCR1B=0x2e;
sfrb TCCR1A=0x2f;
sfrb SFIOR=0x30;
sfrb OSCCAL=0x31;
sfrb OCFR=0x31;
sfrb TCNT0=0x32;
sfrb TCCR0=0x33;
sfrb MCUCSR=0x34;
sfrb MCUCR=0x35;
sfrb TWCR=0x36;
sfrb SPMCR=0x37;
sfrb TIFR=0x38;
sfrb TIMSK=0x39;
sfrb GIFR=0x3a;
sfrb GICR=0x3b;
sfrb OCR0=0x3c;
sfrb SPL=0x3d;
sfrb SPH=0x3e;
sfrb SREG=0x3f;
#pragma used-

```

```
// Interrupt vectors definitions
```

```

#define EXT_INT0 2
#define EXT_INT1 3
#define TIM2_COMP 4

```



```

#define TIM2_OVF 5
#define TIM1_CAPT 6
#define TIM1_COMPA 7
#define TIM1_COMPB 8
#define TIM1_OVF 9
#define TIM0_OVF 10
#define SPI_STC 11
#define USART_RXC 12
#define USART_DRE 13
#define USART_TXC 14
#define ADC_INT 15
#define EE_RDY 16
#define ANA_COMP 17
#define TWI 18
#define EXT_INT2 19
#define TIM0_COMP 20
#define SPM_READY 21

// Needed by the power management functions (sleep.h)
#define __SLEEP_SUPPORTED__
#define __POWERDOWN_SUPPORTED__
#define __POWERSAVE_SUPPORTED__
#define __STANDBY_SUPPORTED__
#define __EXTENDED_STANDBY_SUPPORTED__
#asm
    #ifndef __SLEEP_DEFINED__
    #define __SLEEP_DEFINED__
    .EQU __se_bit=0x40
    .EQU __sm_mask=0xB0
    .EQU __sm_powerdown=0x20
    .EQU __sm_powersave=0x30
    .EQU __sm_standby=0xA0
    .EQU __sm_ext_standby=0xB0
    .EQU __sm_adc_noise_red=0x10
    .SET power_ctrl_reg=mcucr
    #endif
#endasm

#endif

```

```
// CodeVisionAVR C Compiler
```

```
#ifndef _DELAY_INCLUDED_  
#define _DELAY_INCLUDED_
```

```
#pragma used+
```

```
void delay_us(unsigned int n);  
void delay_ms(unsigned int n);
```

```
#pragma used-
```

```
#endif
```



Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 16K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 512 Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 1K Byte Internal SRAM
 - Programming Lock for Software Security
- JTAG (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
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels 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
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega16L
 - 4.5 - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16
- Power 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**

Summary

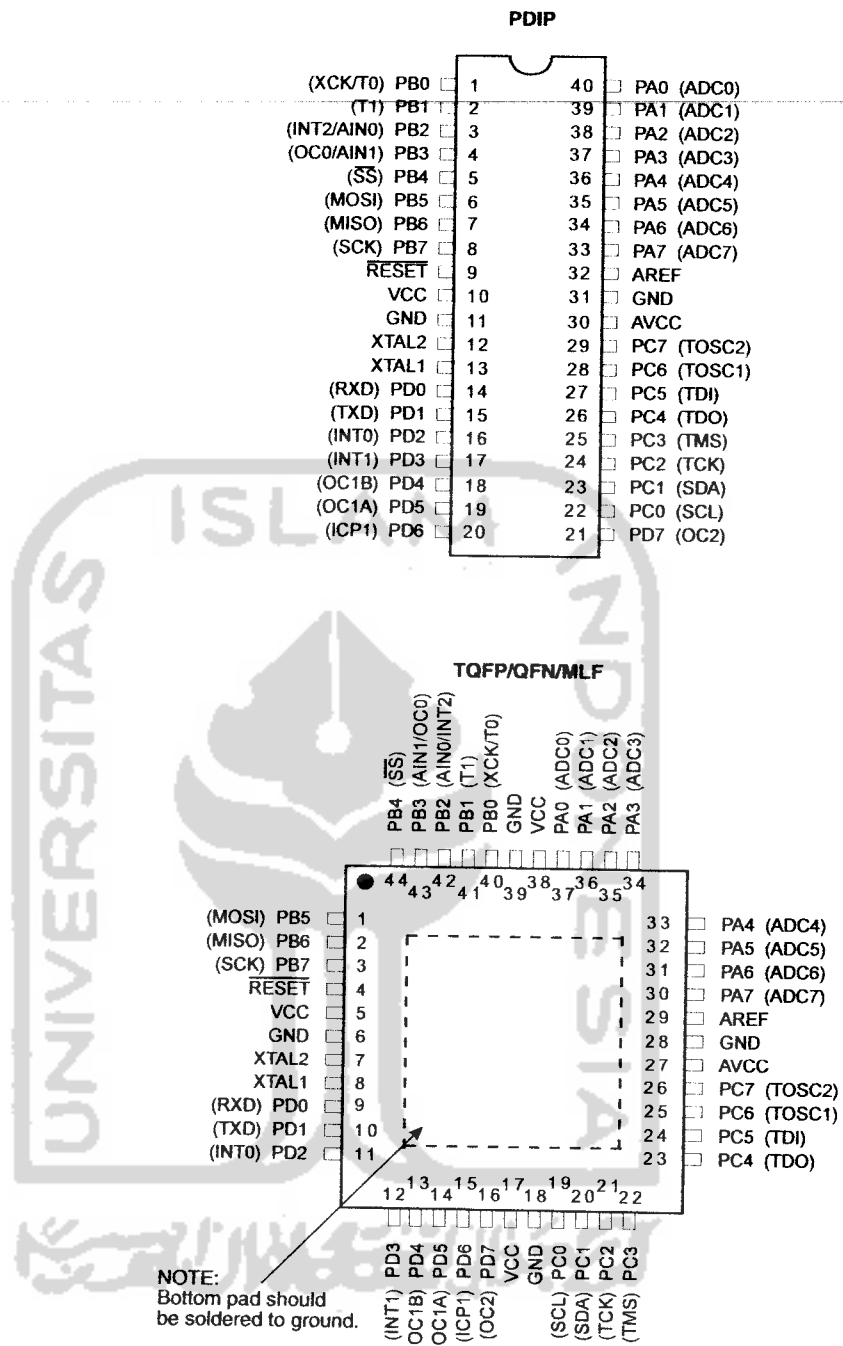
2466LS-AVR-06/05



Note: This is a summary document. A complete document is available on our Web site at www.atmel.com.

Pin Configurations

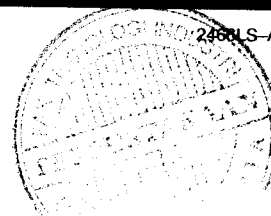
Figure 1. Pinout ATmega16



Disclaimer

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

ATmega16(L)

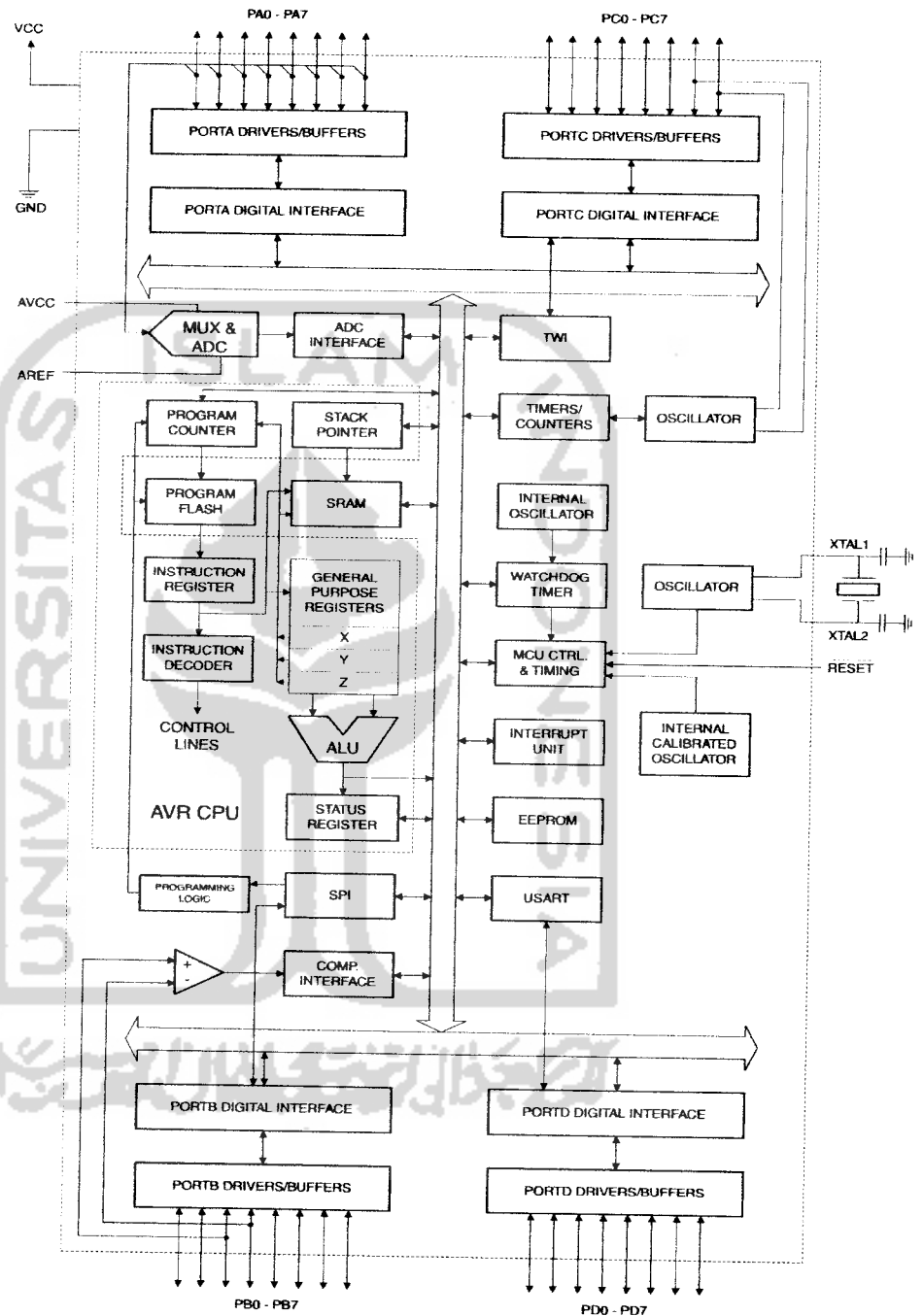


Overview

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

Block Diagram

Figure 2. Block Diagram





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

The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, 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.

Pin Descriptions

V_{CC}

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

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

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

ATmega16(L)

Port B (PB7..PB0)

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

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

Port C (PC7..PC0)

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

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

Port D (PD7..PD0)

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

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

RESET

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

XTAL1

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

XTAL2

Output from the inverting Oscillator amplifier.

VCC

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

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

Resources

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



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$3F (\$5F)	SREG	I	T	H	S	V	N	Z	C	7
\$3E (\$5E)	SPH	-	-	-	-	-	-	-	-	7
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP10	SP9	SP8	10
\$3C (\$5C)	OCRO	Timer/Counter0 Output Compare Register								10
\$3B (\$5B)	GICR	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	83
\$3A (\$5A)	GIFR	INTF1	INTF0	INTF2	-	-	-	-	-	46, 67
\$39 (\$59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	83, 114, 132
\$38 (\$58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	68
\$37 (\$57)	SPMCR	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	84, 115, 132
\$36 (\$56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWME	250
\$35 (\$55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	178
\$34 (\$54)	MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF	30, 66
\$33 (\$53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	39, 67, 229
\$32 (\$52)	TCNT0	Timer/Counter0 (8 Bits)								81
\$31 ⁽¹⁾ (\$51) ⁽¹⁾	OSCCAL	Oscillator Calibration Register								83
	OCDR	On-Chip Debug Register								28
\$30 (\$50)	SFIOR	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10	225
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	PSR2	PSR10	55, 86, 133, 199, 219
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	109
\$2D (\$4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								112
\$2C (\$4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								113
\$2B (\$4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								113
\$2A (\$4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								113
\$29 (\$49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								113
\$28 (\$48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								113
\$27 (\$47)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								113
\$26 (\$46)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								114
\$25 (\$45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	114
\$24 (\$44)	TCNT2	Timer/Counter2 (8 Bits)								127
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register								129
\$22 (\$42)	ASSR	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB	129
\$21 (\$41)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	130
\$20 ⁽²⁾ (\$40) ⁽²⁾	UBRRH	URSEL	-	-	-	-	UBRR[11:8]			41
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	165
\$1F (\$3F)	EEARH	-	-	-	-	-	-	-	EEAR8	164
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte								17
\$1D (\$3D)	EEDR	EEPROM Data Register								17
\$1C (\$3C)	EEDR	-	-	-	-	-	-	-	-	17
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	17
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	64
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	64
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	64
\$17 (\$37)	DDRB	ddb7	ddb6	ddb5	ddb4	ddb3	ddb2	ddb1	ddb0	64
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	64
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	64
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
\$0F (\$2F)	SPDR	SPI Data Register								65
\$0E (\$2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	140
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	-	-	SPI2X	140
\$0C (\$2C)	UDR	USART I/O Data Register								138
\$0B (\$2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	161
\$0A (\$2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	162
\$09 (\$29)	UBRRL	USART Baud Rate Register Low Byte								163
\$08 (\$28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	165
\$07 (\$27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	200
\$06 (\$26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	215
\$05 (\$25)	ADCH	ADC Data Register High Byte								217
\$04 (\$24)	ADCL	ADC Data Register Low Byte								218
\$03 (\$23)	TWDR	Two-wire Serial Interface Data Register								218
\$02 (\$22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	180
										180

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$01 (\$21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	179
\$00 (\$20)	TWBR	Two-wire Serial Interface Bit Rate Register								178

- Notes:
1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debugger specific documentation for details on how to use the OCSR Register.
 2. Refer to the USART description for details on how to access UBRRH and UCSRC.
 3. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 4. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers \$00 to \$1F only.



Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rd, K	Add Immediate to Word	$RdH:RdL \leftarrow RdH:RdL + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rd, K	Subtract Immediate from Word	$RdH:RdL \leftarrow RdH:RdL - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow \text{NOT } Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \text{NOT } Rd + 1$	Z,C,N,V	1
SBR	Rd, K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd, K	Clear Bit(s) in Register	$Rd \leftarrow Rd \cdot (\text{NOT } K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow \text{NOT } Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow \text{NOT } Rd$	Z,N,V	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow \text{STACK}$	None	4
RETI		Interrupt Return	$PC \leftarrow \text{STACK}$	None	4
CPSE	Rd, Rr	Compare, Skip if Equal	if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd, Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd, Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd, K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s)=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s)=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if $(Z=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if $(Z=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if $(C=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if $(C=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if $(C=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if $(C=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if $(N=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if $(N=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if $(H=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if $(T=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if $(T=0)$ $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if $(V=1)$ $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if $(V=0)$ $PC \leftarrow PC + k + 1$	None	1/2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled			
BRID	k	Branch if Interrupt Disabled			
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers			
MOVW	Rd, Rr	Copy Register Word	$Rd \leftarrow Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd+1, Rd \leftarrow Rr+1, Rr$	None	1
LD	Rd, X	Load Indirect	$Rd \leftarrow X$	None	1
LD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow (X)$	None	1
LD	Rd, -X	Load Indirect and Pre-Dec.	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd, Y	Load Indirect	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	$Rd \leftarrow (Y)$	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	None	2
LD	Rd, Z	Load Indirect	$Rd \leftarrow (Y + q)$	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow (Z)$	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (Z + q)$	None	2
ST	X, Rr	Store Indirect	$Rd \leftarrow (k)$	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$(X) \leftarrow Rr$	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	Y, Rr	Store Indirect	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$(Y) \leftarrow Rr$	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
STD	Y+q, Rr	Store Indirect with Displacement	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$(Y + q) \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	$(Z) \leftarrow Rr$	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2
STD	Z+q, Rr	Store Indirect with Displacement	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
STS	k, Rr	Store Direct to SRAM	$(Z + q) \leftarrow Rr$	None	2
LPM		Load Program Memory	$(k) \leftarrow Rr$	None	2
LPM	Rd, Z	Load Program Memory	$RD \leftarrow (Z)$	None	2
LPM	Rd, Z+	Load Program Memory and Post-Inc.	$Rd \leftarrow (Z)$	None	3
SPM		Store Program Memory	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	3
IN	Rd, P	In Port	$(Z) \leftarrow R1.R0$	None	3
OUT	P, Rr	Out Port	$Rd \leftarrow P$	None	-
PUSH	Rr	Push Register on Stack	$P \leftarrow Rr$	None	1
POP	Rd	Pop Register from Stack	$STACK \leftarrow Rr$	None	1
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P, b	Set Bit in I/O Register	$I/O(P, b) \leftarrow 1$	None	2
CBI	P, b	Clear Bit in I/O Register	$I/O(P, b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	None	2
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z, C, N, V	1
ROL	Rd	Rotate Left Through Carry	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z, C, N, V	1
ASR	Rd	Arithmetic Shift Right	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z, C, N, V	1
SWAP	Rd	Swap Nibbles	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z, C, N, V	1
BSET	s	Flag Set	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 1$	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	$SREG(s) \leftarrow 0$	SREG(s)	1
BLD	Rd, b	Bit load from T to Register	$T \leftarrow Rr(b)$	T	1
SEC		Set Carry	$Rd(b) \leftarrow T$	None	1
CLC		Clear Carry	$C \leftarrow 1$	C	1
SEN		Set Negative Flag	$C \leftarrow 0$	C	1
CLN		Clear Negative Flag	$N \leftarrow 1$	N	1
SEZ		Set Zero Flag	$N \leftarrow 0$	N	1
CLZ		Clear Zero Flag	$Z \leftarrow 1$	Z	1
SEI		Global Interrupt Enable	$Z \leftarrow 0$	Z	1
CLI		Global Interrupt Disable	$I \leftarrow 1$	I	1
SEV		Set Signed Test Flag	$I \leftarrow 0$	I	1
CLV		Clear Signed Test Flag	$S \leftarrow 1$	S	1
SET		Set Twos Complement Overflow	$S \leftarrow 0$	S	1
CLT		Clear Twos Complement Overflow	$V \leftarrow 1$	V	1
SET		Set T in SREG	$V \leftarrow 0$	V	1
CLT		Clear T in SREG	$T \leftarrow 1$	T	1
SEH		Set Half Carry Flag in SREG	$T \leftarrow 0$	T	1
			$H \leftarrow 1$	H	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation			
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-Chip Debug Only	None	N/A



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7 - 5.5V	ATmega16L-8AC	44A	Commercial (0°C to 70°C)
		ATmega16L-8PC	40P6	
		ATmega16L-8MC	44M1	
		ATmega16L-8AI	44A	Industrial (-40°C to 85°C)
		ATmega16L-8AU ⁽¹⁾	44A	
		ATmega16L-8PI	40P6	
16	4.5 - 5.5V	ATmega16L-8PU ⁽¹⁾	40P6	Industrial (-40°C to 85°C)
		ATmega16L-8MI	44M1	
		ATmega16L-8MU ⁽¹⁾	44M1	
		ATmega16-16AC	44A	Commercial (0°C to 70°C)
		ATmega16-16PC	40P6	
		ATmega16-16MC	44M1	
ATmega16-16AI	44A	Industrial (-40°C to 85°C)		
ATmega16-16AU ⁽¹⁾	44A			
ATmega16-16PI	40P6			
		ATmega16-16PU ⁽¹⁾	40P6	
		ATmega16-16MI	44M1	
		ATmega16-16MU ⁽¹⁾	44M1	

Note: 1. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.



Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44M1	44-pad, 7 x 7 x 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

Packaging Information

44A

COMMON DIMENSIONS
(Unit of Measure = mm)

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

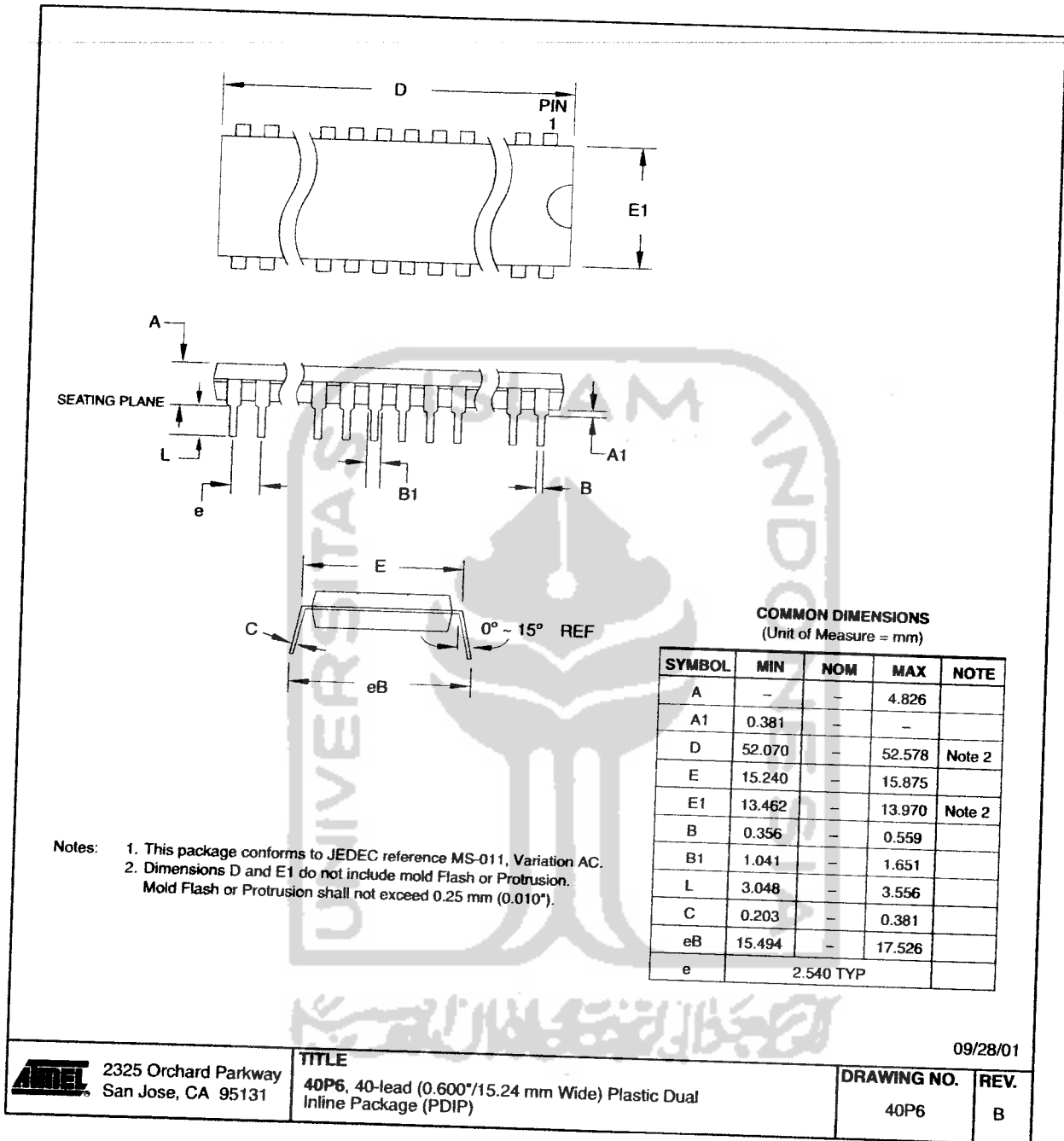
Notes:

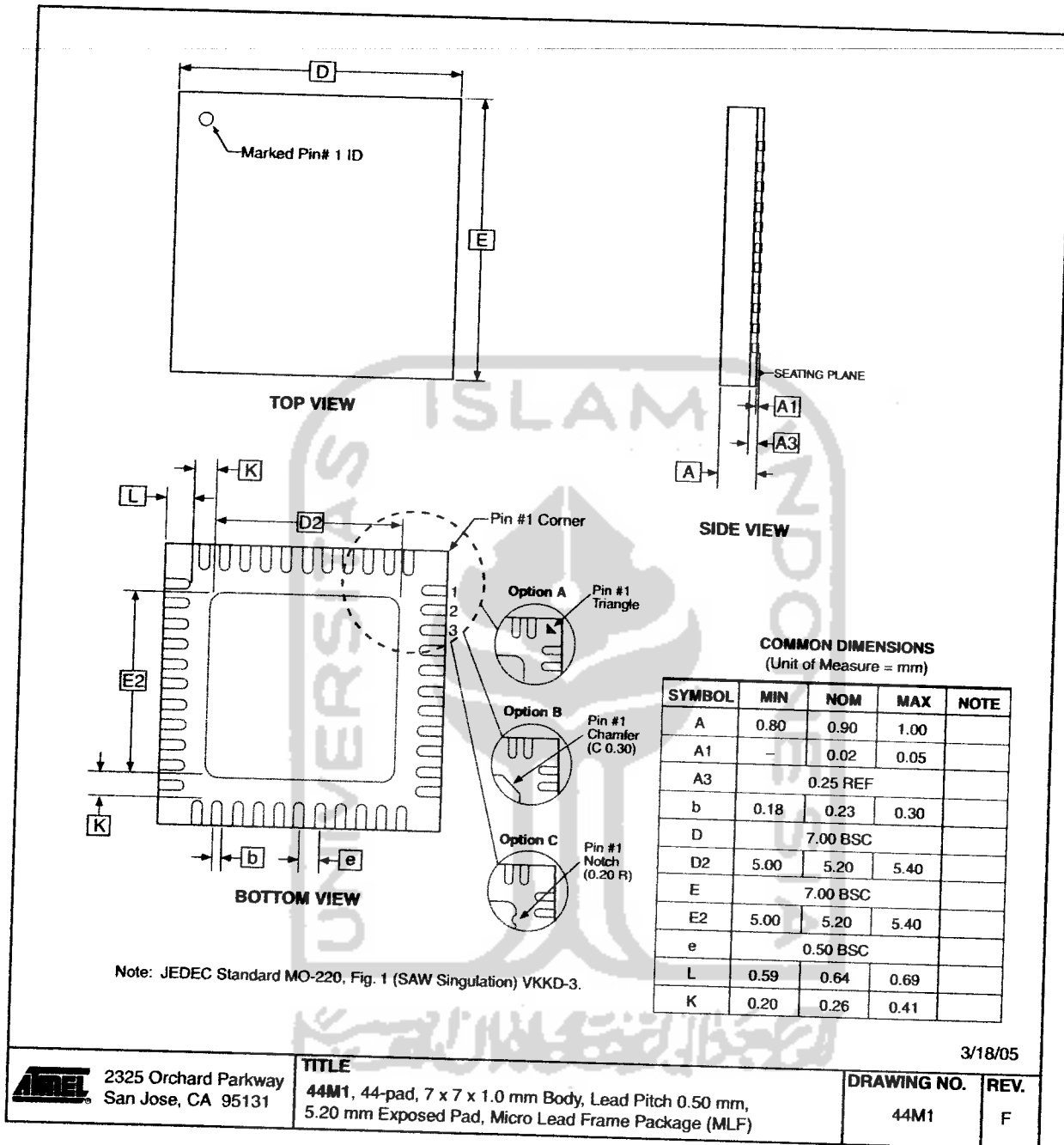
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

2325 Orchard Parkway San Jose, CA 95131	TITLE 44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Quad Flat Package (TQFP)	DRAWING NO. 44A	REV. B

40P6





Errata

The revision letter in this section refers to the revision of the ATmega16 device.

ATmega16(L) Rev. I

- **IDCODE masks data from TDI input**

1. **IDCODE masks data from TDI input**

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

ATmega16(L) Rev. H

- **IDCODE masks data from TDI input**

1. **IDCODE masks data from TDI input**

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

ATmega16(L) Rev. G

- **IDCODE masks data from TDI input**

1. **IDCODE masks data from TDI input**

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

Rev. 2466L-06/05

1. Updated note in “Bit Rate Generator Unit” on page 176.
2. Updated values for V_{INT} in “ADC Characteristics” on page 297.
3. Updated “Serial Programming Instruction set” on page 276.
4. Updated USART init C-code example in “USART” on page 142.

Rev. 2466K-04/05

1. Updated “Ordering Information” on page 11.
2. MLF-package alternative changed to “Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF”.
3. Updated “Electrical Characteristics” on page 291.

Rev. 2466J-10/04

1. Updated “Ordering Information” on page 11.

Rev. 2466I-10/04

1. Removed references to analog ground.
2. Updated Table 7 on page 26, Table 15 on page 36, Table 16 on page 40, Table 81 on page 208, Table 116 on page 276, and Table 119 on page 293.
3. Updated “Pinout ATmega16” on page 2.
4. Updated features in “Analog to Digital Converter” on page 202.
5. Updated “Version” on page 227.
6. Updated “Calibration Byte” on page 261.
7. Added “Page Size” on page 262.

Rev. 2466H-12/03

1. Updated “Calibrated Internal RC Oscillator” on page 27.

Rev. 2466G-10/03

1. Removed “Preliminary” from the datasheet.
2. Changed ICP to ICP1 in the datasheet.
3. Updated “JTAG Interface and On-chip Debug System” on page 34.
4. Updated assembly and C code examples in “Watchdog Timer Control Register – WDTCR” on page 41.
5. Updated Figure 46 on page 101.
6. Updated Table 15 on page 36, Table 82 on page 215 and Table 115 on page 276.

7. Updated “Test Access Port – TAP” on page 220 regarding JTAGEN.
 8. Updated description for the JTD bit on page 229.
 9. Added note 2 to Figure 126 on page 252.
 10. Added a note regarding JTAGEN fuse to Table 105 on page 260.
 11. Updated Absolute Maximum Ratings* and DC Characteristics in “Electrical Characteristics” on page 291.
 12. Updated “ATmega16 Typical Characteristics” on page 299.
 13. Fixed typo for 16 MHz QFN/MLF package in “Ordering Information” on page 11.
 14. Added a proposal for solving problems regarding the JTAG instruction IDCODE in “Errata” on page 15.
1. Added note about masking out unused bits when reading the Program Counter in “Stack Pointer” on page 10.
 2. Added Chip Erase as a first step in “Programming the Flash” on page 288 and “Programming the EEPROM” on page 289.
 3. Added the section “Unconnected pins” on page 53.
 4. Added tips on how to disable the OCD system in “On-chip Debug System” on page 34.
 5. Removed reference to the “Multi-purpose Oscillator” application note and “32 kHz Crystal Oscillator” application note, which do not exist.
 6. Added information about PWM symmetry for Timer0 and Timer2.
 7. Added note in “Filling the Temporary Buffer (Page Loading)” on page 253 about writing to the EEPROM during an SPM Page Load.
 8. Removed ADHSM completely.
 9. Added Table 73, “TWI Bit Rate Prescaler,” on page 180 to describe the TWPS bits in the “TWI Status Register – TWSR” on page 179.
 10. Added section “Default Clock Source” on page 23.
 11. Added note about frequency variation when using an external clock. Note added in “External Clock” on page 29. An extra row and a note added in Table 118 on page 293.
 12. Various minor TWI corrections.
 13. Added “Power Consumption” data in “Features” on page 1.

Rev. 2466F-02/03

14. Added section “EEPROM Write During Power-down Sleep Mode” on page 20.
15. Added note about Differential Mode with Auto Triggering in “Prescaling and Conversion Timing” on page 205.
16. Added updated “Packaging Information” on page 12.

Rev. 2466E-10/02

1. Updated “DC Characteristics” on page 291.

Rev. 2466D-09/02

1. Changed all Flash write/erase cycles from 1,000 to 10,000.
2. Updated the following tables: Table 4 on page 24, Table 15 on page 36, Table 42 on page 83, Table 45 on page 110, Table 46 on page 110, Table 59 on page 141, Table 67 on page 165, Table 90 on page 234, Table 102 on page 258, “DC Characteristics” on page 291, Table 119 on page 293, Table 121 on page 295, and Table 122 on page 297.
3. Updated “Errata” on page 15.

Rev. 2466C-03/02

1. Updated typical EEPROM programming time, Table 1 on page 18.
2. Updated typical start-up time in the following tables:
Table 3 on page 23, Table 5 on page 25, Table 6 on page 26, Table 8 on page 27, Table 9 on page 27, and Table 10 on page 28.
3. Updated Table 17 on page 41 with typical WDT Time-out.
4. Added Some Preliminary Test Limits and Characterization Data.
Removed some of the TBD's in the following tables and pages:
Table 15 on page 36, Table 16 on page 40, Table 116 on page 272 (table removed in document review #D), “Electrical Characteristics” on page 291, Table 119 on page 293, Table 121 on page 295, and Table 122 on page 297.
5. Updated TWI Chapter.
Added the note at the end of the “Bit Rate Generator Unit” on page 176.
6. Corrected description of ADSC bit in “ADC Control and Status Register A – ADCSRA” on page 217.
7. Improved description on how to do a polarity check of the ADC doff results in “ADC Conversion Result” on page 214.
8. Added JTAG version number for rev. H in Table 87 on page 227.
9. Added note regarding OCDEN Fuse below Table 105 on page 260.
10. Updated Programming Figures:
Figure 127 on page 262 and Figure 136 on page 274 are updated to also reflect that AVCC must be connected during Programming mode. Figure 131 on page 270 added to illustrate how to program the fuses.

11. Added a note regarding usage of the “PROG_PAGELOAD (\$6)” on page 280 and “PROG_PAGEREAD (\$7)” on page 280.
12. Removed alternative algorithm for leaving JTAG Programming mode.
See “Leaving Programming Mode” on page 288.
13. Added Calibrated RC Oscillator characterization curves in section “ATmega16 Typical Characteristics” on page 299.
14. Corrected ordering code for QFN/MLF package (16MHz) in “Ordering Information” on page 11.
15. Corrected Table 90, “Scan Signals for the Oscillators(1)(2)(3),” on page 234.





Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chanterrie

BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© Atmel Corporation 2005. All rights reserved. Atmel®, logo and combinations thereof, Everywhere You Are®, AVR®, AVR Studio®, and others, are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.



Printed on recycled paper.

2466LS-AVR-06/05