



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Unand.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Unand.

PERANCANGAN SENSOR PENGUKUR LEVEL CAIRAN UNTUK MENGUKUR VOLUME CAIRAN DI TABUNG FERMENTASI PADA PROSES PEMBUATAN BIOETANOL

SKRIPSI



**DUANO UTAMA
06175049**

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS ANDALAS
PADANG 2012**

Abstrak

Kebutuhan akan bahan bakar alternatif pengganti bahan bakar fosil sangat dibutuhkan saat sekarang ini. Bioetanol adalah salah satu bahan bakar alternatif yang sedang dikembangkan. Untuk memenuhi kebutuhan tersebut diperlukan alat-alat produksi bioetanol yang dapat bekerja secara otomatis.

Dalam proses produksi bioetanol salah satu bagian yang terpenting adalah proses fermentasi . Proses fermentasi akan terbentuk endapan bahan baku pada dasar tabung dan cairan di atasnya. Untuk mengukur volume cairan dibutuhkan sebuah sensor karena hanya cairan sajalah yang akan diolah pada proses selanjutnya. Sensor ini terdiri dari sensor encoder dan sensor ultrasonik. Sensor encoder digunakan untuk mengukur ketinggian endapan, sedangkan sensor ultrasonik digunakan untuk mengukur tinggi total cairan pada tabung. Semua aktifitas dari kedua sensor di kendalikan oleh sebuah Mikrokontroller secara otomatis.

Sensor pengukur volume ini dikalibrasi sebanyak tiga kali pada ketinggian 4 cm, 7 cm dan 10 cm dengan persentase kesalahan masing-masing sebesar 8.9 %, 7.5 %, dan 9.42 %. Pengukuran tinggi dan volume pada tabung fermentasi menghasilkan persentase kesalahan sebesar 14.52 % dan 14.39 %.

Kata kunci : Bioetanol, Tabung Fermentasi, Sensor Encoder, Sensor Ultrasonik, Mikrokontroller.

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Syukur Alhamdulillah penulis ucapkan kehadiran Allah SWT, karena limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan penulisan tugas akhir ini, serta shalawat dan salam selalu tercurah kepada baginda nabi besar, Muhammad SAW, yang selalu menjadi panutan bagi umat manusia hingga akhir zaman.

Tugas akhir yang berjudul **“PERANCANGAN SENSOR PENGUKUR LEVEL CAIRAN UNTUK MENGUKUR VOLUME CAIRAN DI TABUNG FERMENTASI PADA PROSES PEMBUATAN BIOETANOL”** ini merupakan salah satu syarat untuk menyelesaikan program Strata-1 di Jurusan Teknik Elektro Fakultas Teknik Universitas Andalas. Dukungan dan semangat yang diberikan dari berbagai pihak sangat membantu penulis menyelesaikan tugas akhir ini, oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah banyak membantu terutama sekali kepada :

1. Kedua orang tua dan segenap keluarga yang telah memberikan kasih sayang, dukungan moril dan materil serta doa yang senantiasa menyertai penulis selama menempuh pendidikan ini. Semoga Allah selalu memberikan kasih sayang yang berlipat ganda kepada mereka.
2. Bapak Rahmadi Kurnia, DR. ENG. selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Andalas.

3. Bapak Mumuh Muharam, MT dan Bapak M, Ilhamdi Rusydi, MT sebagai pembimbing atas segala bantuan berupa ilmu, bimbingan dan kesabaran yang telah diberikan selama penulis melakukan penelitian sehingga tugas akhir ini dapat diselesaikan.
4. Bapak Heru Dibyo Laksono, MT selaku Kepala Laboratorium Kontrol Teknik Elektro.
5. Bapak dan Ibu Dosen di lingkungan Fakultas Teknik umumnya dan Jurusan Teknik Elektro khususnya atas segala didikan dan ilmu yang telah diberikan.
6. Pegawai Jurusan atas bantuan dalam segala urusan administrasi.
7. Semua pihak yang telah membantu baik langsung maupun tidak langsung, yang tidak mungkin disebutkan satu persatu.

Semoga Allah SWT melimpahkan rahmat beserta karunia-Nya kepada semua pihak yang telah banyak membantu penulis dalam menyelesaikan penelitian tugas akhir dan penyusunan laporan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, karena kesempurnaan itu hanya milik Allah SWT. Oleh karena itu saran dan masukan dari pembaca sangat penulis hargai. Akhir kata semoga laporan tugas akhir ini bermanfaat dan dapat menambah wawasan bagi semua pembaca dan penulis sendiri.

Padang, 3 Februari 2012

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN	
HALAMAN PERSEMBAHAN	
ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	5
BAB II TEORI DASAR	7
2.1 Bioetanol	7
2.1.1 Gasohol	9
2.1.2 Bahan Baku Pembuatan Etanol dari Residu Produksi	10
2.2 Komponen Alat	11
2.2.1 Sistem Aktuator	11
2.2.1.1 Motor DC Servo	11

2.2.1.2 Motor DC Brushless (DC-BL)	14
2.2.1.3 Motor DC Stepper	15
2.2.2 Catu Daya	19
2.2.3 Sensor JarakPING)))™ <i>Ultrasonic Range Finder</i>	21
2.2.4 Sensor Encoder	23
2.2.5 Mikrokontroler	24
2.2.5.1 Mikrokontroler ATMEL	25
2.2.5.2 Mikrokontroler PIC	26
2.2.5.3 Mikrokontroler Renesas	27
BAB III METODOLOGI PENELITIAN	29
3.1 Gambaran Sistem	29
3.2 Metode Penelitian	30
BAB IV PERANCANGAN SISTEM	32
4.1 Garis Besar Perancangan Sistem	32
4.2 Perancangan Perangkat Keras	33
4.2.1 Perancangan Mekanik	34
4.2.2 Perancangan Elektrik	34
4.2.2.1 Modul Mikrokontroler Atmega 8535	34

4.2.2.1.1 Mikrokontroler AVR ATMEGA 8535	36
4.2.2.1.2 Catu Daya / Power Supply	38
4.2.2.1.3 Rangkaian Komunikasi Serial	39
4.2.2.1.4 LCD 2x 16	40
4.2.2.2 Sensor Ping Ultrasonic Range Finder	40
4.2.2.3 Sensor Encoder	42
4.2.2.4 Motor DC	43
4.2.2.5 Motor Steper	44
4.2.2.6 Sistem Tata Letak Perangkat Keras	44
4.3 Perancangan Perangkat Lunak (Software)	46
BAB V HASIL DAN ANALISA	48
5.1 Pengujian untuk Kalibrasi	48
5.1.1 Pengujian Ketinggian 4 Cm	48
5.1.2 Pengujian Ketinggian 7 Cm	51
5.1.3 Pengujian Ketinggian 10 Cm	53
5.2 Pengujian di Tabung Fermentasi	55
BAB VI PENUTUP	62
6.1 Kesimpulan	62
6.2 Saran	63
DAFTAR PUSTAKA	64
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1	Pulsa servo continuous	13
Gambar 2.2	Pulsa servo standar 180 ⁰	14
Gambar 2.3	Bentuk pulsa motor stepper	15
Gambar 2.4	Bearing motor stepper	16
Gambar 2.5	Prinsip rangkaian penyearah	19
Gambar 2.6	Prinsip penyearah gelombang penuh	20
Gambar 2.7	Susunan kaki regulator	21
Gambar 2.8	Bentuk dan Konfigurasi Pin Sensor PING))) TM	21
Gambar 2.9	Ilustrasi Cara Kerja Sensor PING))) TM	22
Gambar 2.10	Diagram Waktu Sensor PING))) TM	22
Gambar 2.11	Disk incremental encoder dua kanal dan sinyal keluarannya	24
Gambar 2.12	Perkembangan Mikrokontroler keluaran ATMEL	25
Gambar 3.1	Gambaran Sistem	29
Gambar 4.1	Diagram Blok Sistem	33
Gambar 4.2	Skema Rangkaian Sistem Minimum AVR ATmega 8535	35
Gambar 4.3	Modul Mikrokontroler ATmega 8535 pada satu board	35
Gambar 4.4	Pin Out Mikrokontroler ATMEGA 8535	37
Gambar 4.5	Skema Rangkaian Power Supply	38
Gambar 4.6	Skema Rangkaian Komunikasi Port Serial	40
Gambar 4.7	Sensor Ping))) Ultrasonic Range Finder	41
Gambar 4.8	Skema Pengukuran	41

Gambar 4.9 Skema Rangkaian Optocoupler	43
Gambar 4.10 Skema Rangkaian Driver Motor DC	43
Gambar 4.11 Skema Rangkaian Driver Motor Stepper	44
Gambar 4.12 Sensor Volume Cairan tampak depan	44
Gambar 4.13 Sensor Pengukur Volume Cairan tampak belakang	45
Gambar 4.14 Sensor Pengukur Cairan tampak samping	45
Gambar 4.15 Diagram Alir sensor Pengukur Volume Cairan	47
Gambar 5.1 Grafik Hasil Pengujian Ketinggian 4 cm	50
Gambar 5.2 Grafik Hasil Pengujian Ketinggian 7 cm	53
Gambar 5.3 Grafik Hasil Pengujian Ketinggian 10 cm	55
Gambar 5.4 Grafik Hasil Pengukuran Ketinggian	59
Gambar 5.5 Grafik Hasil Pengukuran Volume	60



DAFTAR TABEL

Tabel 2.1 Keadaan incremental encoder dua kanal	23
Tabel 5.1 Pengujian Ketinggian 4 cm	49
Tabel 5.2 Pengujian Ketinggian 7 cm	51
Tabel 5.3 Pengujian Ketinggian 10 cm	54
Tabel 5.4 Pengukuran Tinggi dan Volume	57



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Seiring dengan perkembangan zaman yang semakin pesat mendorong manusia untuk melakukan eksploitasi sumber daya alam secara berlebihan. Salah satu sumber daya alam yang menjadi objek eksploitasi yang sangat besar adalah minyak bumi. Saat sekarang ini kehidupan manusia sudah sangat bergantung pada bahan bakar fosil, baik itu dalam kehidupan rumah tangga maupun proses produksi industri. Minyak bumi atau bahan bakar fosil merupakan jenis sumber daya alam yang tidak dapat diperbaharui dan membutuhkan waktu berjuta-juta tahun dalam proses pembentukannya di alam, sementara kebutuhan manusia akan bahan bakar meningkat setiap waktu sehingga mengakibatkan jumlah ketersediannya semakin berkurang. Untuk mengatasi ketersediaan bahan bakar fosil yang semakin menipis dibutuhkan suatu jenis bahan bakar lain yang dapat menggantikan kebutuhan akan bahan bakar fosil^[1].

Salah satu energi alternatif yang menjanjikan adalah bioetanol. Bioethanol adalah ethanol yang bahan utamanya dari tumbuhan dan umumnya menggunakan proses fermentasi. Ethanol atau ethyl alkohol C_2H_5OH berupa cairan bening tak berwarna, terurai secara biologis (*biodegradable*), toksisitas rendah dan tidak menimbulkan polusi udara yg besar bila bocor. Ethanol adalah bahan bakar beroktan tinggi dan dapat menggantikan timbal sebagai peningkat nilai oktan dalam bensin. Dengan mencampur ethanol dengan bensin, akan mengoksigenasi campuran bahan bakar sehingga dapat terbakar lebih sempurna dan mengurangi emisi gas buang (seperti karbonmonoksida/CO).

Bioethanol dapat dibuat dari proses fermentasi singkong. Singkong (Manihot utilissima) sering juga disebut sebagai ubi kayu atau ketela pohon, merupakan tanaman yang sangat populer di seluruh dunia, khususnya di negara-negara tropis. Potensi singkong di Indonesia cukup besar maka dipilihlah singkong sebagai bahan baku utama^[2].

Bioetanol memang potensial dimanfaatkan sebagai bahan bakar kendaraan bermotor. Syaratnya, etanol alami itu mesti berkadar kemurnian 99,5%. Syarat itu mutlak karena jika berkadar di bawah 90%, mesin tidak bisa menyala karena kandungan airnya terlampaui tinggi. Sebetulnya bioetanol berkadar kemurnian 95% masih layak dimanfaatkan sebagai bahan bakar motor. Hanya saja, dengan kadar kemurnian itu perlu penambahan zat antikorosif pada tangki bahan bakar agar tidak menimbulkan karat. Sayangnya, saat ini banyak produsen yang menghasilkan bioetanol dengan kemurnian yang masih rendah^[3].

Untuk memproses bahan baku berupa singkong menjadi bioetanol dibutuhkan suatu alat yang dapat bekerja dengan efektif dan efisien. Penulis melakukan perancangan dan pembuatan model alat pemroses bioetanol secara otomatis yang mana setiap komponen atau bagian-bagian dari alat tersebut dikendalikan melalui hardware dan software yang terintegrasi dalam suatu rangkaian elektronik.

Adapun beberapa penelitian yang menjadi acuan penulis adalah :

1. Penelitian dilakukan oleh Supriyanto (2007) dengan judul **“Prospek Pengembangan Industri Bioetanol dari Ubi Kayu”**. dan Penelitian yang dilakukan oleh Heppy Rikana dan Reski Adam (2010) dengan judul **“Pembuatan Bioetanol dari Singkong Secara Fermentasi Menggunakan ragi Tape”**. Dari penelitian ini dapat dilihat prospek ubi kayu sebagai bahan dasar pembuatan bioetanol dari segala aspek yang mendukung.

2. Penelitian dilakukan oleh Teguh Wiguna (2007) dengan judul “**Pengukur Volume Zat Cair Menggunakan Gelombang Ultrasonik Berbasis Mikrokontroller AT89S51**”. Dari penelitian ini dapat dilihat prinsip penggunaan sensor ultrasonik sebagai alat pengukur volume cairan.

1.2.Rumusan Masalah

Sesuai dengan latar belakang masalah yang telah dikemukakan di depan, maka dapat dirumuskan permasalahan yang harus dihadapi, yaitu :

1. Perancangan dan pembuatan model alat pembuat bioetanol otomatis berbasis mikrokontroler.
2. Perancangan sensor pendeteksi jumlah volume cairan yang berada pada tabung fermentasi pada proses pembuatan bioetanol berbasis mikrokontroller.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk menghasilkan sebuah sensor pendeteksi jumlah volume cairan yang berada pada tabung fermentasi pada proses pembuatan bioetanol berbasis mikrokontroller. Sensor ini diharapkan mampu untuk menentukan volume cairan yang akan diolah pada proses selanjutnya.

1.4 Batasan Masalah

Hal yang menjadi batasan masalah pada penelitian ini adalah :

1. Pengukuran volume cairan dilakukan pada tabung fermentasi yang terdapat cairan dan endapan.

2. Pembuatan pengukur volume cairan ini menggunakan sensor ultrasonik, sensor encoder dan mikrokontroler Atmega8535 yang hasil pengukurannya akan ditampilkan pada LCD.
3. Wadah zat cair telah ditentukan sebelumnya agar dapat dilakukan perhitungan volumenya
4. Wadah cairan yang akan diukur berbentuk tabung.
5. Mikrokontroler Atmega8535 digunakan untuk mengatur sensor PING *Ultrasonik range finder* dalam pemancaran sinyal, mengatur pergerakan motor, membaca perubahan bit dari encoder, menghitung jarak yang terukur dari sensor PING dengan encoder.
6. Bahasa pemrograman pada mikrokontroler Atmega8535 menggunakan bahasa C.

1.5 Manfaat Penelitian

Dari penelitian yang telah dilakukan dapat diambil manfaat, yaitu

1. Menghasilkan suatu alat yang bermanfaat dan dapat memudahkan dalam pemrosesan bioethanol.
2. Memanfaatkan penggunaan mikrokontroler dalam mengendalikan alat pemrosesan bioethanol.
3. Alat tersebut dapat menghasilkan produk bioethanol yang lebih baik dari pada produk yang dihasilkan secara konvensional.
4. Alat tersebut dapat menjadi batu pijakan dalam mengembangkan alat yang lebih baik yang lebih berdaya guna dan berdaya hasil di masa depan.

1.6 Sistematika Penulisan

Sistematika penulisan bertujuan untuk lebih mempermudah dan memperjelas dalam penyampaian informasi pembahasan masalah, dengan susunan sebagai berikut:

Bab I : Pendahuluan

Berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan Tugas Akhir.

Bab II : Tinjauan Pustaka

Membahas tentang teori-teori pendukung yang digunakan dalam perencanaan dan pembuatan Tugas Akhir. Berisi teori-teori tentang bentuk sistem mekanik alat, jenis-jenis mikrokontroler, sensor dan penelitian-penelitian pendukung lainnya.

Bab III : Metode Penelitian

Berisikan tentang penjelasan tahap-tahap penelitian dan data yang akan diambil pada penelitian.

Bab IV : Perancangan dan Implementasi Sensor

Berisikan tentang perancangan sensor yang akan dibuat untuk Tugas Akhir, meliputi garis besar perancangan sistem, perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*) yang digunakan.

Bab V : Hasil dan Analisa

Membahas hasil dari pengujian dan analisa alat. Pengujian dilakukan pada ketepatan sensor ultrasonik dan sensor encoder dalam menentukan volume cairan.

Bab VI : Penutup

Penutup berisikan kesimpulan hasil perancangan, pengujian dan pembahasan sistem dari penulisan Tugas Akhir, serta saran untuk dapat dijadikan bahan pertimbangan dalam pengembangan perancangan sistem berikutnya.



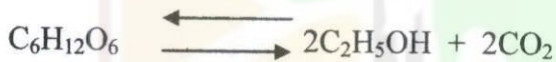
BAB II

TEORI DASAR

2.1. Bioetanol ^[4]

Louis Pasteur untuk pertama kalinya mengenalkan metode fermentasi. Dia melakukan fermentasi gula menggunakan mikroorganisme. Dia telah membuka cakrawala baru memproduksi senyawa kimia dengan bantuan mikroorganisme. Sehingga kita tidak harus capai-capai melakukan sintesis senyawa kimia, biarkan saja mikroorganisme yang bekerja memproduksinya.

Pada tahun 1815, Gay-Lussac memformulasikan konversi glukosa menjadi etanol dan karbondioksida. Formulasinya sebagai berikut :



Dalam perkembangannya produksi alkohol yang paling banyak digunakan adalah metode fermentasi dan distilasi.

Mikroorganisme yang digunakan untuk fermentasi alkohol :

Bakteri : *Clostridium acetobutylicum*, *Klebsiella pneumoniae*, *Leuconoctoc mesenteroides*, *Sarcina ventriculi*, *Zymomonas mobilis*, dll.

Fungi : *Aspergillus oryzae*, *Endomyces lactis*, *Kloeckera sp.*, *Kluyveromyces fragilis*, *Mucor sp.*, *Neurospora crassa*, *Rhizopus sp.*, *Saccharomyces beticus*, *S. cerevisiae*, *S. ellipsoideus*, *S. oviformis*, *S. saki*, *Torula sp.*, dll

Baru-baru ini teknologi DNA rekombinan telah membantu penggunaan mikroorganisme dalam proses industri. Setelah USA dan Brazil, India adalah negara terbesar ketiga dalam memproduksi bioetanol.

Selama perang dunia II campuran etanol dan bensin telah digunakan di Eropa. Namun, setelah perang berakhir bioetanol kalah bersaing dengan bensin yang harganya lebih murah. Penggunaan campuran alkohol dan bensin digunakan lagi pada tahun 1970-an akibat embargo minyak negara-negara Arab terhadap negara-negara barat pada tahun 1973 yang menyebabkan krisis minyak.

Pada tahun 1985 Brazil mengeluarkan program pencampuran 20% bioetanol dengan bensin untuk menghemat 40% konsumsi bensin. Negara ini telah memasarkan 1 juta mobil dengan bahan bakar 100% bioetanol.

Kelebihan-kelebihan bioetanol dibandingkan bensin:

1. Bioetanol aman digunakan sebagai bahan bakar, titik nyala etanol tiga kali lebih tinggi dibandingkan bensin.
2. Emisi hidokarbon lebih sedikit

Kekurangan-kekurangan bioetanol dibandingkan bensin:

1. Mesin dingin lebih sulit melakukan starter
2. Bioetanol bereaksi dengan logam seperti magnesium dan aluminium.

Sebagai alternatif digunakan campuran bioetanol dengan bensin. Sebelum dicampur, bioetanol harus dimurnikan hingga 100%. Campuran ini dikenal dengan sebutan gasohol.

Substrat yang dapat difermentasikan menjadi alkohol :

1. Bahan bergula (*sugary materials*) : tebu dan sisa produknya (molase, bagase), gula bit, tapioca, kentang manis, sorghum manis, dll. Molasses tebu digunakan besar-besaran di beberapa negara untuk memproduksi alkohol.
2. Bahan-bahan berpati (*starchy materials*) : tapioka, maizena, barley, gandum, padi, dan kentang. Jagung dan ubikayu adalah dua kelompok substrat yang menarik perhatian. 11,7 kg tepung jagung dapat dikonversi menjadi 7 liter etanol.
3. Bahan-bahan lignoselulosa (*lignosellulosic material*) : sumber selulosa dan lignoselulosa berasal dari limbah pertanian dan kayu. Akan tetapi, hasil etanol dari lignoselulosa sedikit karena kekurangan teknologi untuk mengkonversi pentosa menjadi etanol. 409 liter etanol dapat diproduksi dari 1 ton lignoselulosa.

2.1.1. Gasohol ^[4]

Bioetanol bersifat multi-guna karena dicampur dengan bensin pada komposisi berapapun memberikan dampak yang positif. Pencampuran bioetanol absolut sebanyak 10 % dengan bensin (90%), sering disebut Gasohol E-10. Gasohol singkatan dari gasoline (bensin) plus alkohol (bioetanol). Etanol absolut memiliki angka oktan (ON) 117, sedangkan Premium hanya 87-88. Gasohol E-10 secara proporsional memiliki ON 92 atau setara Pertamina. Pada komposisi ini bioetanol dikenal sebagai *octan enhancer* (aditif) yang paling ramah lingkungan

dan di negara-negara maju telah menggeser penggunaan *Tetra Ethyl Lead* (TEL) maupun *Methyl Tertiary Butyl Ether* (MTBE).

Pencampuran sampai dengan 24 % masih dapat menggunakan mobil bensin konvensional. Di atas itu, diperlukan mobil khusus yang telah banyak diproduksi di AS maupun Brazil. Yang populer dan diminati saat ini adalah Flexible-Fuel Vehicle (FFV). Ini sejenis “mobil cerdas” karena dilengkapi dengan sensor dan panel otomatisasi yang dapat mengatur mesin untuk menggunakan campuran bensin-bioetanol pada komposisi berapapun.

2.1.2. Bahan Baku Pembuatan Etanol dari Residu Produksi ^[5]

Bahan baku pembuatan etanol dari residu produksi terdiri dari berbagai bahan antara lain ;

a. Produksi etanol dari tetes (molasses)

Tetes merupakan hasil sampingan proses pembuatan gula. Tetes mengandung sejumlah besar gula baik sukrosa maupun gula pereduksi. Total kandungan gula berkisar 48-56 % sedangkan pH-nya 5,5 -6.5.

b. Pembuatan etanol dari onggok

Onggok merupakan hasil sampingan pengolahan ubi kayu. Komponen utama pada onggok adalah pati dan serat kasar. Kandungan karbohidrat onggok sekitar 65 % persen dan serat kasar 8%.

c. Pembuatan etanol dari selulosa (bagas)

Bagas merupakan residu padat pada proses pengolahan tebu menjadi gula. yang sejauh ini masih belum banyak dimanfaatkan menjadi produk yang mempunyai nilai tambah. Bagas yang termasuk biomassa mengandung lignocellulose sangat dimungkinkan untuk dimanfaatkan menjadi sumber energi

alternatif bioetanol atau biogas. Pemanfaatan bagas menjadi etanol merupakan suatu skenario yang mengaju pada kebijakan pemerintah yang telah menetapkan salah fokus penelitian dan penerapan Iptek sampai tahun 2025 adalah penciptaan dan pemanfaatan energi baru dan terbarukan.

2.2 Komponen Alat

2.2.1 Sistem Aktuator

Aktuator adalah perangkat elektromekanik yang menghasilkan daya gerakan. Dapat dibuat dari sistem motor listrik (Motor DC (*permanent magnet*, *brushless*, *shunt*, dan *series*), Motor DC Servo, Motor DC Stepper, ultrasonic motor, linear motor, torque motor, solenoid, dan sebagainya), sistem pneumatik (perangkat kompresi bernasis udara atau gas nitrogen), dan perangkat hidrolik (berbasis bahan cair seperti oli). Untuk meningkatkan tenaga mekanik actuator atau torsi gerakan dapat dipasang sistem gearbox, baik sistem direct-gear (sistem lurus, sistem ohmic/worm-gear, planetary gear, dan sebagainya), sprochet-chain (gir-rantai, *gir-belt*, ataupun sistem *wire-roller*, dan sebagainya)^[6].

2.2.1.1 Motor DC Servo

Motor DC Servo (DC-SV) pada dasarnya adalah motor DC-MP dengan kualifikasi khusus yang sesuai dengan aplikasi "*servoing*" di dalam teknik kontrol. Dalam kamus Oxford istilah "*servo*" diartikan sebagai "*a mechanism that controls a larger mechanism*".

Beberapa tipe motor DC servo yang dijual bersama dengan paket rangkaian *drivernya* telah memiliki rangkaian kontrol kecepatan yang menyatu di dalamnya. Putaran motor tidak lagi berdasarkan tegangan *supply* ke motor,

namun berdasarkan tegangan *supply* ke motor, namun berdasarkan tegangan input khusus yang berfungsi sebagai referensi kecepatan *output*^[7].

Adapun jenis-jenis motor servo adalah:

a. Motor Servo Standar Standar 180^0

Motor jenis ini hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing- masing sudut mencapai 90^0 sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180^0 .

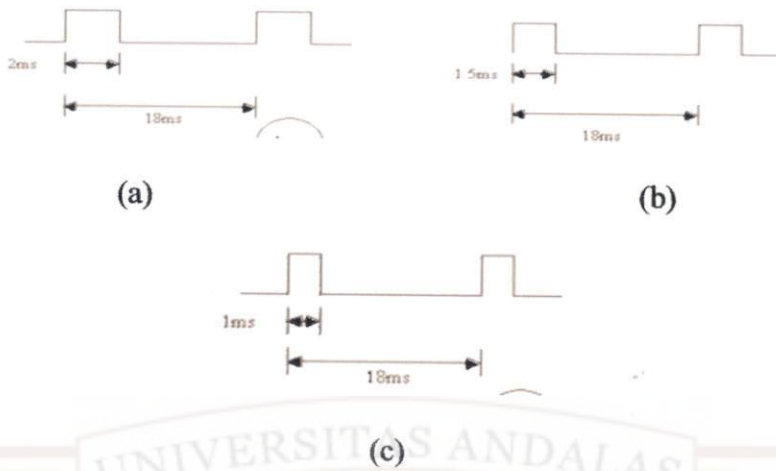
b. Motor Servo Continuous

Motor servo jenis ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat diputar secara kontinyu).

Motor servo standar 180^0 akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Adapun kondisi yang terdapat pada motor servo standar 180^0 adalah:

- a. Pada saat *Ton duty cycle* 1,5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut 0^0 / netral).
- b. Pada saat *Ton duty cycle* kurang dari 1,5 ms, maka rotor dari motor akan berputar kearah kanan dengan membentuk sudut yang besarnya linear terhadap besarnya *Ton duty cycle*, dan akan bertahan pada posisi tersebut.
- c. Pada saat *Ton duty cycle* lebih dari 1,5 ms, maka rotor dari motor akan berputar kearah kiri dengan membentuk sudut yang besarnya linear terhadap besarnya *Ton duty cycle*, dan akan bertahan pada posisi tersebut.

Gambar 2.3 adalah bentuk pulsa motor servo continuous pada saat putar kiri, berhenti, dan putar kanan.

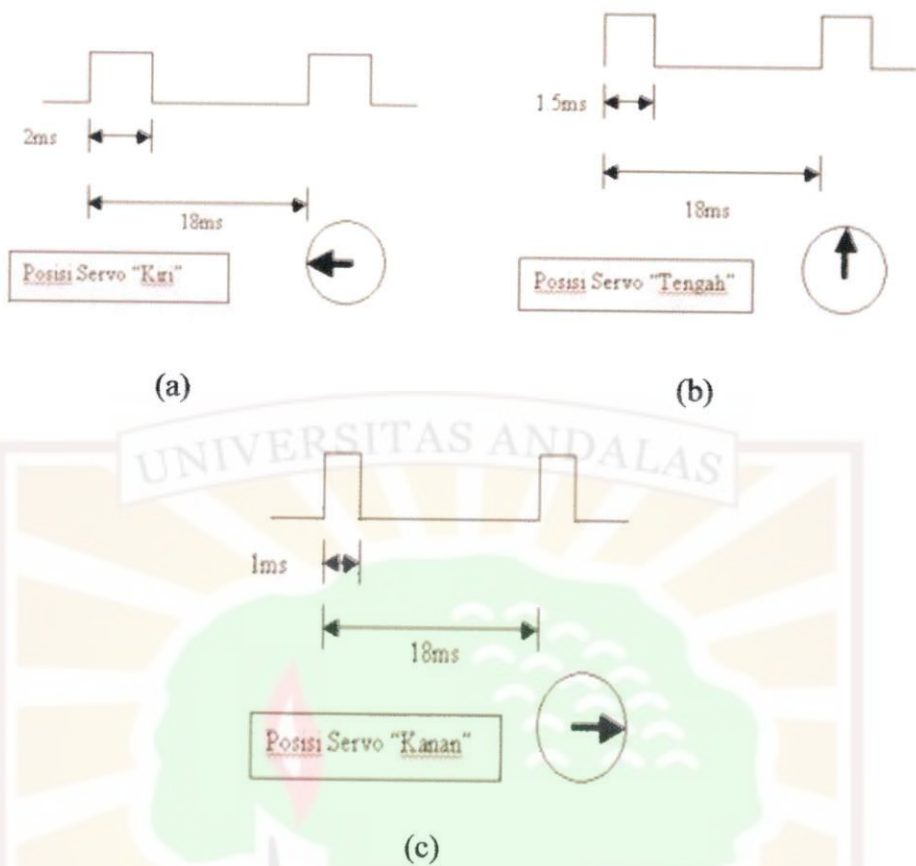


Gambar 2.1 *Pulsa servo continiuos (a) servo putar kiri (b) servo berhenti dan (c) servo putar kanan.*

Motor servo continiuos memiliki memiliki frekuensi PWM yang sama dengan servo standar 180^0 yaitu 50 Hz, adapun kondisi yang terdapat pada motor servo adalah sebagai berikut:

- Pada saat *Ton duty cycle* 1,5 ms, maka rotor dari motor akan diam.
- Pada saat *Ton duty cycle* kurang dari 1,5 ms, maka rotor dari motor akan berputar kearah kanan (CW), semakin nilai *Ton duty cylce* menjauhi 1,5 ms maka putaran rotor akan semakin cepat.
- Pada saat *Ton duty cycle* lebih dari 1,5 ms, maka rotor dari motor akan berputar kearah kiri (CCW), semakin nilai *Ton duty cylce* menjauhi 1,5 ms maka putaran rotor akan semakin cepat. Gambar 2.5 dibawah adalah bentuk pulsa motor DC servo standar 180^0 pada saat posisi kiri, posisi tengah, dan posisi kanan.

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS



Gambar 2.2 Pulsa servo standar 180⁰ (a) servo posisi kiri, (b) servo posisi tengah, dan (c) servo posisi kanan

2.2.1.2 Motor DC Brushless (DC-BL) ^[8]

Perbedaan utama antara motor DC-MP dengan motor DC brushless adalah terletak pada pembangkitan medan magnet untuk menghasilkan gaya gerak. Jika pada motor DC-MP medan magnet yang dikontrol berada di rotor dan medan magnet tetap berada di stator. Maka sebaliknya, motor DC-BL menggunakan pembangkitan medan magnet stator untuk mengontrol gerakanya sedang medan magnet tetap berada di rotor.

Prinsip kerja motor DC-BL mirip seperti motor AC asinkron. Putaran diperoleh dari perbedaan kutub medan magnet yang dihasilkan oleh fasa tegangan yang berbeda.

2.2.1.3 Motor DC Stepper^[9]

Motor stepper adalah motor listrik yang dikendalikan dengan pulsa-pulsa digital, bukan dengan memberikan tegangan yang terus-menerus. Deretan pulsa diterjemahkan menjadi putaran shaft, dimana setiap putaran membutuhkan jumlah pulsa yang ditentukan. Satu pulsa menghasilkan satu kenaikan putaran atau step, yang merupakan bagian dari satu putaran penuh. Oleh karena itu, perhitungan jumlah pulsa dapat diterapkan untuk mendapatkan jumlah putaran yang diinginkan. Gambar 2.5 merupakan perhitungan pulsa secara otomatis menunjukkan besarnya putaran yang telah dilakukan, tanpa memerlukan informasi balik (*feedback*).



Gambar 2.3 Bentuk pulsa motor stepper

Ketepatan kontrol gerak motor stepper terutama dipengaruhi oleh jumlah step tiap putaran; semakin banyak jumlah step, semakin tepat gerak yang dihasilkan. Untuk ketepatan yang lebih tinggi, beberapa *driver* motor stepper membagi *step* normal menjadi setengah *step* (*half step*) atau mikro *step*.

Bagian-bagian dari motor stepper yaitu tersusun atas rotor, stator, bearing, *casing* dan sumbu. Sumbu merupakan pegangan dari rotor dimana sumbu merupakan bagian tengah dari rotor, sehingga ketika rotor berputar sumbu ikut berputar. Stator memiliki dua bagian yaitu pelat inti dan lilitan. Plat inti dari motor stepper ini biasanya menyatu dengan *casing*. *Casing* motor

stepper terbuat dari aluminium dan ini berfungsi sebagaiudukan bearing dan stator pemegangnya adalah boud sebanyak empat buah. Di dalam motor stapper memiliki dua buah bearing yaitu bearing bagian atas dan bearing bagian bawah.

Gambar 2.6 adalah bearing motor stepper.



Gambar 2.4 Bearing motor stepper

Pada motor stepper umumnya tertulis spesifikasi N_p (= pulsa / rotasi). Sedangkan kecepatan pulsa diekspresikan sebagai pps (= pulsa per second) dan kecepatan putar umumnya ditulis sebagai ω (= rotasi / menit atau rpm). Kecepatan putar motor stepper (rpm) dapat diekspresikan menggunakan kecepatan pulsa (pps) sebagai berikut.

$$\begin{aligned}\omega &= 60 \frac{pps}{N_p} [\text{rotasi / menit}] \\ &= \frac{60}{N_p} pps\end{aligned}$$

Oleh karena 1 rotasi = 360 o, maka tingkat ketelitian motor stepper dapat diekspresikan dalam rumus sebagai berikut.

$$\begin{aligned}\delta &= \frac{o}{\text{pulsa}} \\ \delta &= \frac{1}{N_p} [\text{rotasi / pulsa}] \times 360^\circ \\ \delta &= \frac{360^\circ}{N_p} [^\circ / \text{pulsa}]\end{aligned}$$

Pada dasarnya motor stepper dikelompokkan menjadi 3 jenis, yaitu:

a. *Permanent Magnet (PM)*

Sesuai namanya, Motor stepper berjenis PM memiliki rotor berupa magnet permanen. Biasanya memiliki kecepatan rendah, alat dengan torsi rendah dan sudut langkah besar, bias 45 atau 90 derajat.

b. *Variable Reluctance (VR)*

Motor stepper jenis ini memiliki bentuk rotor yang unik yaitu berbentuk silinder dan pada semua unitnya memiliki gerigi yang memiliki hubungan dengan kutub-kutub stator. Rotor pada magnet tipe ini tidak menggunakan magnet permanen. Stator terlilit oleh lilitan sehingga pada saat teraliri arus, stator akan menghasilkan kutub magnet. Jumlah gerigi pada rotor akan menentukan langkah atau step motor. Perbedaan motor stepper berjenis PM dengan VR yaitu motor berjenis VR memiliki torsi yang relatif lebih kecil dibanding dengan motor stepper berjenis PM. Hal lain yang dapat dilihat adalah sisa kemagnetan sangat kecil sehingga pada saat motor stepper tidak dialiri arus maka ketika diputar tidak ada torsi yang melawan. Sudut langkah motor stepper berjenis VR ini bervariasi yaitu sekitar sampai dengan 30° . Motor stepper berjenis VR ini memiliki torsi yang kecil. Sering ditemukan pada printer dan instrumen-instrumen pabrik yang ringan yang tidak membutuhkan torsi yang besar.

Motor mempunyai 3 pasang kutub stator (A, B, C) yang diset terpisah 15 derajat. Arus dialirkan ke kutub A melalui lilitan motor yang menyebabkan tarikan magnetic yang menyejajarkan gigi rotor ke kutub A. jika kita memberi energi ke kutub B maka akan menyebabkan rotor berputar 15 derajat sejajar

kutub B. proses ini akan berlanjut ke kutub C dan kembali ke kutub A searah dengan jarum jam.

c. Permanent Magnet – Hybrid (PM-H)

Permanent magnet hybrid merupakan penyempurnaan motor stepper di mana motor stepper ini memiliki kecepatan 1000 step/detik namun juga memiliki torsi yang cukup besar sehingga dapat dikatakan bahwa PM-H merupakan motor stepper kombinasi antara PM dan VR motor stepper. Motor hybrid mengkombinasikan karakteristik terbaik dari motor variable reluktansi dan motor magnet permanent. Motor ini dibangun dengan kutub stator yang banyak-gigi dan rotor magnet permanent. Motor hybrid standar mempunyai 200 gigi rotor dan berputar pada 1,8 derajat sudut step. Karena memperlihatkan torsi tinggi dan dinamis serta berputar dengan kecepatan yang tinggi maka motor ini digunakan pada aplikasi yang sangat luas.

Dilihat dari lilitannya motor stepper terbagi menjadi 2 jenis yaitu :

a. Motor Stepper Unipolar

Motor stepper unipolar terdiri dari dua lilitan yang memiliki *center tap*. *Center tap* dari masing masing lilitan ada yang berupa kabel terpisah ada juga yang sudah terhubung didalamnya sehingga *center tap* yang keluar hanya satu kabel. Untuk motor stepper yang *center tap*nya ada pada masing – masing lilitan kabel inputnya ada 6 kabel. Namun jika *center tap*nya sudah terhubung di dalam kabel inputnya hanya 5 kabel. *Center tap* dari motor stepper dapat dihubungkan ke pentanahan atau ada juga yang menghubungkannya ke +VCC hal ini sangat dipengaruhi oleh *driver* yang digunakan.

b. Motor Stepper Bipolar

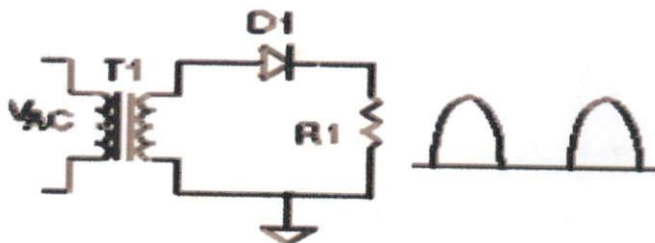
Motor stepper bipolar memiliki dua lilitan perbedaaan dari tipe unipolar adalah bahwa pada tipe bipolar lilitannya tidak memiliki *center tap*. Keunggulan tipe bipolar yaitu memiliki torsi yang lebih besar jika dibandingkan dengan tipe unipolar untuk ukuran yang sama. Pada motor stepper tipe ini hanya memiliki empat kabel masukan. Namun ntuk menggerakkan motor stepper tipe ini lebih rumit jika dibandingkan dengan menggerakkan motor stepper tipe unipolar.

2.2.2 Catu Daya ^[10]

Catu daya merupakan hal terpenting dari sebuah sistem kelistrikan dan elektronika. Catu daya berfungsi memberikan daya listrik kepada seluruh rangkaian pada sistem. Agar sistem dapat bekerja dengan baik maka catu daya harus dapat memberikan tegangan listrik yang stabil.

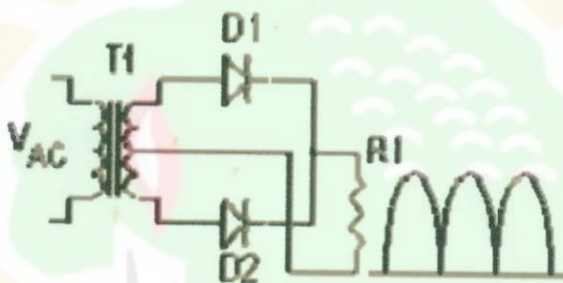
Perangkat elektronika biasanya dicatu oleh suplai arus searah DC (*direct current*). Batrai atau *accu* merupakan sumber catu daya DC yang paling baik. Namun untuk aplikasi yang membutuhkan daya yang lebih besar digunakan sumber arus bolak balik yang telah dirubah menjadi arus searah. Pada catu daya terdapat rangkaian yang mampu menyearahkan arus AC (*alternative current*) menjadi DC yang disebut dengan *rectifier* (penyearah).

Prinsip penyearah yang paling sederhana ditunjukkan pada gambar 2.7 berikut ini.



Gambar 2.5 Prinsip rangkaian penyearah

Transformator (T1) diperlukan untuk menurunkan tegangan AC dari jala-jala listrik pada kumparan primernya menjadi tegangan AC yang lebih kecil pada kumparan sekundernya. Dioda (D1) berperan hanya untuk merubah dari arus AC menjadi DC dan meneruskan tegangan positif ke beban R1. Ini yang disebut dengan penyearah setengah gelombang (*half wave*). Untuk mendapatkan penyearah gelombang penuh (*full wave*) diperlukan transformator dengan *center tap* (CT) seperti pada gambar 2.8 berikut ini.



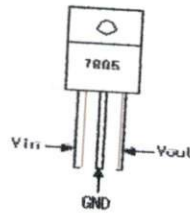
Gambar 2.6 Prinsip penyearah gelombang penuh

Tegangan positif fasa yang pertama dari gambar 2.8 diteruskan oleh D1 sedangkan fasa yang berikutnya dilewatkan melalui D2 ke beban R1 dengan CT transformator sebagai *common ground*. Dengan demikian beban R1 mendapat suplai tegangan gelombang penuh seperti gambar di atas.

Rangkaian penyearah sudah cukup bagus jika tegangan *ripple*-nya kecil, namun ada masalah stabilitas. Jika tegangan PLN naik/turun, maka tegangan outputnya juga akan naik/turun. Untuk beberapa aplikasi perubahan tegangan ini cukup mengganggu, sehingga diperlukan komponen aktif yang dapat meregulasi tegangan keluaran ini menjadi stabil.

Voltage regulator berfungsi sebagai filter tegangan agar sesuai dengan keinginan. Oleh karena itu biasanya dalam rangkaian catu daya, regulator

tegangan ini selalu dipakai untuk stabilnya output dari tegangan. Gambar 2.9 adalah susunan kaki regulator tersebut.



Gambar 2.7 Susunan kaki regulator

2.2.3 Sensor Jarak PING)))™ Ultrasonic Range Finder [11]

Sensor PING)))™ Ultrasonic Range Finder adalah sensor produksi Parallax yang banyak digunakan untuk aplikasi atau kontes robot. Kelebihan sensor ini adalah hanya membutuhkan 1 pin sinyal (SIG), selain jalur Vcc (5 Volt) dan Ground. Perhatikan Gambar 2.9 di bawah ini.

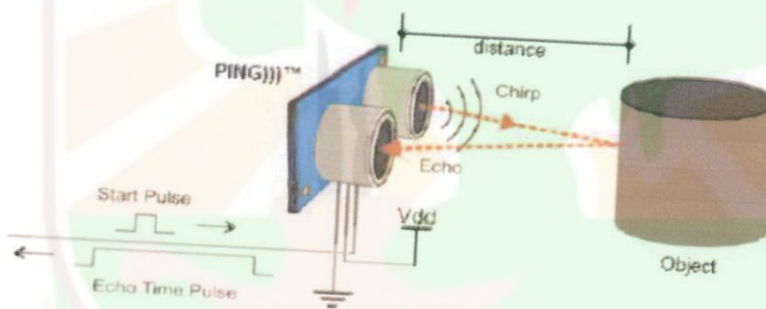


Gambar 2.8 Bentuk dan Konfigurasi Pin Sensor PING)))™

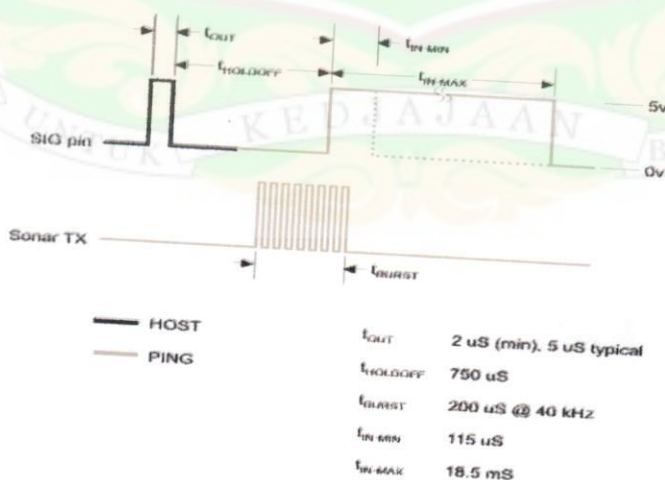
Karakteristik sensor PING)))™ yaitu:

- Kisaran pengukuran 3 cm – 300 cm.
- Frekuensi 40 kHz.
- *Input trigger-positive TTL pulse, 2 μs (Min), 5 μs tipikal.*
- *Echo hold off 750 μs dari fall off trigger pulse.*
- *Delay sebelum melakukan pengukuran selanjutnya 200 μs.*
- *Burst indikator LED menampilkan aktifitas sensor.*

Sensor PING)))™ mengukur jarak obyek dengan cara memancarkan gelombang ultrasonik (40 kHz) selama t_{BURST} (200 μ s) kemudian menunggu pantulannya. Sensor PING)))™ memancarkan gelombang ultrasonik sesuai dengan input kontrol dari pin SIG (pulsa trigger dengan t_{OUT} min. 2 μ s). Gelombang ultrasonik ini melalui udara dengan kecepatan kurang lebih 344 meter/detik, mengenai obyek dan memantul kembali ke sensor PING)))™. Modul PING)))™ akan mengeluarkan pulsa “high” pada pin SIG setelah memancarkan gelombang ultrasonik. Dan setelah gelombang pantulan terdeteksi, sensor PING)))™ akan membuat pin SIG “low”. Lebar pulsa “high” (t_{IN}) ini sesuai dengan lama waktu tempuh gelombang ultrasonik untuk dua kali jarak obyek, sehingga jarak obyek yang terukur adalah $[(t_{IN} \text{ s} \times 344 \text{ m/s}) \div 2]$ meter.



Gambar 2.9 Ilustrasi Cara Kerja Sensor PING)))™



Gambar 2.10 Diagram Waktu Sensor PING)))™

Sensor PING)))TM ini hanya membutuhkan 1 pin port mikrokontroler dan tidak memerlukan komponen tambahan. Karena 1 pin digunakan sebagai input dan output, maka harus menentukan saat yang tepat untuk mengatur port mikrokontroler sebagai input (dibuat "high") setelah mengeluarkan pulsa *trigger* ke sensor PING)))TM. Bila terlalu cepat atau terlalu lambat mengatur pin SIG sebagai input maka akan menyebabkan kesalahan pengukuran jarak. Sensor PING)))TM ini dapat digunakan untuk mengukur jarak benda sejauh 3 cm sampai 300 cm.

2.2.4 Sensor Encoder ^[12]

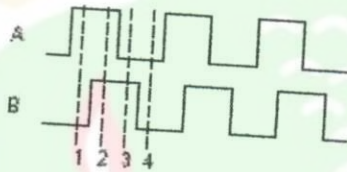
Incremental encoder digunakan untuk menghitung kecepatan, arah dan posisi relatif. Tachometer encoder satu kanal adalah *incremental* encoder yang paling sederhana. Encoder ini dipakai ketika tidak dibutuhkan informasi arah, sebagai contoh yaitu sebagai sensor umpan balik pada sistem kendali kecepatan. Sensor ini sangat cocok digunakan pada kecepatan menengah sampai dengan kecepatan tinggi, sedangkan untuk kecepatan rendah performansinya menurun karena jumlah pulsa yang dihasilkan untuk tiap satuan waktu sangat sedikit sehingga perhitungan kecepatan tidak akan akurat.

Menurut Rusydi (2004), agar diperoleh akurasi tinggi maka jumlah pulsa yang dihasilkan tiap putaran harus diperbanyak atau dengan kata lain resolusi sensor harus diperbesar.

Tabel 2.1 Keadaan *incremental* encoder dua kanal

Keadaan	Kanal A	Kanal B
K1	Tinggi	Rendah
K2	Tinggi	Tinggi

K3	Rendah	Tinggi
K4	Rendah	Rendah



Gambar 2.11 *Disk incremental encoder dua kanal dan sinyal keluarannya*

Tabel 2.1 adalah keluaran encoder incremental dua kanal seperti gambar 2.13, dimana kanal warna putih memberikan keluaran tinggi dan kanal warna hitam memberikan keluaran rendah.

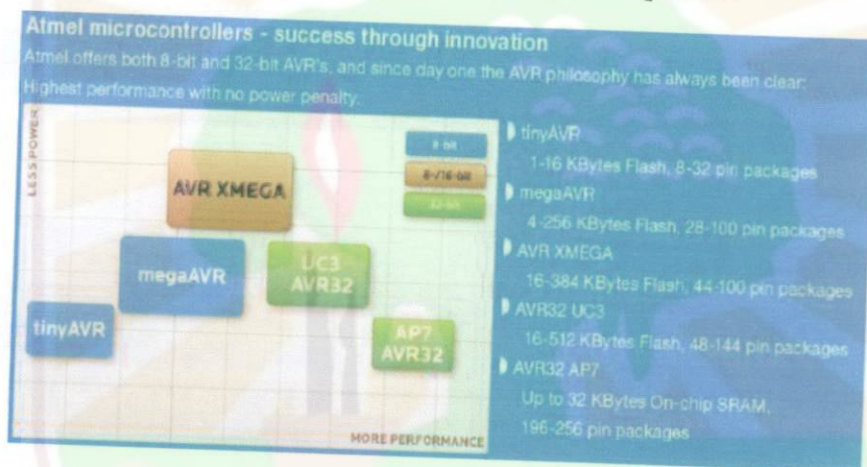
2.2.5 Mikrokontroler

Mikrokontroler adalah sebuah alat pengendali (kontroler) berukuran sangat kecil yang dikemas dalam sebuah chip. Sebuah mikrokontroler pada dasarnya bekerja seperti sebuah mikroprosesor pada computer. Keduanya memiliki sebuah CPU yang menjalankan instruksi program, melakukan logika dasar, dan pemindahan data. Namun agar dapat digunakan, sebuah mikroprosesor memerlukan tambahan komponen, seperti memori untuk menyimpan program dan data, juga interface input-output untuk berhubungan dengan dunia luar. Sebuah microcontroller telah memiliki memori dan *interface* input-output di dalamnya, bahkan beberapa mikrokontroler memiliki unit ADC yang dapat menerima

masukannya sinyal analog secara langsung. Karena berukuran kecil, murah dan menyerap daya yang rendah, mikrokontroler merupakan alat kontrol yang paling tepat untuk “ditanamkan” pada berbagai peralatan^[13].

2.2.5.1 Mikrokontroler ATMEL^[14]

Mikrokontroler keluaran ATMEL dapat dikatakan sebagai microcontroller terlaris dan termurah saat ini. Chip microcontroller ini dapat diprogram menggunakan port parallel atau serial. Selain itu, dapat beroperasi hanya dengan 1 chip dan beberapa komponen dasar seperti kristal, resistor, dan kapasitor.



Gambar 2.12 Perkembangan Mikrokontroler keluaran ATMEL

a. Tiny AVR

Mikrokontroler yang terdiri dari 8 sampai 32 pin dengan memory flash untuk menyimpan program hingga 16 K Bytes, dilengkapi SRAM dan EEPROM 512 Bytes.

b. Mega AVR

Mikrokontroler dengan unjuk kerja tinggi, dilengkapi pengali perangkat keras (*hardware multiplier*), mampu menyimpan program hingga 256 K Bytes dilengkapi EEPROM 4K Bytes dan SRAM 8K Bytes.

c. AVR XMEGA

Mikrokontroler 8/16-bit XMEGA memiliki peripheral baru dan canggih dengan unjuk kerja tinggi, Sistem event dan DMA yang ditingkatkan, serta merupakan pengembangan keluarga AVR untuk pasar *low power* dan *high performance*.

d. AVR32 UC3

Unjuk kerja tinggi, microcontroller flash AVR32 32-bit daya rendah. Memiliki flash hingga 512K Bytes dan SRAM 128K Bytes.

e. AVR32 AP7

Unjuk kerja tinggi, prosesor aplikasi AVR32 32-bit daya rendah, memiliki SRAM hingga 32K Bytes.

2.2.5.2 Mikrokontroler PIC^[15]

PIC ialah keluarga mikrokontroler tipe RISC (*Reduced Instruction Set Computing*) buatan Microchip Technology. Pada awalnya PIC dibuat menggunakan teknologi General instrument 16 bit CPU yaitu CP1600. PIC pertama kali dibuat pertama kali tahun 1975 untuk meningkatkan performa Sistem pada I/O. Saat ini PIC telah dilengkapi dengan EPROM dan komunikasi serial, UART, kernel control motor, serta memori program dari 512 word hingga 32 word. 1 word disini sama dengan 1 instruksi bahasa assembly yang bervariasi dari 12 sampai 16 bit tergantung dari PICmicro tersebut.

Beberapa fitur yang ada pada mikrokontroler PIC

1. PICF84

f. Instruksi 35 single word

g. Memori program 1024 word

- h. 68 Bytes Data RAM
- i. 64 byte Data EEPROM
- j. 14 bit wide instruction word
- k. 8 byte wide data
- l. 15 *Special Function Hardware Register*

2. PIC16F877

- ◆ 35 single word instruction
- ◆ Program memori 14 Kb
- ◆ 8 Chnnel ADC
- ◆ 368 bytes Data RAM
- ◆ 128 Bytes Data EEPROM
- ◆ 33 I/O
- ◆ 25mA sink/source per I/O
- ◆ Self Programming
- ◆ Parallel Slave Port

2.2.5.3 Mikrokontroler Renesas

Mikrokontroler Renesas yang dikenal ialah R8C/Tiny Series. Renesas sendiri adalah produsen semikonduktor yang merupakan gabungan dari Mitsubishi dan Hitachi. Mikrokontroler R8C/Tiny Series termasuk keluarga M16C, dan mempunyai beberapa seri antara lain R8C/10, R8C/11, R8C/12, dll. Masing-masing seri berbeda dalam besarnya flash ROM dan jumlah pin I/O. R8C memiliki beberapa fitur, antara lain:

- ◆ Flash ROM dan internal RAM relative cukup besar
- ◆ Sudah ada ON chip debugger

- ◆ Buit-in ADC 10 bit sebanyak 12 channel
- ◆ 2 buah UART, 1 bisa dipakai untuk mode sinkron
- ◆ Tomer 8 bit dan 1 timer 16 bit (dengan prescaler dan beberapa mode kerja)
- ◆ 2 buah rangkaian built-in untuk membangkitkan clock

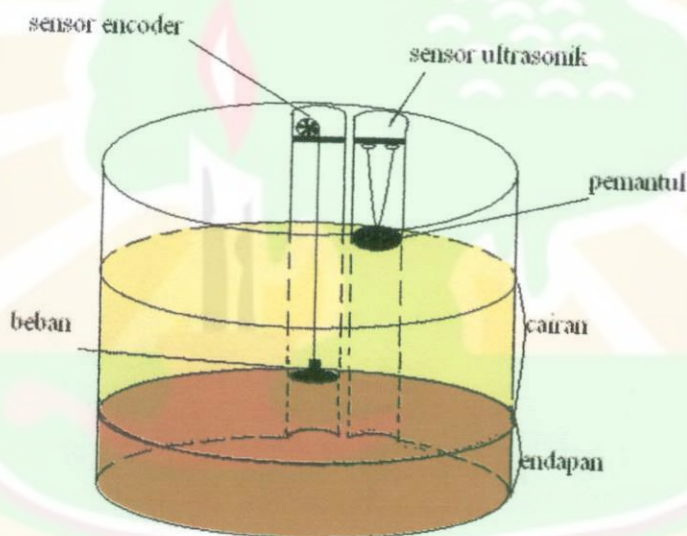


BAB III

METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metodologi yang digunakan dalam penelitian ini. Penjelasan tersebut mencakup garis besar sistem, perancangan alat, perancangan pelaporan informasi dari sensor ke komputer, perancangan pemrograman pergerakan sensor.

3.1 Gambaran Sistem



Gambar 3.1 *Gambaran Sistem Sensor Volume Bioetanol*

Secara garis besar alat ini merupakan perancangan sebuah sensor baru dengan memanfaatkan sensor-sensor yang telah ada. Sistem ini nantinya akan mendeteksi selisih dari jarak antara lempengan yang ada pada endapan adonan dengan lempengan yang berada di atas permukaan cairan. Hasil yang diperoleh nantinya akan diproses dan akan menghasilkan keluaran berupa volume dari

cairan yang ada pada tabung fermentasi seperti yang terlihat pada gambar 3.1 yaitu tabung pertama.

3.2 Metode Penelitian

Dalam penulisan Tugas Akhir, metode penelitian yang digunakan penulis adalah sebagai berikut:

1. Studi pustaka

Langkah pertama dalam pengerjaan Tugas Akhir adalah mencari referensi sebanyak mungkin untuk menunjang pengerjaan Tugas Akhir, baik itu berupa makalah penelitian, buku acuan, *datasheet*, laporan Tugas Akhir maupun bahan-bahan dari internet. Daftar literatur-literatur tersebut dapat dilihat pada daftar kepustakaan.

2. Pengumpulan referensi, meliputi:

- a. Pengumpulan teori mengenai pengolahan Bioethanol.
- b. Pengumpulan teori sensor ultrasonik.
- c. Pengumpulan teori motor DC servo.
- d. Pengumpulan teori dan pemrograman dengan bahasa C pada mikrokontroler Atmega8535.
- e. Pengumpulan teori komunikasi mikrokontroler dengan komputer.

3. Perancangan alat, meliputi:

- a. Merancang bentuk mekanik alat.
- b. Merancang bentuk rangkaian catu daya (*power supply*) dan rangkaian sensor.
- c. Merancang sistem minimum mikrokontroler Atmega8535.
- d. Merancang bentuk pemanfaatan sensor ultrasonik.

4. Pembuatan alat, meliputi:
 - a. Membuat bentuk mekanik alat.
 - b. Membuat rangkaian catu daya (*power supply*) dan rangkaian sensor.
 - c. Membuat rangkaian pengantarmukaan komunikasi alat dengan komputer.
5. Pembuatan program
Membuat perangkat lunak (*software*) pergerakan sensor dalam mendeteksi jarak dan pelaporan data yang didapat ke unit komputer.
6. Pengujian sistem, meliputi:
 - a. Pengujian kinerja sensor ultrasonik.
 - b. Pengujian ketepatan pengukuran volume oleh sensor ultrasonik.
7. Analisa data
Analisa yang dilakukan pada penelitian ini adalah pengujian ketepatan pengukuran volume oleh alat yang dirancang.
8. Pengambilan kesimpulan
Pengambilan kesimpulan dilakukan dengan melihat hasil dari pengujian sistem dan analisa yang telah dilakukan pada alat pendeteksi ketinggian.

BAB IV

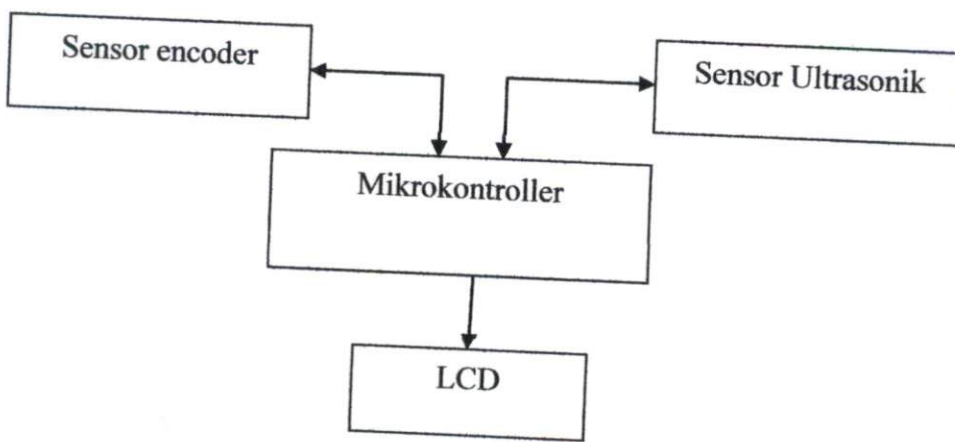
PERANCANGAN SISTEM

Bab perancangan dan implementasi sistem ini akan menjelaskan tentang cara kerja sistem yang terdapat dalam garis besar perancangan sistem dan diikuti dengan penjelasan tentang perancangan dan pembuatan perangkat keras (*hardware*) yang terdiri dari perancangan mekanik dan rangkaian elektrik yang digunakan pada *sistem pengukur volume cairan*. Kemudian diikuti dengan perancangan dan pembuatan perangkat lunak (*software*) yang akan berfungsi untuk mengatur atau mengendalikan *sistem pengukur volume cairan pada tabung fermentasi pada proses pembuatan bioetanol*.

4.1 Garis Besar Perancangan Sistem

Pada bagian ini akan dijelaskan secara umum bagaimana sistem dapat bekerja dengan baik. Perancangan ini meliputi perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*).

Pada sistem pengukur volume cairan ini digunakan mikrokontroler ATMega 8535 sebagai pusat pengendali *sistem plan* dengan kata lain mikrokontroler berperan sebagai otak sistem. Input pada sistem ini adalah sensor ultrasonik dan sensor encoder, sedangkan outputnya adalah jumlah volume cairan yang terdapat pada tabung fermentasi tersebut. Untuk gambaran sistem secara umum dapat dilihat pada gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

Secara umum cara kerja dari perancangan ini adalah mikrokontroler akan membaca input dari sensor ultrasonik dan sensor encoder. Sensor ultrasonic akan memberitahukan kepada mikrokontroler jarak yang terukur, jarak ini merupakan tinggi total dari isi tabung fermentasi. Sementara itu sensor encoder juga akan menghasilkan sejumlah jarak yang merupakan tinggi dari endapan yang ada pada tabung fermentasi. Dari kedua tinggi yang dihasilkan tadi, mikrokontroler akan melakukan proses matematika untuk mendapatkan selisih tinggi yang merupakan tinggi cairan yang terdapat pada tabung.

Setelah mendapatkan tinggi cairan tersebut, maka mikrokontroler akan menghitung volume cairan yang ada berdasarkan persamaan volume wadah yang telah ditentukan sebelumnya.

4.2 Perancangan Perangkat Keras

Perancangan perangkat keras (*Hardware*) terbagi menjadi dua bagian, yaitu perancangan mekanik dan perancangan elektrik. Perancangan mekanik terdiri dari perancangan sistem beban, perancangan sistem pelampung, dan perancangan tata letak sensor pada tabung fermentasi. Sedangkan perancangan

elektrik meliputi jenis sensor-sensor yang digunakan, sistem minimum mikrokontroler dan sumber tegangan (*power supply*).

4.2.1 Perancangan Mekanik

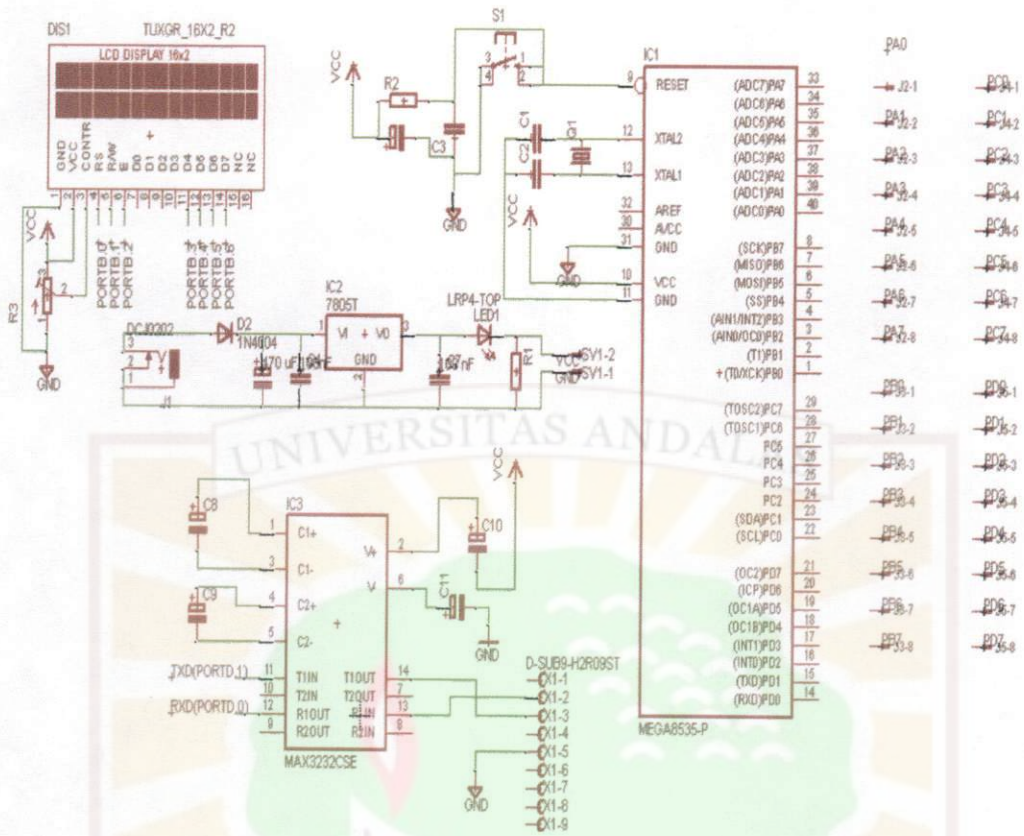
Perancangan mekanik dalam penelitian ini sangat penting karena mempengaruhi kinerja sistem secara keseluruhan. Perancangan mekanik meliputi perancangan sistem beban. Sistem beban merupakan salah satu hal yang terpenting karena berhubungan langsung dengan salah satu sensor yang digunakan. Sistem beban ini terdiri dari sebuah beban, seutas tali dan sebuah gulungan. Cara kerja beban adalah dengan memanfaatkan gaya gravitasi. Beban diletakkan pada posisi siap pada ketinggian tertentu yang nantinya akan dijatuhkan untuk bergerak menembus cairan dan berhenti ketika bertemu dengan padatan.

4.2.2 Perancangan Elektrik

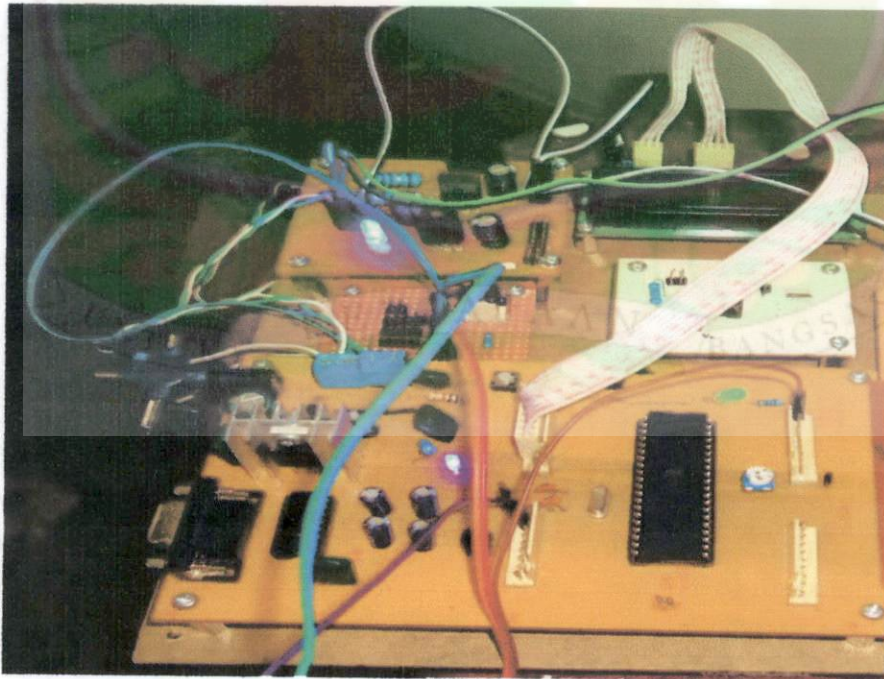
Perancangan elektrik yang diperlukan di antaranya yaitu rangkaian sistem minimum mikrokontroler, rangkaian sensor-sensor yang digunakan, rangkaian driver motor stepper, rangkaian driver motor dc dan rangkaian sumber tegangan (*power supply*).

4.2.2.1 Modul Mikrokontroler Atmega 8535

Modul Mikrokontroler ATmega 8535 terdiri dari sistem minimum ATmega 8535, power supply, dan rangkaian komunikasi serial RS 232 yang berada dalam 1 board PCB.



Gambar 4.2 Skema Rangkaian Sistem Minimum AVR ATmega 8535



Gambar 4.3 Modul Mikrokontroler ATmega 8535 pada satu board

4.2.2.1.1 Mikrokontroler AVR ATMEGA 8535

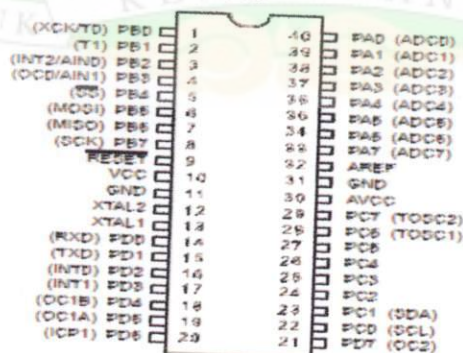
Mikrokontroler AVR (Alf and Vegard's Risc prosesor) memiliki arsitektur RISC 8 bit, di mana semua instruksi dikemas dalam kode 16-bit (16-bits word) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock, berbeda dengan instruksi MCS 51 yang membutuhkan 12 siklus clock. Tentu saja itu terjadi karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda. AVR berteknologi RISC (Reduced Instruction Set Computing), sedangkan seri MCS 51 berteknologi CISC (Complex Instruction Set Computing). Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Mikrokontroler ATMEGA 8535 merupakan mikrokontroler 8-bit keluaran ATMEL. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama. Oleh karena itu, dipergunakan salah satu AVR produk Atmel, yaitu ATmega8535. Selain mudah didapatkan dan lebih murah ATmega8535 juga memiliki fasilitas yang lengkap.

Arsitektur ATmega8535

- Saluran IO sebanyak 32 buah, yaitu Port A, Port B, Port C dan Port D
- ADC 10 bit sebanyak 8 Channel
- Tiga buah timer / counter
- 32 register
- Watchdog Timer dengan oscilator internal
- SRAM sebanyak 512 byte
- Memori Flash sebesar 8 kb
- Sumber Interrupt internal dan eksternal

- Port SPI (Serial Peripheral Interface)
 - EEPROM on board sebanyak 512 byte
 - Komparator analog
 - Port USART (Universal Synchronous Asynchronous Receiver Transmitter)
- Fitur ATmega8535
- Sistem processor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.
 - Ukuran memory flash 8KB, SRAM sebesar 512 byte, EEPROM sebesar 512 byte.
 - ADC internal dengan resolusi 10 bit sebanyak 8 channel
 - Port komunikasi serial USART dengan kecepatan maksimal 2.5 Mbps
 - Mode Sleep untuk penghematan penggunaan daya listrik

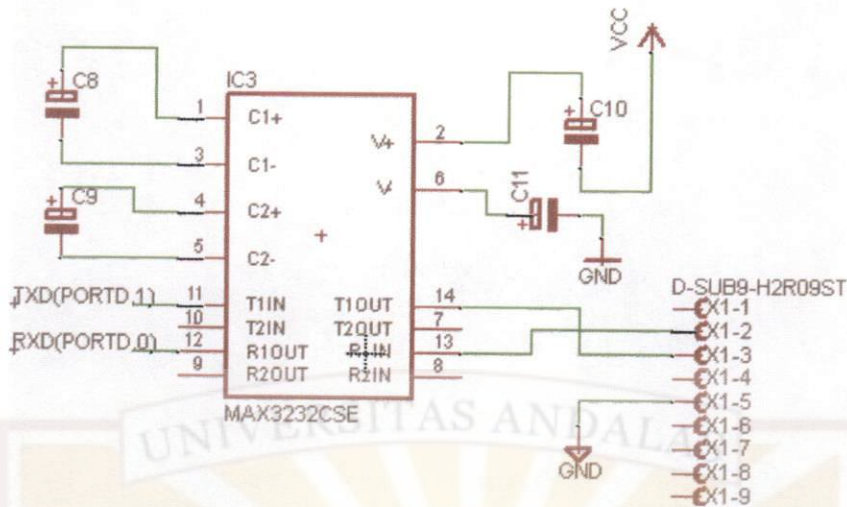
Untuk memprogram mikrokontroler dapat menggunakan bahasa assembler atau bahasa tingkat tinggi yaitu bahasa C. Bahasa yang digunakan memiliki keunggulan tersendiri, untuk bahasa assembler dapat diminimalisasi penggunaan memori program sedangkan dengan bahasa C menawarkan kecepatan dalam pembuatan program. Untuk bahasa assembler dapat ditulis dengan menggunakan text editor setelah itu dapat dikompilasi dengan tool tertentu misalnya asm51 untuk MCS51 dan AVR Studio untuk AVR.. Berikut ini merupakan gambar dan konfigurasi dari Mikrokontroler ATMEGA 8535



Gambar 4.4 PinOut Mikrokontroler ATMEGA 8535

4.2.2.1.3 Rangkaian Komunikasi Serial

Pada mikrokontroler AVR ATMEGA 8535 terdapat Pin-Pin (Port) untuk melakukan komunikasi serial yaitu Rx (Receive) dan Tx (Transmitte). Rx digunakan untuk mengirimkan data secara serial sedangkan Tx digunakan untuk menerima data secara serial pula. Komunikasi serial pada mikrokontroler ini masih menggunakan level sinyal TTL (Transistor Transistor Logic) yaitu sinyal yang memiliki gelombang level datanya antara 0 dan 5 volt. Dengan fasilitas Rx dan TX ini mikrokontroler bisa komunikasi secara serial baik antar devais atau dengan computer yang terhubung dengan rangkaian komunikasi serial yang dibuat. Dalam menggunakan mikrokontroler untuk berkomunikasi dengan komputer atau device lainnya maka Rx dan Tx tidak bisa langsung dihubungkan begitu saja dengan device tersbut karena level sinyal yang digunakan berbeda-beda. Contohnya komunikasi serial untuk komputer menggunakan sinyal RS232 yaitu sinyal yang gelombang level sinyalnya antara +25V sampai -25V. Oleh karena itu, jika ingin diharapkan terjadi komunikasi antara mikrokontroler dengan komputer dibutuhkan sebuah buffer yang dapat mengubah sinyal level TTL dari mikrokontroler menjadi sinyal level RS232. Salah Buffer yang sering digunakan adalah IC MAX232CPE.



Gambar 4.6 Skema Rangkaian Komunikasi Port Serial

4.2.2.1.4 LCD 2x 16

LCD 2x16 terdiri dari 2 baris dan 16 kolom. LCD modul ini bisa dipakai untuk interface dengan mikrokontroler / mikroprosesor 8 bit atau 4 bit bila menginginkan efisiensi pemakaian port. Pengiriman data ke LCD ada dua macam yaitu sebagai intruksi dan sebagai data character yang akan ditampilkan, keduanya dibedakan oleh sebuah kaki yang diberi nama RS(Register Select) dimana bila logika='1' (high) maka data yang diterima LCD adalah data character sedangkan bila RS='0' (Low) maka data yang diterima LCD adalah data intruksi. Intruksi diperlukan untuk initialisasi LCD, untuk meletakkan cursor pada baris dan kolom tertentu, untuk menghapus layar dll.

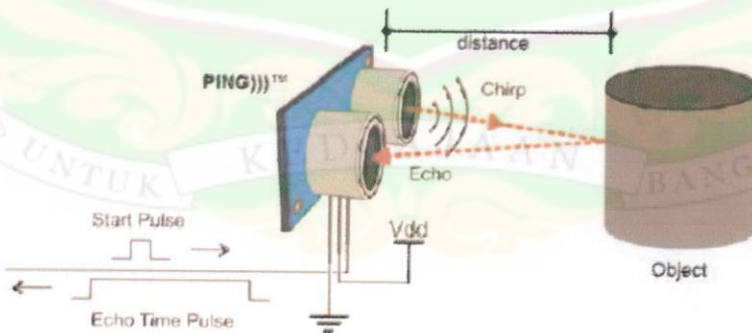
4.2.2.2 Sensor Ping Ultrasonic Range Finder

Ping Ultrasonic Range Finder merupakan modul pengukur jarak dengan gelombang ultrasonic buatan Parallax. Inc yang didesain khusus yang dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran dari sensor Ping berupa

pulsa yang lebarnya mempresentasikan jarak, lebar pulsanya bervariasi dari 115 us sampai 18,5 us. Pada dasarnya, Ping))) terdiri dari sebuah chip pembangkit sinyal 40KHz, sebuah speaker ultrasonik dan sebuah mikropon ultrasonik. Speaker ultrasonik mengubah sinyal 40 KHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya. Pada modul Ping))) terdapat 3 pin yang digunakan untuk jalur power supply (+5V), ground dan signal. Pin signal dapat langsung dihubungkan dengan Mikrokontroler tanpa tambahan komponen apapun. Berikut ini bentuk fisik dari sensor ping ultrasonic dan ilustrasi pengukuran jarak suatu objek.



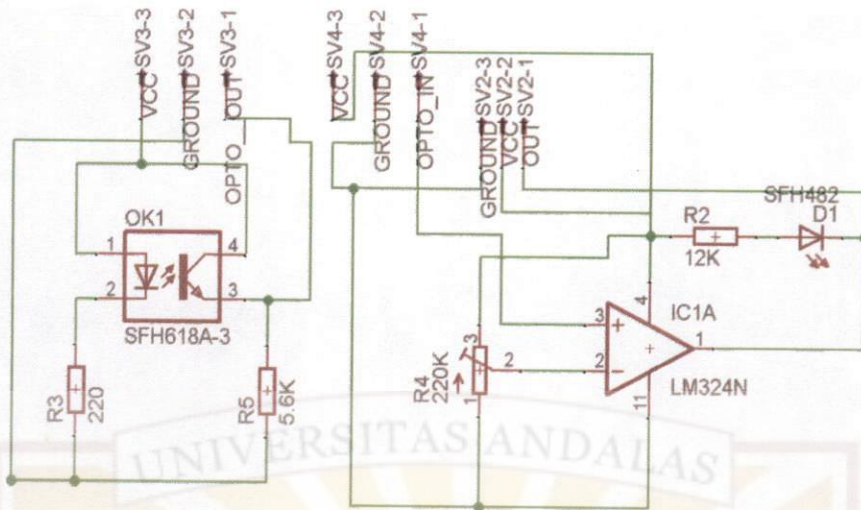
Gambar 4.7 Sensor Ping))) Ultrasonic Range Finder



Gambar 4.8 Skema Pengukuran

4.2.2.3 Sensor Encoder/

Incremental encoder merupakan suatu glass disk atau suatu piringan plastik berbentuk lingkaran yang terdapat pola berupa jalur-jalur daerah gelap dan terang pada permukaan juring piringan encoder. Daerah gelap dan terang ini merepresentasikan bit-bit 0 dan 1. Pada sensor volume ini digunakan incremental encoder dengan jumlah juring sebanyak 36 dengan 18 buah jumlah track/jalur berwarna hitam/gelap dan 18 buah jumlah track/jalur berwarna putih atau terang. Incremental encoder memiliki diameter 5cm dengan keliling 15,7 cm. Sehingga masing-masing juring pada incremental encoder ini memiliki keliling 0,436 cm. Jadi setiap perubahan bit dari 0 ke 1 atau sebaliknya akan dilakukan penambahan jarak sebesar 0,436 cm. Incremental encoder ini dikaitkan dengan pemberat, sehingga ketika penahan pemberat dilepaskan maka pemberat akan jatuh ke dasar tangki sampai batas padatan dari bahan baku. Pergerakan pemberat ini akan menggerakkan piringan encoder. Agar pengukuran dari encoder dapat terbaca oleh Mikrokontroler, maka encoder harus dihubungkan dengan rangkaian optocoupler. Optocoupler akan membaca perubahan track pada encoder. Perubahan posisi encoder dari daerah gelap ke terang atau sebaliknya akan menghasilkan keluaran berupa tegangan yang berbeda antara daerah gelap (bit 0) dan terang (bit 1). Karena nilai tegangan yang masih rendah maka nilai tegangan keluaran optocoupler tersebut dikuatkan menggunakan penguat yaitu LM 324N. Keluaran dari LM 324N inilah yang selanjutnya akan menjadi masukan pada pin interrupt (INT0) pada Mikrokontroler. Selanjutnya setiap perubahan bit akan mengaktifkan program interrupt eksternal. Berikut ini merupakan gambar dari incremental encoder beserta rangkain optocouplernya.

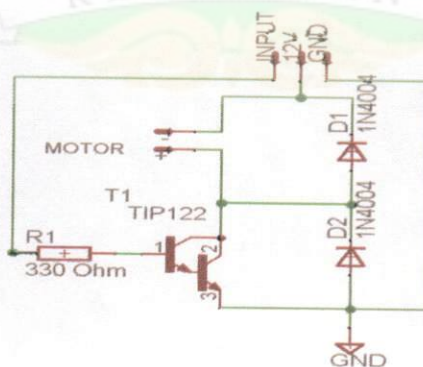


Gambar 4.9 Skema Rangkaian Optocoupler

Pergerakan encoder ini merepresentasikan jarak antara permukaan tangki dengan batas padatan dari bahan baku.

4.2.2.4 Motor DC

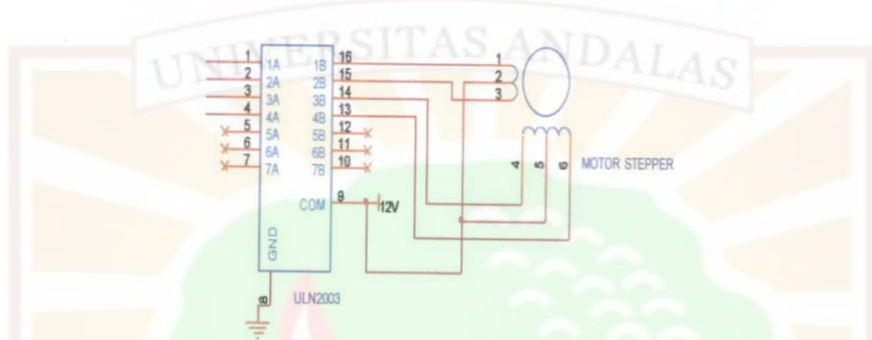
Motor DC pada sensor volume digunakan untuk menggerakkan gear yang akan menarik beban kembali ke atas permukaan tangki setelah selesai melakukan pengukuran. Motor DC ini diatur pergerakannya melalui PORT Mikrokontroler. Berikut merupakan gambaran dari motor DC yang digunakan beserta rangkaian driver yang mengatur pergerakannya.



Gambar 4.10 Skema Rangkaian Driver Motor DC

4.2.2.5 Motor Steper

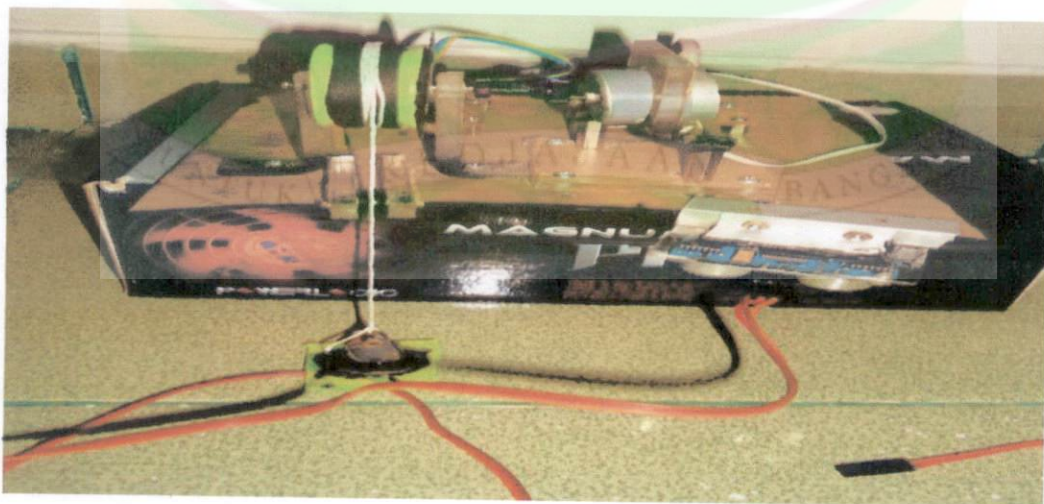
Motor stepper digunakan sebagai penggerak dari penahan beban. Pergerakan motor stepper akan menggerakkan penahan beban yang selanjutnya akan menjatuhkan beban ke dalam bahan baku. Pergerakan stepper ini diatur oleh Mikrokontroler. Berikut ini gambar dari Motor Stepper dan rangkain drivernya.



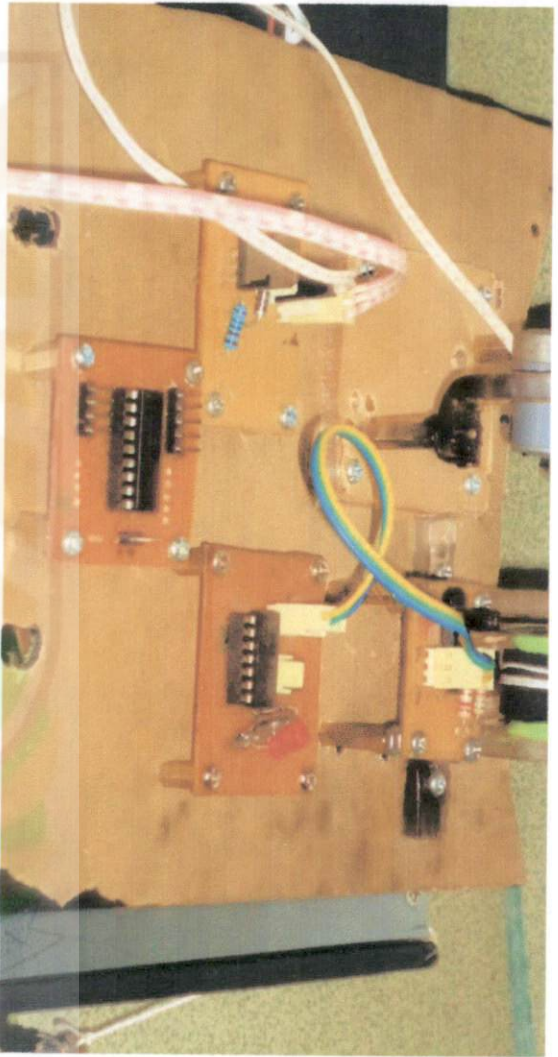
Gambar 4.11 Skema Rangkaian Driver Motor Stepper

4.2.2.6 Sistem Tata Letak Perangkat Keras

Perangkat keras yang telah didesain ini diletakkan dalam satu buah papan akirik yang terlihat seperti gambar-gambar berikut ini,



Gambar 4.12 Sensor Volume Cairan tampak depan



Gambar 4.13 Sensor Pengukur Volume Cairan tampak belakang



Gambar 4.14 Sensor Pengukur Cairan tampak samping

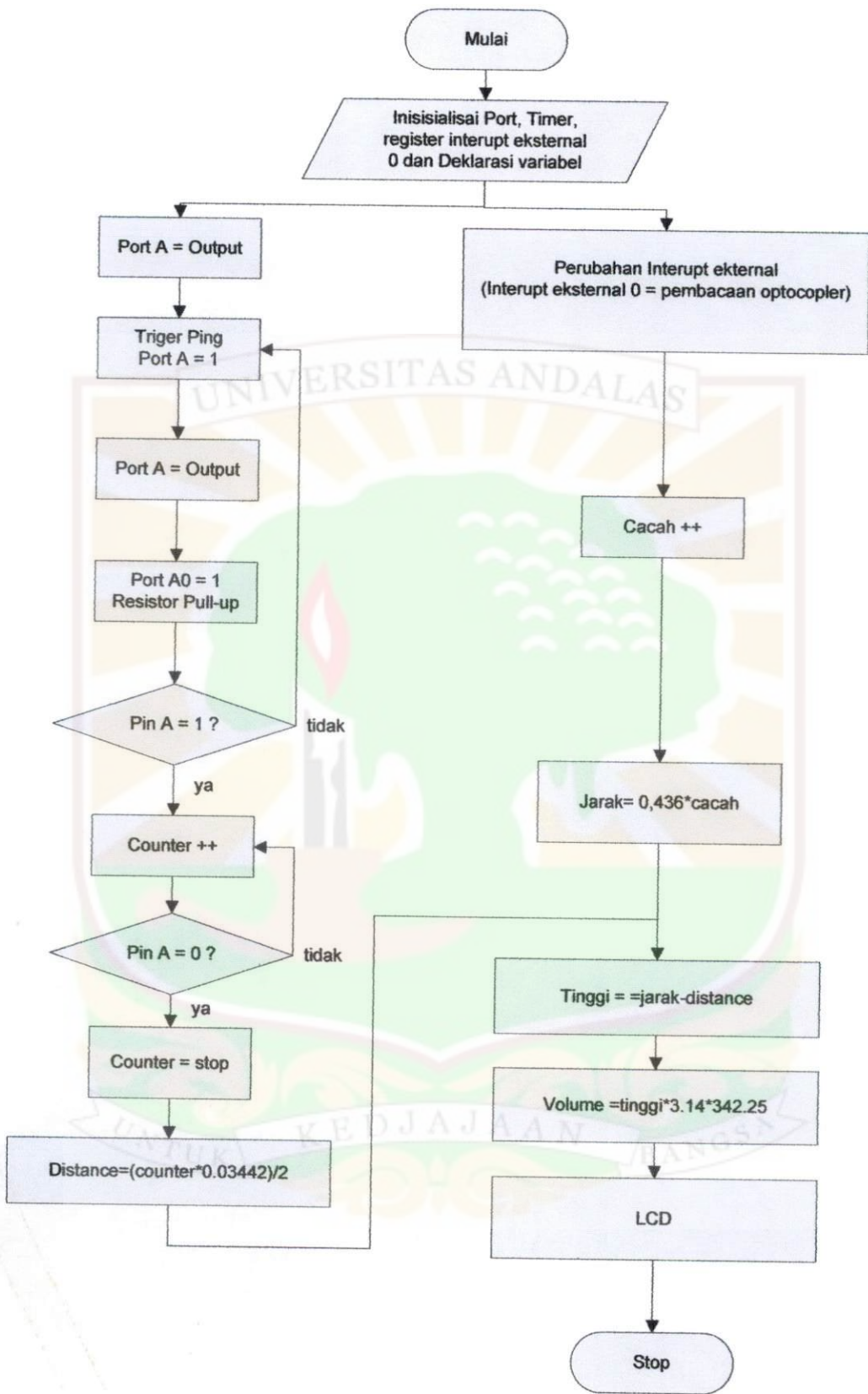
4.3 Perancangan Perangkat Lunak (Software)

Perancangan perangkat lunak ini secara garis besar bertujuan untuk mengatur kerja sistem seperti perintah ke sensor dan pembacaan hasil sensor.

Perancangan program utama ini berdasarkan algoritma kerja berikut :

1. Pengaktifan sensor ultrasonik dan optocoupler.
2. Pembacaan hasil pengukuran sensor Ultrasonik.
3. Pembacaan jumlah juring incremental encoder yang bergerak karena pengaruh tarikan beban.
4. Penghitungan jarak yang terbaca dari jumlah juring dengan cara mengalikan banyak juring dengan lebar tiap juring.
5. Pengurangan hasil pengukuran encoder dengan hasil pengukuran ultrasonik dicatat sebagai tinggi.
6. Perhitungan volume cairan dengan tinggi yang telah diperoleh dan luas alas wadah yang telah ditentukan.

Dari algoritma ini dapat dibuat menjadi program dengan diagram alir sebagai berikut :



Gambar 4.15 Diagram Alir sensor Pengukur Volume Cairan

BAB V

HASIL DAN ANALISA

5.1 Pengujian untuk Kalibrasi

Setelah semua komponen disatukan maka dibutuhkan suatu pengujian ataupun kalibrasi terhadap sensor pengukur volume ini. Proses ini dilakukan dengan cara manual yaitu mengisi sejumlah cairan pada suatu wadah dan dengan ketinggian air tertentu. Wadah yang digunakan adalah sebuah wadah berbentuk tabung dengan diameter 17 cm dan tinggi ember 25 cm. Dalam proses kalibrasi ini dimasukkan sejumlah endapan berupa pasir dan setelah itu ditetapkan tinggi air bervariasi yaitu 4 cm, 7 cm, dan 10 cm.

Dari masing-masing ketinggian air ini ditentukan volume air yang terdapat dalam wadah tersebut. Penghitungan volume menggunakan rumus volume tabung yaitu $V = \pi r^2 t$, dimana π (phi) adalah konstanta senilai 3.14, r adalah jari-jari wadah dan t adalah tinggi cairan.

5.1.1 Pengujian Ketinggian 4 Cm.

Untuk ketinggian 4 cm secara perhitungan didapatkan volume sebesar :

$$V = \pi r^2 t$$

$$V = 3.14 \times (8.5 \text{ cm})^2 \times 4 \text{ cm}$$

$$V = 3.14 \times 72.25 \text{ cm}^2 \times 4 \text{ cm}$$

$$V = 907.46 \text{ cm}^3$$

Sementara untuk hasil pengukuran dengan sensor dapat dilihat pada table 5.1.

Tabel 5.1 Pengujian Ketinggian 4 cm

No	Pengukuran tinggi cairan (t)	Selisih dengan ketinggian tetapan (Δt)	Pengukuran volume cairan (V)	Selisih dengan volume perhitungan (ΔV)
1	4.52 cm	0.52 cm	1048.26 cm ³	140.80 cm ³
2	4.32 cm	0.32 cm	948.78 cm ³	41.32 cm ³
3	3.44 cm	0.56 cm	781.13 cm ³	126.33 cm ³
4	4.18 cm	0.18 cm	948.25 cm ³	40.79 cm ³
5	4.20 cm	0.20 cm	953.80 cm ³	46.34 cm ³
	$\bar{t} = 4.132$ cm	$\overline{\Delta t} = 0.356$ cm	$\bar{V} = 936.04$ cm ³	$\overline{\Delta V} = 79.116$ cm ³

Persentase kesalahan pengukuran tinggi dapat dihitung dengan persamaan berikut :

$$\text{Error tinggi } (E_t) = \frac{\overline{\Delta t}}{t} \times 100 \%$$

$$E_t = \frac{0,356}{4} \times 100 \%$$

$$E_t = 8.9 \%$$

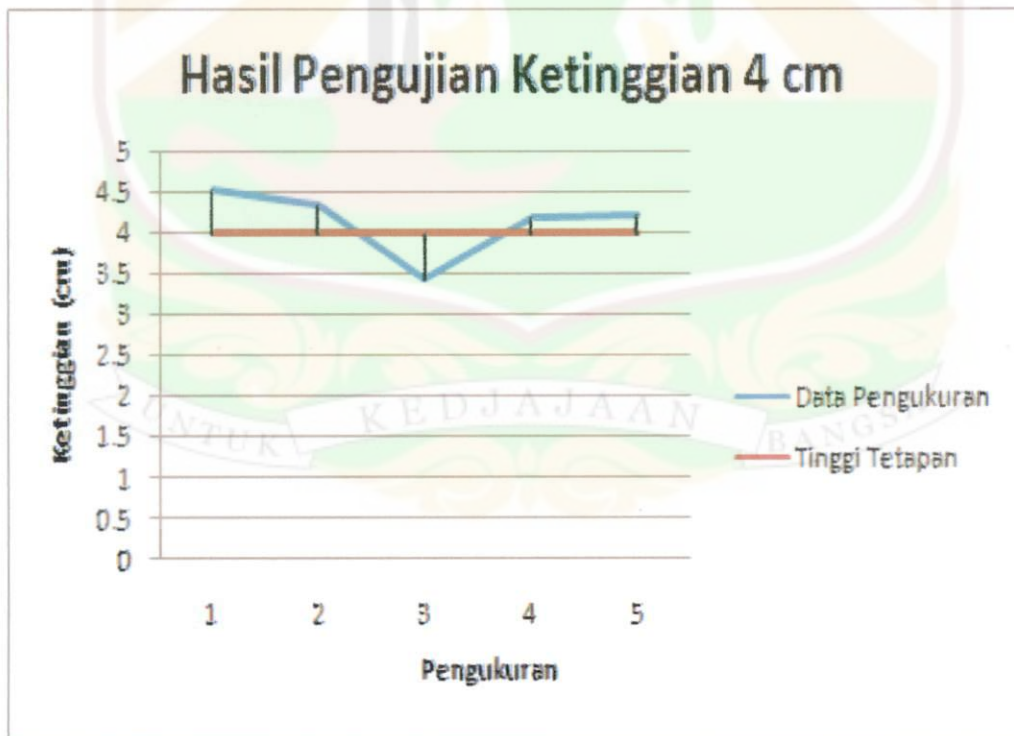
Persentase kesalahan pengukuran volume dapat dihitung dengan persamaan berikut :

$$\text{Error Volume } (E_v) = \frac{\overline{\Delta V}}{V} \times 100 \%$$

$$E_t = \frac{79,116}{907.46} \times 100 \%$$

$$E_t = 8.71 \%$$

Pada ketinggian 4 cm dari lima kali pengambilan data diperoleh rata-rata pengukuran sebesar 4.132 cm dan error rata-rata sebesar 0.356 cm sehingga diperoleh kinerja sistem sebesar $t_4 = 4.132 \pm 0.356$ cm. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan tinggi tetapan. Dari perhitungan tersebut didapatkan error sebesar 8,9 %. Untuk pengukuran volume juga dilakukan lima kali pengambilan data diperoleh rata-rata pengukuran sebesar 936.04 cm^3 dan error rata-rata sebesar $79,116 \text{ cm}^3$ sehingga diperoleh kinerja sistem sebesar $V_4 = 936.04 \pm 79,116 \text{ cm}^3$. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan volume perhitungan. Dari perhitungan tersebut didapatkan error sebesar 8,71 %. Hasil pengujian ini dapat dilihat pada gambar 5.1.



Gambar 5.1 Grafik Hasil Pengujian Ketinggian 4 cm

5.1.2 Pengujian Ketinggian 7 Cm.

Untuk ketinggian 7 cm secara perhitungan didapatkan volume sebesar :

$$V = \pi r^2 t$$

$$V = 3.14 \times (8.5 \text{ cm})^2 \times 7 \text{ cm}$$

$$V = 3.14 \times 72.25 \text{ cm}^2 \times 7 \text{ cm}$$

$$V = 1588.055 \text{ cm}^3$$

Sementara untuk hasil pengukuran dengan sensor dapat dilihat pada table 5.2.

Tabel 5.2 Pengujian Ketinggian 7 cm

No	Pengukuran tinggi cairan (t)	Selisih dengan ketinggian tetapan (Δt)	Pengukuran volume cairan (V)	Selisih dengan volume perhitungan (ΔV)
1	7.20 cm	0.20 cm	1667.42 cm ³	79.365 cm ³
2	7.25 cm	0.25 cm	1671.78 cm ³	83.725 cm ³
3	6.22 cm	0.78 cm	1411.25 cm ³	176.805 cm ³
4	6.27 cm	0.73 cm	1537.85 cm ³	50.205 cm ³
5	6.28 cm	0.72 cm	1425.77 cm ³	162.285 cm ³
	$\bar{t} = 6.644 \text{ cm}$	$\overline{\Delta t} = 0.536 \text{ cm}$	$\bar{V} = 1542.814 \text{ cm}^3$	$\overline{\Delta V} = 110.477 \text{ cm}^3$

Persentase kesalahan pengukuran tinggi dapat dihitung dengan persamaan berikut :

$$\text{Error tinggi } (E_t) = \frac{\overline{\Delta t}}{t} \times 100 \%$$

$$E_t = \frac{0.536}{7} \times 100 \%$$

$$E_t = 7.65 \%$$

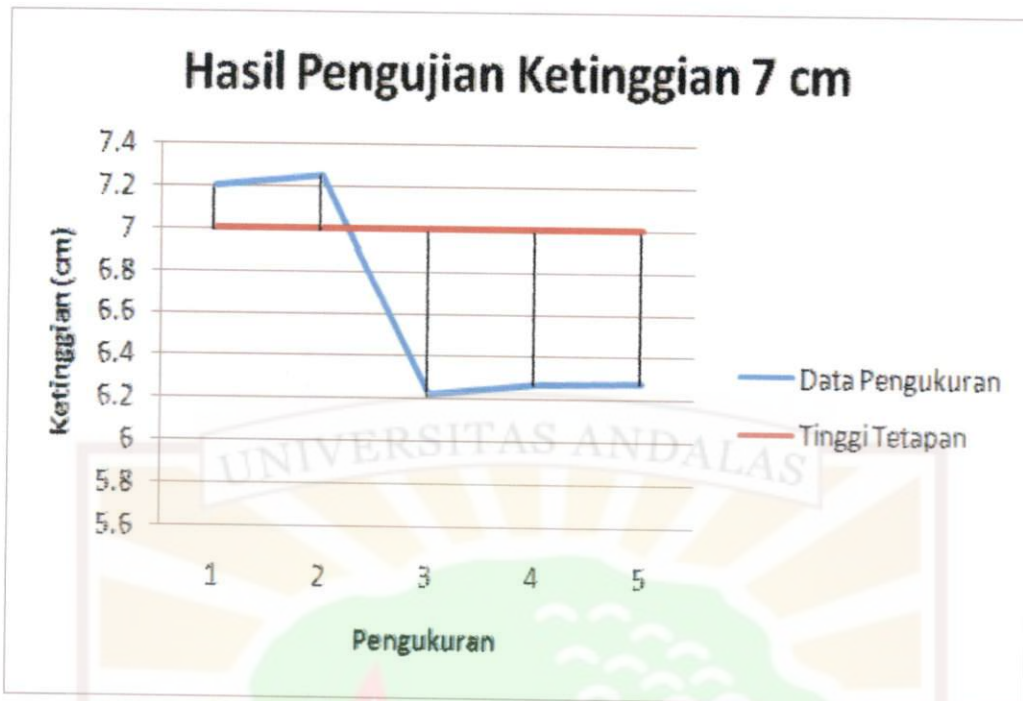
Persentase kesalahan pengukuran volume dapat dihitung dengan persamaan berikut :

$$\text{Error Volume (E}_v\text{)} = \frac{\overline{\Delta V}}{V} \times 100 \%$$

$$E_t = \frac{110.477}{1588.055} \times 100 \%$$

$$E_t = 6.93 \%$$

Pada ketinggian 7 cm dari lima kali pengambilan data diperoleh rata-rata pengukuran sebesar 6.644 cm dan error rata-rata sebesar 0.536 cm sehingga diperoleh kinerja sistem sebesar $t_7 = 6.644 \pm 0.536$ cm. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan tinggi tetapan. Dari perhitungan tersebut didapatkan error sebesar 7.65 %. Untuk pengukuran volume juga dilakukan lima kali pengambilan data diperoleh rata-rata pengukuran sebesar 1542.814 cm³ dan error rata-rata sebesar 110.477 cm³ sehingga diperoleh kinerja sistem sebesar $V_7 = 1542.814 \pm 110.477$ cm³. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan volume perhitungan. Dari perhitungan tersebut didapatkan error sebesar 6.93 %. Hasil pengujian ini dapat dilihat pada gambar 5.2.



Gambar 5.2 Grafik Hasil Pengujian Ketinggian 7 cm

5.1.3 Pengujian Ketinggian 10 Cm.

Untuk ketinggian 10 cm secara perhitungan didapatkan volume sebesar :

$$V = \pi r^2 t$$

$$V = 3.14 \times (8.5 \text{ cm})^2 \times 10 \text{ cm}$$

$$V = 3.14 \times 72.25 \text{ cm}^2 \times 10 \text{ cm}$$

$$V = 2268.65 \text{ cm}^3$$

Sementara untuk hasil pengukuran dengan sensor dapat dilihat pada table 5.1.

Tabel 5.3 Pengujian Ketinggian 10 cm

No	Pengukuran tinggi cairan (t)	Selisih dengan ketinggian tetapan (Δt)	Pengukuran volume cairan (V)	Selisih dengan volume perhitungan (ΔV)
1	9.85 cm	0.15 cm	2183.45 cm ³	85.20 cm ³
2	9.88 cm	0.12 cm	2203.16 cm ³	65.49 cm ³
3	7.68 cm	2.32 cm	1727.57 cm ³	541.08 cm ³
4	8.20 cm	1.80 cm	1938.22 cm ³	330.43 cm ³
5	9.68 cm	0.32 cm	2141.68 cm ³	126.97 cm ³
	$\bar{t} = 9.058$ cm	$\overline{\Delta t} = 0.942$ cm	$\bar{V} = 2038.816$ cm ³	$\overline{\Delta V} = 229.834$ cm ³

Persentase kesalahan pengukuran tinggi dapat dihitung dengan persamaan berikut :

$$\text{Error tinggi (E}_t\text{)} = \frac{\overline{\Delta t}}{t} \times 100 \%$$

$$E_t = \frac{0,942}{10} \times 100 \%$$

$$E_t = 9.42 \%$$

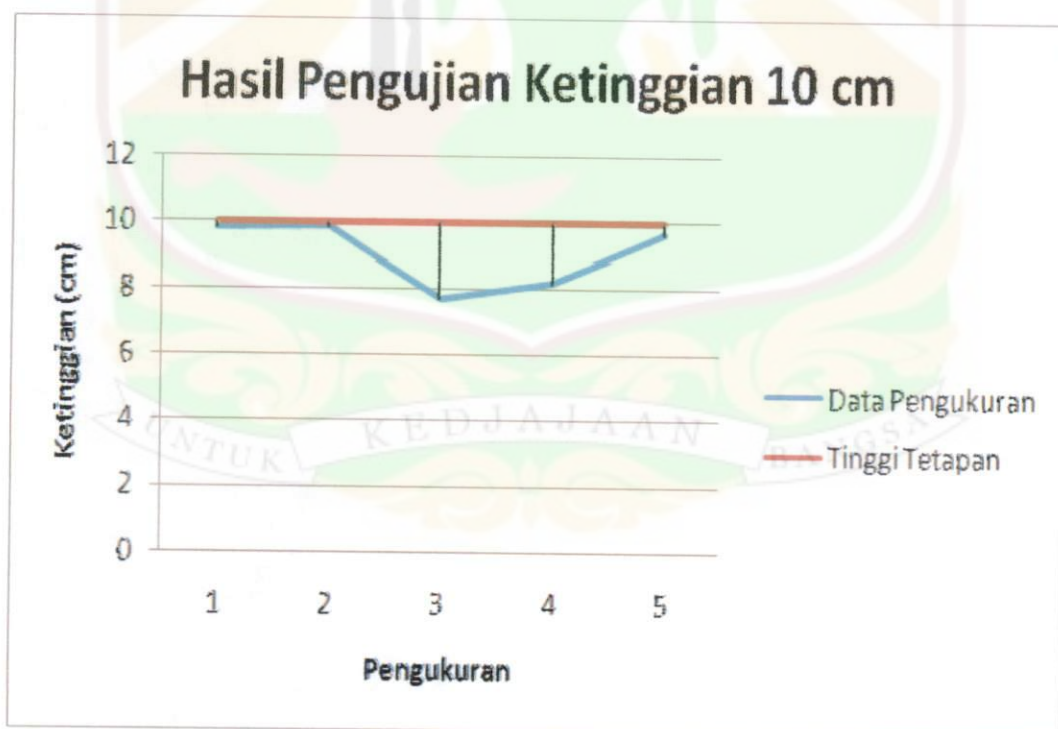
Persentase kesalahan pengukuran volume dapat dihitung dengan persamaan berikut :

$$\text{Error Volume (E}_v\text{)} = \frac{\overline{\Delta V}}{V} \times 100 \%$$

$$E_t = \frac{229.834}{2268.65} \times 100 \%$$

$$E_t = 10.13 \%$$

Pada ketinggian 10 cm dari lima kali pengambilan data diperoleh rata-rata pengukuran sebesar 9.058 cm dan error rata-rata sebesar 0.942 cm sehingga diperoleh kinerja sistem sebesar $t_{10} = 9.058 \pm 0.942$ cm. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan tinggi tetapan. Dari perhitungan tersebut didapatkan error sebesar 9.42 %. Untuk pengukuran volume juga dilakukan lima kali pengambilan data diperoleh rata-rata pengukuran sebesar 2038.816 cm³ dan error rata-rata sebesar 229.834 cm³ sehingga diperoleh kinerja sistem sebesar $V_{10} = 2038.816 \pm 229.834$ cm³. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan volume perhitungan. Dari perhitungan tersebut didapatkan error sebesar 10.13 %. Hasil pengujian ini dapat dilihat pada gambar 5.3.



Gambar 5.3 Grafik Hasil Pengujian Ketinggian 10 cm

5.2 Pengujian di Tabung Fermentasi

Pengambilan data dilakukan pada tabung pertama proses pembuatan bioetanol. Tabung I ini merupakan tempat fermentasi bahan baku ubi kayu yang akan diolah menjadi bioetanol. Tabung I ini memiliki ukuran diameter sebesar 37 cm dan tinggi 43 cm. Dengan ukuran dari tabung I, dapat dilakukan pengukuran volume cairan yang terdapat didalamnya.

Pengambilan data volume cairan ini dilakukan dengan jumlah bahan baku ubi kayu sebanyak 3 kg ubi kayu cincang yang telah dikeringkan. Sedangkan jumlah air yang digunakan untuk pencampuran ubi ini tidak ditentukan jumlahnya. Pengambilan data ini dilakukan setelah proses fermentasi dilakukan, didapatkan tinggi cairan dengan pengukuran secara manual adalah 8 cm dan kemudian dilakukan perhitungan dengan menggunakan rumus volume tabung didapatkan acuan volume cairan sebanyak :

$$V = \pi r^2 t$$

$$V = 3.14 \times (18.5 \text{ cm})^2 \times 8 \text{ cm}$$

$$V = 3.14 \times 342.25 \text{ cm}^2 \times 8 \text{ cm}$$

$$V = 8597.32 \text{ cm}^3.$$

Setelah itu dilakukan proses pengambilan data sebanyak 20 kali pengukuran. Data yang didapat seperti yang terlihat pada tabel 5.4.

MILIK
UPT PERPUSTAKAAN
UNIVERSITAS ANDALAS

Tabel 5.4 Pengukuran Tinggi dan Volume pada Tabung Fermentasi

No	t sensor (cm)	t manual (cm)	Error (Δt)	V sensor (cm^3)	V hitung manual (cm^3)	Error (ΔV)
1	7.61	8	0.39	8181.43	8597.32	415.89
2	8.92	8	0.93	9591.28	8597.32	993.96
3	6.96	8	1.04	7482.21	8597.32	1115.11
4	9.52	8	1.52	10013.04	8597.32	1415.72
5	12.85	8	4.85	13809.30	8597.32	5211.98
6	6.70	8	1.30	7204.78	8597.32	1392.54
7	7.10	8	0.90	7626.59	8597.32	970.73
8	9.61	8	1.61	10330.43	8597.32	1733.11
9	10.62	8	2.62	11410.90	8597.32	2813.58
10	7.66	8	0.34	8236.92	8597.32	360.4
11	7.97	8	0.03	8566.26	8597.32	31.06
12	7.36	8	0.64	7907.59	8597.32	698.73
13	7.05	8	0.95	7578.26	8597.32	1019.06
14	7.01	8	0.99	7530.54	8597.32	1066.78
15	9.01	8	1.01	9687.29	8597.32	1089.97
16	7.53	8	0.47	8096.12	8597.32	501.2
17	7.00	8	1.00	7522.76	8597.32	1074.56
18	8.93	8	0.93	9594.82	8597.32	997.5
19	8.09	8	0.09	8695.72	8597.32	98.4
20	9.63	8	1.63	10345.95	8597.32	1748.63
	$\bar{t} = 8.3565$ cm	t = 8 cm	$\overline{\Delta t} = 1.162$	$\bar{V} = 8970.61$ cm^3	V = 8597.32 cm^3	$\overline{\Delta V} = 1237.446$ cm^3

Keterangan :
 t sensor = pengukuran tinggi oleh sensor
 t manual = pengukuran tinggi secara manual
 V sensor = pengukuran volume oleh sensor
 V hitung manual = perhitungan volume dengan ukuran tinggi manual

Persentase kesalahan pengukuran tinggi dapat dihitung dengan persamaan berikut :

$$\text{Error tinggi } (E_t) = \frac{\overline{\Delta t}}{t} \times 100 \%$$

$$E_t = \frac{1.162}{8} \times 100 \%$$

$$E_t = 14.52 \%$$

Persentase kesalahan pengukuran volume dapat dihitung dengan persamaan berikut :

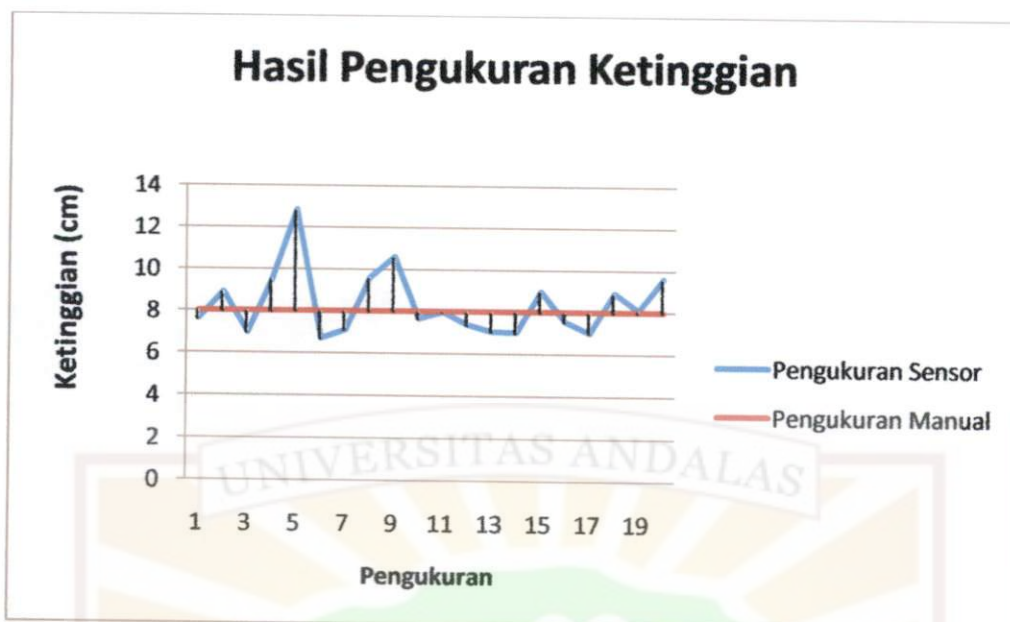
$$\text{Error Volume (E}_v\text{)} = \frac{\overline{\Delta V}}{V} \times 100 \%$$

$$E_t = \frac{1237.446}{8597.32} \times 100 \%$$

$$E_t = 14.39 \%$$

Dari 20 kali pengambilan data tinggi cairan pada tabung fermentasi ini diperoleh rata-rata pengukuran tinggi sebesar 8.3565 cm dan error rata-rata sebesar 1.162 cm sehingga diperoleh kinerja sistem sebesar $t_f = 8.3565 \pm 1.162$ cm. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan tinggi pengukuran manual. Dari perhitungan tersebut didapatkan error sebesar 14.52 %. Untuk pengukuran volume juga dilakukan 20 pengambilan data diperoleh rata-rata pengukuran sebesar 8970.61 cm³ dan error rata-rata sebesar 1237.446 cm³ sehingga diperoleh kinerja sistem sebesar $V_f = 8970.61 \pm 1237.446$ cm³. Untuk masing-masing pengukuran dilakukan pencarian error dengan mengambil nilai mutlak dari selisih pengukuran dengan volume perhitungan. Dari perhitungan tersebut didapatkan error sebesar 14.39 %

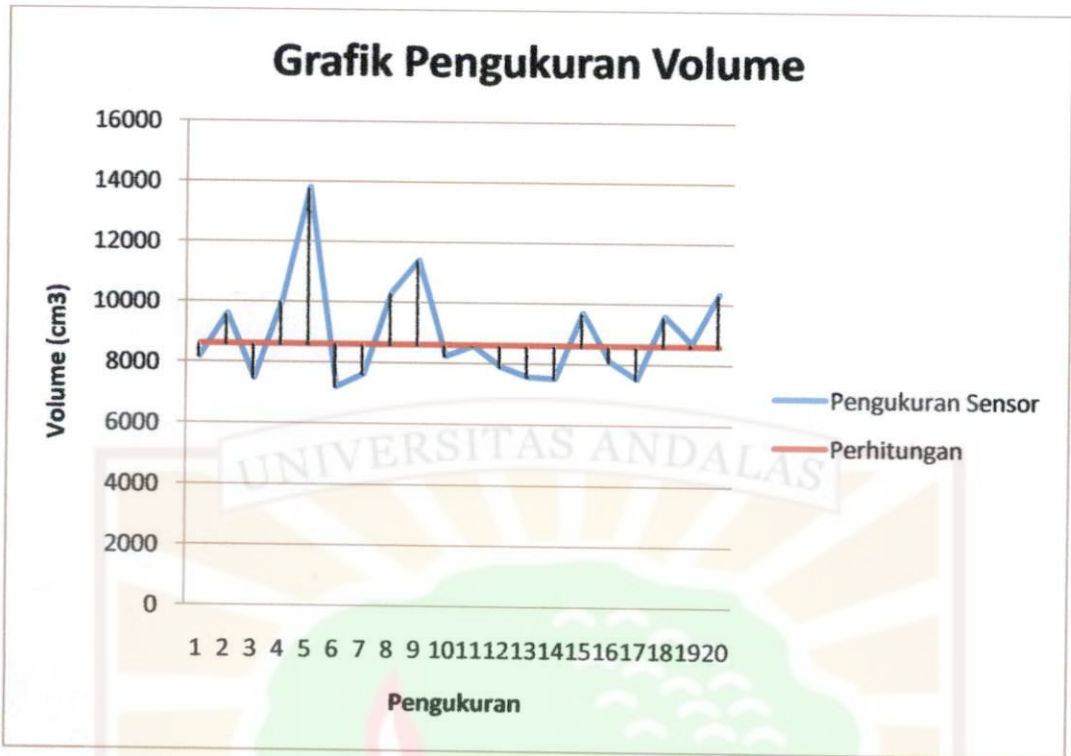
Dari 20 kali pengambilan data tinggi ini dibandingkan dengan pengukuran manual sebesar 8 cm. Data-data tersebut dapat dilihat pada grafik hasil pengukuran pada gambar 5.4.



Gambar 5.4 Grafik Hasil Pengukuran Ketinggian

Dari grafik dapat dilihat bahwa error paling besar didapatkan pada pengukuran kelima. Pengukuran kelima ini memperoleh nilai 12.85 cm, sedangkan pengukuran manual hanya 8 cm. terdapat selisih pengukuran sebesar 4.85 cm. sedangkan untuk pengukuran dengan error terkecil pada pengukuran kesebelas. Pengukuran ini menghasilkan nilai tinggi 7.97 cm hanya berselisih 0.03 cm dari pengukuran manual.

Pengukuran volume juga dilakukan sebanyak 20 kali sama seperti pengambilan data ketinggian. Dari 20 data tersebut dibandingkan dengan data perhitungan secara manual menggunakan pengukuran tinggi manual yaitu 8 cm, diperoleh volume cairan sebesar 8597.32 cm^3 . Data tersebut dapat dilihat pada grafik hasil pengukuran volume pada gambar 5.5.



Gambar 5.5 Grafik Hasil Pengukuran Volume

Dari grafik dapat dilihat bahwa error paling besar didapatkan pada pengukuran kelima. Pengukuran kelima ini memperoleh nilai 13809.30 cm^3 , sedangkan volume perhitungan manual hanya 8597.32 cm^3 . terdapat selisih sebesar 5211.98 cm^3 . sedangkan untuk pengukuran dengan error terkecil pada pengukuran kesebelas. Pengukuran ini memperoleh volume sebesar 8566.26 cm^3 hanya berselisih 31.06 cm^3 dari perhitungan manual.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan dari perancangan sistem yang telah dibuat dan hasil pengujian yang telah dilakukan, maka dapat diambil beberapa kesimpulan antara lain :

1. Pengujian ketinggian untuk kalibrasi dilakukan sebanyak tiga variasi ketinggian yaitu 4 cm, 7 cm, dan 10 cm didapatkan persentase error masing-masing yaitu 8.9 %, 7.5 %, dan 9.42 %.
2. Pengukuran tinggi cairan secara manual pada tabung fermentasi diperoleh tinggi sebesar ± 8 cm, sedangkan pengukuran tinggi oleh sensor yang dilakukan sebanyak 20 kali diperoleh error terbesar dipengukuran kelima yaitu 12.85 cm terdapat selisih sebesar 4.85 cm, dan untuk pengukuran dengan error terkecil pada pengukuran kesebelas yaitu 7.97 cm dengan selisih 0.03 cm. pengukuran tinggi ini menghasilkan persentase kesalahan pengukuran tinggi sebesar 14,52%
3. Dengan menggunakan tinggi cairan yang diperoleh pada pengukuran manual dilakukan perhitungan volume cairan yang terdapat pada tabung fermentasi yang diperoleh sebesar 8597.32 cm^3 , pengukuran volume dengan menggunakan sensor juga dilakukan sebanyak 20 kali dengan error terbesar dipengukuran kelima yaitu sebesar 13809.30 cm^3 terdapat selisih sebesar 5211.98 cm^3 . sedangkan untuk pengukuran dengan error terkecil pada pengukuran kesebelas sebesar 8566.26 cm^3

dengan selisih 31.06 cm^3 . pengukuran volume ini menghasilkan persentase kesalahan pengukuran volume sebesar $14,39 \%$.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka saran yang dapat diberikan untuk penelitian selanjutnya adalah untuk menganalisa fungsi kerja dari sensor ultrasonik dan sensor encoder agar dalam pengukuran volume selanjutnya dapat digunakan salah satu sensor saja.



DAFTAR PUSTAKA

- [1] Yakinudin, Andal. 2010. *Bioetanol Singkong Sebagai Sumber Bahan Bakar Terbaharukan dan Solusi untuk Meningkatkan Penghasilan Petani Singkong*. Bogor.
- [2] Rikana, Heppy, Reski Adam. 2010. *Tugas Akhir : Pembuatan Bioetanol dari Singkong Secara Fermentasi Menggunakan Rgai Tape*. Semarang.
- [3] “Bioetanol” , Diakses dari http://www.energiterbarukan.net/index.php?Itemid=43&id=27&option=com_content&task=view pada 2 Desember 2010 pukul 20:45 WIB.
- [4] “MEMBANGUN INDUSTRI BIOETANOL NASIONAL SEBAGAI PASOKAN ENERGI BERKELANJUTAN DALAM MENGHADAPI KRISIS ENERGI GLOBAL” , Diakses dari <http://mahasiswaanegarawan.wordpress.com/2007/08/18/membangun-industri-bioetanol-nasional-sebagai-pasokan-energi-berkelanjutan-dalam-menghadapi-krisis-energi-global/> pada 2 Desember 2010 pukul 20:45 WIB.
- [5] Rajman. 2010. *Etanol*. Sulawesi Tengah
- [6] Pitowarno, Endra. 2006. *Robotika, Desain, Kontrol, dan Kecerdasan Buatan*. Penerbit Andi: Yogyakarta.

- [7][12] Rusydi, Muhammad Ilhamdi.2004. *Tugas Akhir :Perancangan dan Implementasi Sistem Kendali PID Adaptif pada Pergerakan SYNCBOT*.Bandung.
- [8][11] Ermadi, Ade.2009. *Tugas Akhir :Perancangan dan Implementasi Robot Mobil Pendeteksi dan Pemadam Api Menggunakan Sensor Ultraviolet dan Ultrasonik Berbasis Mikrokontroler Renesas R8c/13*.Padang.
- [9] “Motor Stepper”. Diakses dari http://www.itelkom.ac.id/library/index.php?option=com_content&view=article&id=27:motor-stepper&catid=16:mikroprocessorkontroller&Itemid=15/ pada 6 Januari 2011 pukul 21:31 WIB.
- [10] Untari, Putri Lisma.2009.*Tugas Akhir : Perancangan dan Implementasi Kamera Penjejak Objek dengan Pengendali PID*.Padang.
- [13] Artanto, Dian. 2009. *Merakit PLC dengan Mikrokontroler*. PT Elex Media Komputindo: Jakarta.
- [14] Eko Putra, Agfianto dan Dhani Nugraha. 2010. *Free e-book Tutorial Pemrograman Mikrokontroler AVR dengan WinAVR GCC (ATMega16/32/8535)* Diakses dari <http://www.klikdisini.com/embedded> tanggal 21 Februari 2011 pukul 14.00 WIB.
- [15] Budiharto, Widodo. 2006. *Belajar Sendiri Membuat Robot Cerdas*. PT Elex Media Komputindo: Jakarta.
- [16] Budiharto, Widodo, Sigit Firmansyah.2005. *Elektronika Digital dan Mikroprosesor*. Pustaka Andi: Yogyakarta.

- [17] Pitowarno, Endra.2006. *Robotika, Desain, Kontrol, dan Kecerdasan Buatan*. Penerbit Andi: Yogyakarta.
- [18] Retnowati, Dwi, Rini Sutanti.2010.*Tugas Akhir : Pemanfaatan Limbah Padat Ampas Singkong dan Lindur Sebagai Bahan Baku Pembuatan Etanol*.Semarang
- [19] Supriyanto.2007.*Prospek Pengembangan Industri Bioetanol dari Ubi Kayu*.Balai Besar Teknologi Pati,BPPT : Jakarta.
- [20] Wiguna, Teguh.2007. *Tugas Akhir : Pengukur Volume Zat Cair Menggunakan Gelombang Ultrasonik Berbasis Mikrokontroller AT89S51*.Semarang.
- [21] "Proses Produksi Bioetanol", Diakses dari <http://www.alpensteel.com/article/51-113-energi-lain-lain/510-proses-produksi-bioetanol.html> pada 2 Desember 2010 pukul 20:45 WIB.
- [22] "PING))) ultrasonic range finder" Diakses dari <http://www.digi-ware.com/file/AN-07.pdf> pada 10 Januari 2011 pukul 19.45 WIB.

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

```
This program was produced by the  
CodeWizardAVR V2.03.4 Standard  
Automatic Program Generator  
© Copyright 1998-2008 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project :  
Version :  
Date : 03/11/2011  
Author :  
Company :  
Comments:
```

```
Chip type : ATmega8535  
Program type : Application  
Clock frequency : 12,000000 MHz  
Memory model : Small  
External RAM size : 0  
Data Stack size : 128
```

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

```
#include <mega8535.h>
```

```
// Alphanumeric LCD Module functions
```

```
#asm
```

```
.equ __lcd_port=0x18 ;PORTB
```

```
#endasm
```

```
#include <lcd.h>
```

```
unsigned char baris1 [33];
```

```
unsigned char baris2 [33];
```

```
unsigned int counter;
```

```
unsigned int cacah;
```

```
float distance;
```

```
float jarak;
```

```
float tinggi;
```

```
float volume;
```

```
// External Interrupt 0 service routine
```

```
interrupt [EXT_INT0] void ext_int0_isr(void)
```

```
{
```

```
// Place your code here
```

```
cacah+=1;
```

```
}
```

```
// Standard Input/Output functions
```

```
#include <stdio.h>
```

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

```
// Declare your global variables here
#define Sigout PORTA.0
#define Sigin PINA.0
#define Dirsig DDRA.0

void main(void)
{
// Declare your local variables here

// Reset Source checking
if (MCUCSR & 1)
{
// Power-on Reset
MCUCSR&=0xE0;
// Place your code here

}
else if (MCUCSR & 2)
{
// External Reset
MCUCSR&=0xE0;
// Place your code here

}
else if (MCUCSR & 4)
{
// Brown-Out Reset
MCUCSR&=0xE0;
// Place your code here

}
else if (MCUCSR & 8)
{
// Watchdog Reset
MCUCSR&=0xE0;
// Place your code here

}
else if (MCUCSR & 0x10)
{
// JTAG Reset
MCUCSR&=0xE0;
// Place your code here

};

// Input/Output Ports initialization
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;
```

```
// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OCO output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
```

```
// Timer/Counter 1 initialization
```

```
// 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;

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

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Any change
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x01;
MCUCSR=0x00;
GIFR=0x40;

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

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

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

// LCD module initialization
lcd_init(16);

// Global enable interrupts
```

```
#asm("sei")
```

```
while (1)
```

```
{  
  tinggi=0;  
  volume=0;  
  counter=0;  
  Dirsig=1;  
  Sigout=1;  
  delay_us(5);  
  Sigout=0;  
  Dirsig=0;  
  Sigout=1;
```

```
  while(Sigin==0)
```

```
  {}
```

```
  while(Sigin==1)
```

```
  {counter++;}
```

```
  distance=(counter*0.03442)/2;
```

```
  jarak= 0,436*cacah;
```

```
  tinggi=jarak-distance;
```

```
  volume=tinggi*3.14*342.25;
```

```
  sprintf(baris1,"Tinggi=%3.2f Cm", tinggi);
```

```
  sprintf(baris2,"Volume=%3.2f Cm",volume);
```

```
  lcd_gotoxy(0,0);
```

```
  lcd_puts(baris1);
```

```
  lcd_gotoxy(0,1);
```

```
  lcd_puts(baris2);
```

```
  delay_ms(200);
```

```
  // Place your code here
```

```
  lcd_clear();
```

```
};
```

```
}
```



Features

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

Advanced RISC Architecture

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

Nonvolatile Program and Data Memories

- 8K Bytes of In-System Self-Programmable Flash
Endurance: 10,000 Write/Erase Cycles
- Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation

512 Bytes EEPROM

- Endurance: 100,000 Write/Erase Cycles

512 Bytes Internal SRAM

- Programming Lock for Software Security

Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels for TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only

Byte-oriented Two-wire Serial Interface

Programmable Serial USART

Master/Slave SPI Serial Interface

Programmable Watchdog Timer with Separate On-chip Oscillator

On-chip Analog Comparator

Special Microcontroller Features

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

I/O and Packages

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

Operating Voltages

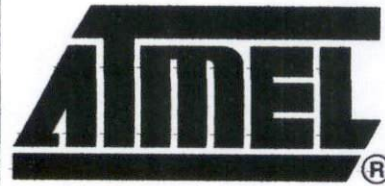
- 2.7 - 5.5V for ATmega8535L

- 4.5 - 5.5V for ATmega8535

Speed Grades

- 0 - 8 MHz for ATmega8535L

- 0 - 16 MHz for ATmega8535



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

ATmega8535
ATmega8535L

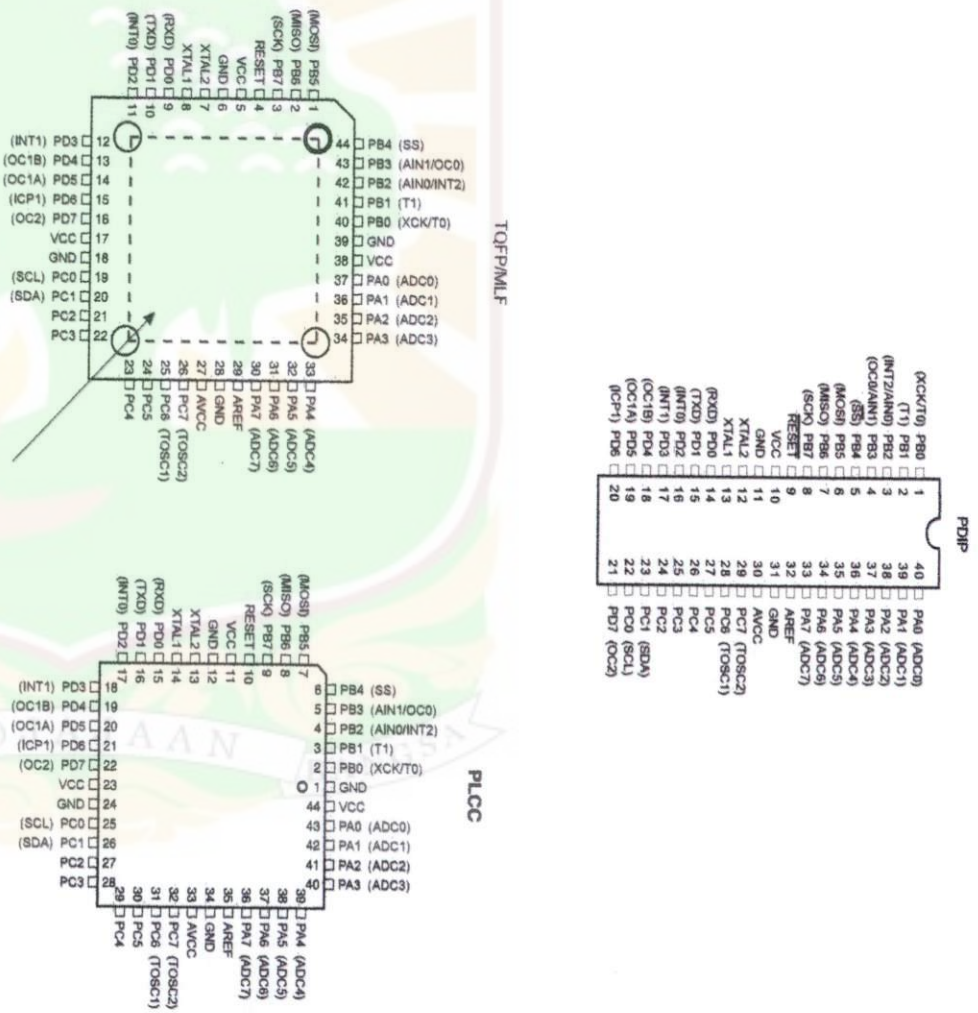
Summary

2502KS-AVR-10/06



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

Figure 1. Pinout ATmega8535



NOTE: MLF Bottom pad should be soldered to ground.

claimer

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

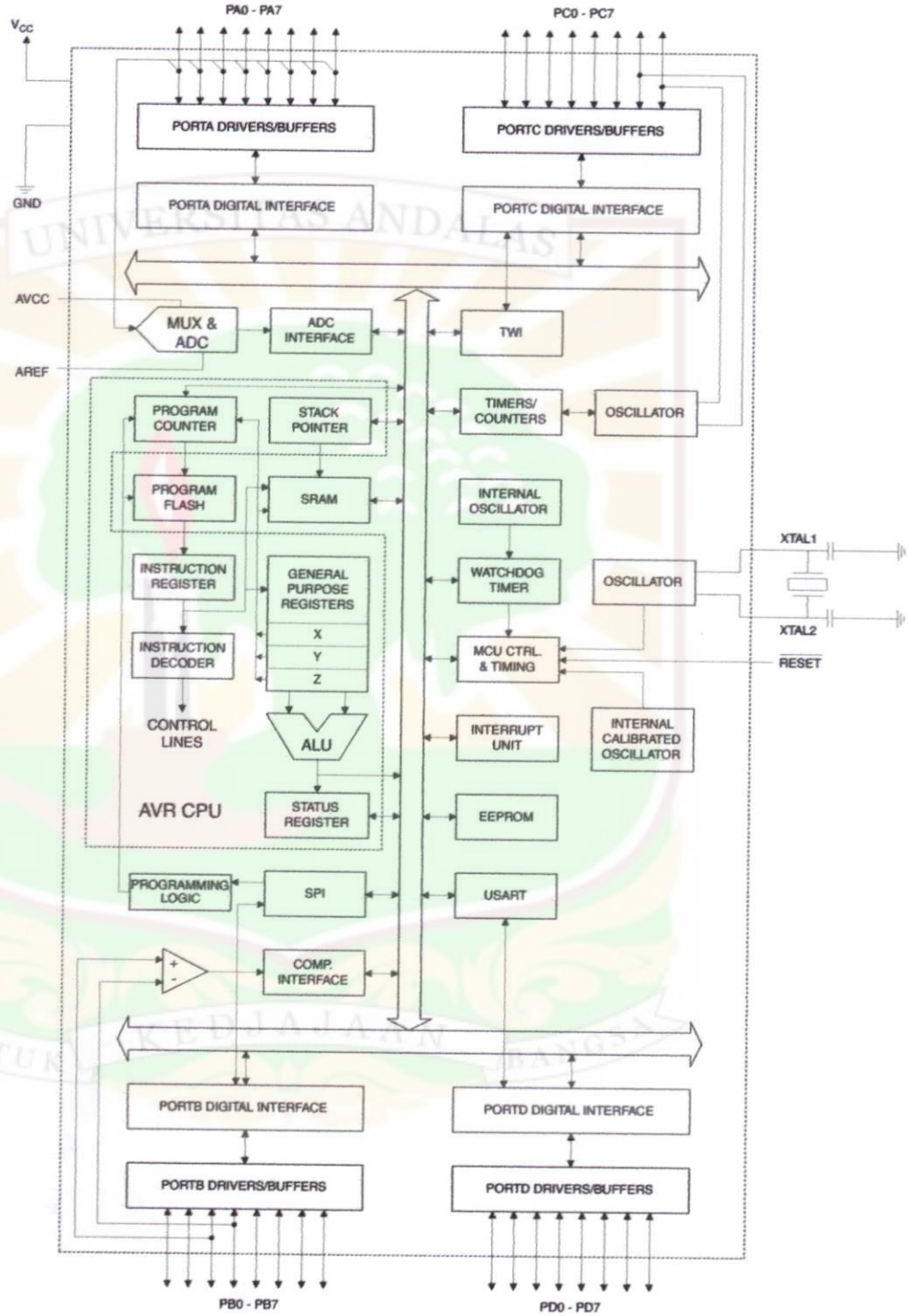
ATmega8535(L)

Overview

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

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All 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 ATmega8535 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain in TQFP package, 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 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 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 ATmega8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega8535 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.

AT90S8535 Compatibility

The ATmega8535 provides all the features of the AT90S8535. In addition, several new features are added. The ATmega8535 is backward compatible with AT90S8535 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S8535 compatibility mode can be selected by programming the S8535C fuse. ATmega8535 is pin compatible with AT90S8535, and can replace the AT90S8535 on current Printed Circuit Boards. However, the location of fuse bits and the electrical characteristics differs between the two devices.

AT90S8535 Compatibility

Programming the S8535C fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 45 for details.
- The double buffering of the USART Receive Register is disabled. See "AVR USART vs. AVR UART – Compatibility" on page 146 for details.

ATmega8535(L)

Descriptions

Digital supply voltage.

Ground.

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.

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 ATmega8535 as listed on page 60.

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.

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 ATmega8535 as listed on page 64.

ET

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 37. Shorter pulses are not guaranteed to generate a reset.

L1

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

L2

Output from the inverting Oscillator amplifier.

C

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.

F

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



about Code examples

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



Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
F (0x5F)	SREG	I	T	H	S	V	N	Z	C	10
E (0x5E)	SPH	–	–	–	–	–	–	SP9	SP8	12
D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
C (0x5C)	OCR0	Timer/Counter0 Output Compare Register								85
B (0x5B)	GICR	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	49, 69
A (0x5A)	GIFR	INTF1	INTF0	INTF2	–	–	–	–	–	70
9 (0x59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	85, 115, 133
8 (0x58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	86, 116, 134
7 (0x57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	228
6 (0x56)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	181
5 (0x55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	32, 68
4 (0x54)	MCUCSR	–	ISC2	–	–	WDRF	BORF	EXTRF	PORF	40, 69
3 (0x53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	83
2 (0x52)	TCNT0	Timer/Counter0 (8 Bits)								85
1 (0x51)	OSCCAL	Oscillator Calibration Register								30
0 (0x50)	SFIOR	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	59,88,135,203,223
F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	110
E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	113
D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								114
C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								114
B (0x4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								114
A (0x4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								114
9 (0x49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								114
8 (0x48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								114
7 (0x47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								114
6 (0x46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								114
5 (0x45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	128
4 (0x44)	TCNT2	Timer/Counter2 (8 Bits)								130
3 (0x43)	OCR2	Timer/Counter2 Output Compare Register								131
2 (0x42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	131
1 (0x41)	WDTCSR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	42
0 (0x40) ⁽¹⁾	UBRRH	URSEL	–	–	–	UBRR[11:8]				169
F (0x3F)	UCSRB	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	167
E (0x3E)	EEARH	–	–	–	–	–	–	–	EEARB	19
D (0x3D)	EEARL	EEPROM Address Register Low Byte								19
C (0x3C)	EEDR	EEPROM Data Register								19
B (0x3B)	EEDR	–	–	–	–	EERIE	EEMWE	EWE	EERE	19
A (0x3A)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	66
9 (0x39)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	66
8 (0x38)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	66
7 (0x37)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	66
6 (0x36)	DDRB	ddb7	ddb6	ddb5	ddb4	ddb3	ddb2	ddb1	ddb0	66
5 (0x35)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	67
4 (0x34)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	67
3 (0x33)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	67
2 (0x32)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	67
1 (0x31)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	67
0 (0x30)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	67
F (0x2F)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	67
E (0x2E)	SPDR	SPI Data Register								143
D (0x2D)	SPSR	SPIF	WCOL	–	–	–	–	–	SP12X	143
C (0x2C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	141
B (0x2B)	UDR	USART I/O Data Register								164
A (0x2A)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	165
9 (0x29)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	166
8 (0x28)	UBRRH	USART Baud Rate Register Low Byte								169
7 (0x27)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	203
6 (0x26)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	219
5 (0x25)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	221
4 (0x24)	ADCH	ADC Data Register High Byte								222
3 (0x23)	ADCL	ADC Data Register Low Byte								222
2 (0x22)	TWDR	Two-wire Serial Interface Data Register								183
1 (0x21)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	183
0 (0x20)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	183

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x00 (0x20)	TWBR	Two-wire Serial Interface Bit Rate Register								181

- Notes:
1. Refer to the USART description for details on how to access UBRRH and UCSRC.
 2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 3. 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 0x00 to 0x1F only.



Instruction Set Summary

Monics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
	Rd,K	Add Immediate to Word	$Rd \leftarrow Rd + K$	Z,C,N,V,S	2
	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
	Rd,K	Subtract Immediate from Word	$Rd \leftarrow Rd - K$	Z,C,N,V,S	2
	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \wedge Rr$	Z,N,V	1
	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \wedge K$	Z,N,V	1
	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
	Rd	One's Complement	$Rd \leftarrow \sim Rd$	Z,C,N,V	1
	Rd	Two's Complement	$Rd \leftarrow \sim Rd + 1$	Z,C,N,V,H	1
	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \wedge (\sim K)$	Z,N,V	1
	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \wedge Rd$	Z,N,V	1
	Rd	Clear Register	$Rd \leftarrow 0$	Z,N,V	1
	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
CONTROL INSTRUCTIONS					
	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
		Subroutine Return	$PC \leftarrow STACK$	None	4
		Interrupt Return	$PC \leftarrow STACK$	I	4
	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
	Rd,Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
	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
	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Equal	if (Z = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Not Equal	if (Z = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Same or Higher	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Lower	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Minus	if (N = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Plus	if (N = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Greater or Equal, Signed	if (N @ V = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Less Than Zero, Signed	if (N @ V = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Half Carry Flag Set	if (H = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Half Carry Flag Cleared	if (H = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if T Flag Set	if (T = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if T Flag Cleared	if (T = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Overflow Flag is Set	if (V = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Overflow Flag is Cleared	if (V = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Interrupt Enabled	if (I = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
	k	Branch if Interrupt Disabled	if (I = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
TRANSFER INSTRUCTIONS					

Mnemonic	Operands	Description	Operation	Flags	#Clocks
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
MOVW	Rd, Rr	Copy Register Word	$Rd+1:Rd \leftarrow Rr+1:Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LDD	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	2
LDD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LDD	Rd, -X	Load Indirect and Pre-Dec.	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LDD	Rd, Y	Load Indirect	$Rd \leftarrow (Y)$	None	2
LDD	Rd, Y+	Load Indirect and Post-Inc.	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LDD	Rd, -Y	Load Indirect and Pre-Dec.	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	$Rd \leftarrow (Y + q)$	None	2
LDD	Rd, Z	Load Indirect	$Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	2
LDD	Rd, -Z	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (k)$	None	2
ST	X, Rr	Store Indirect	$(X) \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
ST	Y+q, Rr	Store Indirect with Displacement	$(Y + q) \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$(Z) \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
ST	Z+q, Rr	Store Indirect with Displacement	$(Z + q) \leftarrow Rr$	None	2
ST	k, Rr	Store Direct to SRAM	$(k) \leftarrow Rr$	None	2
LD	Rd, Z	Load Program Memory	$Rd \leftarrow (Z)$	None	3
LD	Rd, Z+	Load Program Memory and Post-Inc	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	3
LD	Rd, P	In Port	$Rd \leftarrow P$	None	1
LD	P, Rr	Out Port	$P \leftarrow Rr$	None	1
LD	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2
LD	Rd	Pop Register from Stack	$Rd \leftarrow STACK$	None	2
AND BIT-TEST INSTRUCTIONS					
SBIW	P, b	Set Bit in I/O Register	$I/O(P, b) \leftarrow 1$	None	2
CBIW	P, b	Clear Bit in I/O Register	$I/O(P, b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z, C, N, V	1
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	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z, C, N, V	1
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z, C, N, V	1
SWAP	Rd	Swap Nibbles	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
SBIC	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
SBIC	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
SBIC	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
SBIC	Rd, b	Bit load from T to Register	$Rd(b) \leftarrow T$	None	1
SBIC		Set Carry	$C \leftarrow 1$	C	1
SBIC		Clear Carry	$C \leftarrow 0$	C	1
SBIC		Set Negative Flag	$N \leftarrow 1$	N	1
SBIC		Clear Negative Flag	$N \leftarrow 0$	N	1
SBIC		Set Zero Flag	$Z \leftarrow 1$	Z	1
SBIC		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SBIC		Global Interrupt Enable	$I \leftarrow 1$	I	1
SBIC		Global Interrupt Disable	$I \leftarrow 0$	I	1
SBIC		Set Signed Test Flag	$S \leftarrow 1$	S	1
SBIC		Clear Signed Test Flag	$S \leftarrow 0$	S	1
SBIC		Set Twos Complement Overflow	$V \leftarrow 1$	V	1
SBIC		Clear Twos Complement Overflow	$V \leftarrow 0$	V	1
SBIC		Set T in SREG	$T \leftarrow 1$	T	1
SBIC		Clear T in SREG	$T \leftarrow 0$	T	1
SBIC		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1
SBIC		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
CONTROL INSTRUCTIONS					
RAMPD		No Operation		None	1



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package ⁽¹⁾	Operation Range			
8	2.7 - 5.5V	ATmega8535L-8AC	44A	Commercial (0°C to 70°C)			
		ATmega8535L-8PC	40P6				
		ATmega8535L-8JC	44J				
		ATmega8535L-8MC	44M1				
		ATmega8535L-8AI	44A	Industrial (-40°C to 85°C)			
		ATmega8535L-8PI	40P6				
		ATmega8535L-8JI	44J				
		ATmega8535L-8MI	44M1				
		ATmega8535L-8AU ⁽²⁾	44A				
		ATmega8535L-8PU ⁽²⁾	40P6				
		ATmega8535L-8JU ⁽²⁾	44J				
		ATmega8535L-8MU ⁽²⁾	44M1				
		16	4.5 - 5.5V		ATmega8535-16AC	44A	Commercial (0°C to 70°C)
					ATmega8535-16PC	40P6	
ATmega8535-16JC	44J						
ATmega8535-16MC	44M1						
ATmega8535-16AI	44A			Industrial (-40°C to 85°C)			
ATmega8535-16PI	40P6						
ATmega8535-16JI	44J						
ATmega8535-16MI	44M1						
ATmega8535-16AU ⁽²⁾	44A						
ATmega8535-16PU ⁽²⁾	40P6						
ATmega8535-16JU ⁽²⁾	44J						
ATmega8535-16MU ⁽²⁾	44M1						

1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities..
2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

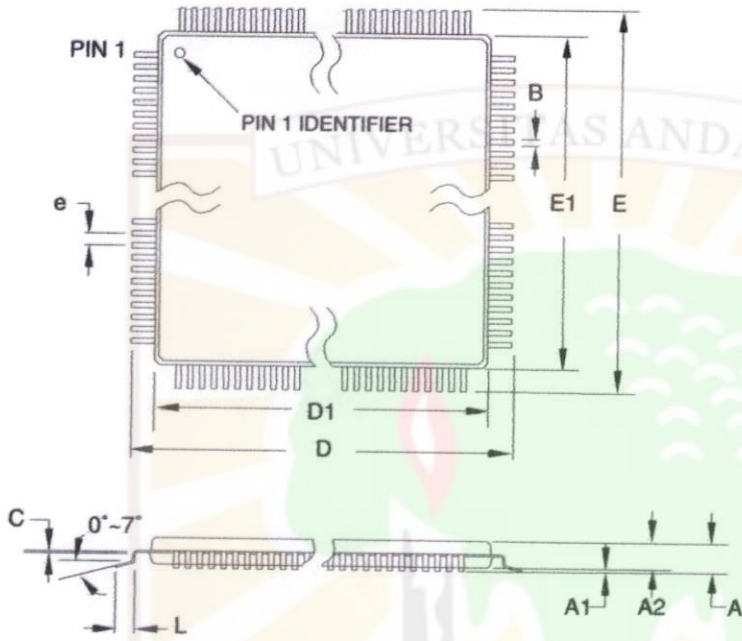
Package Type

44	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
	44-lead, Plastic J-leaded Chip Carrier (PLCC)
11-A	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

A



COMMON DIMENSIONS
(Unit of Measure = mm)

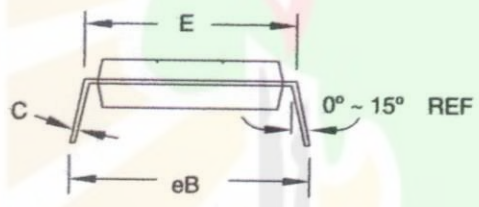
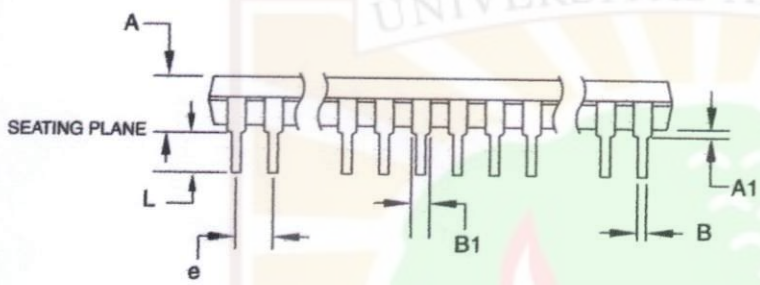
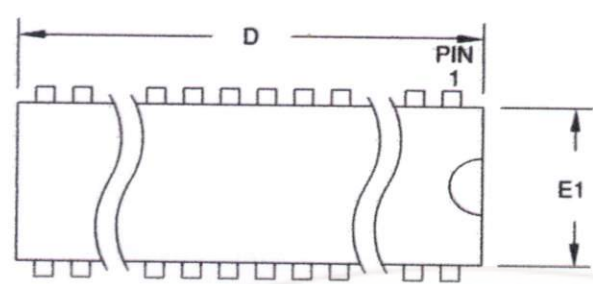
SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	1.20	
A1	0.05	-	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	-	0.45	
C	0.09	-	0.20	
L	0.45	-	0.75	
e	0.80 TYP			

- 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 Plastic Quad Flat Package (TQFP)	DRAWING NO.	REV.
		44A	B

ATmega8535(L)



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.826	
A1	0.381	-	-	
D	52.070	-	52.578	Note 2
E	15.240	-	15.875	
E1	13.462	-	13.970	Note 2
B	0.356	-	0.559	
B1	1.041	-	1.651	
L	3.048	-	3.556	
C	0.203	-	0.381	
eB	15.494	-	17.526	
e	2.540 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
 2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

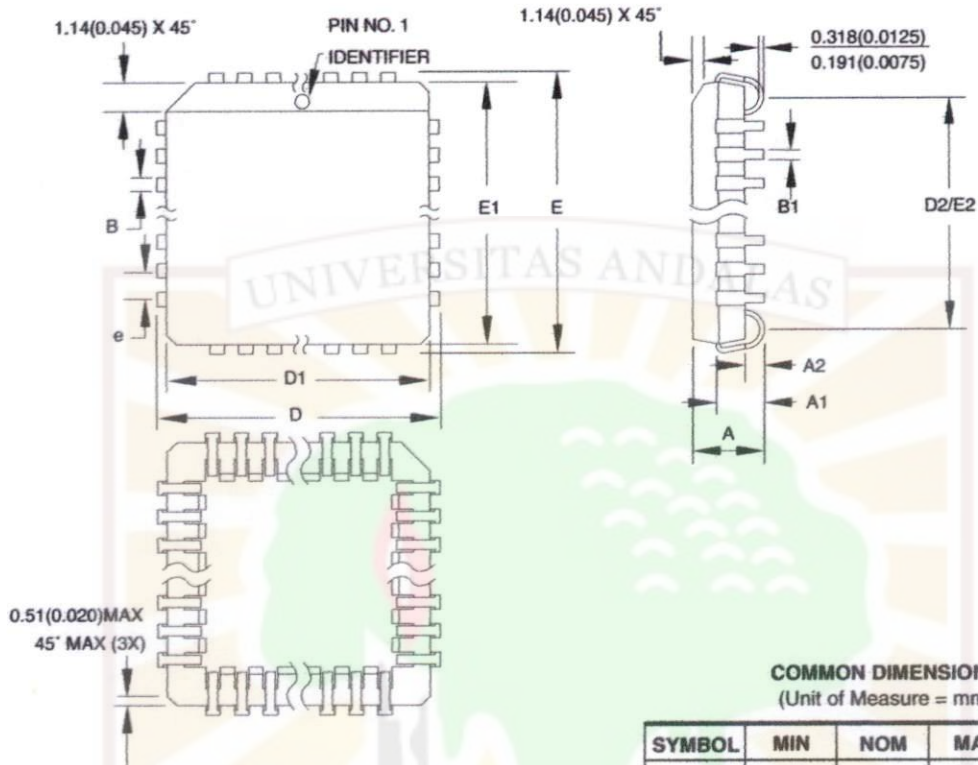
09/28/01

AMEL 2325 Orchard Parkway
San Jose, CA 95131

TITLE
40P6, 40-lead (0.600"/15.24 mm Wide) Plastic Dual In-line Package (PDIP)

DRAWING NO. 40P6
REV. B



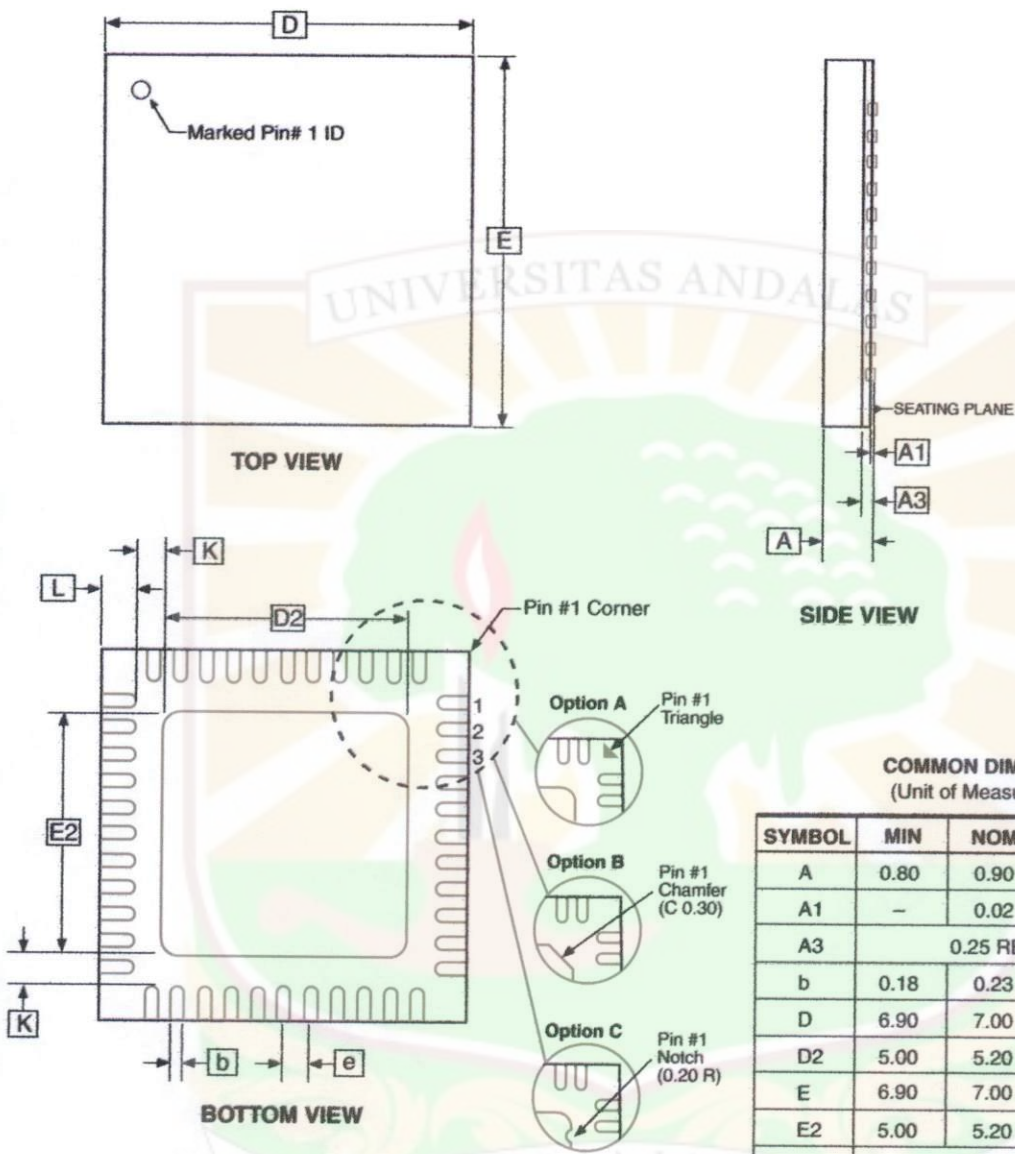


- Notes:
1. This package conforms to JEDEC reference MS-018, Variation AC.
 2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010" (0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
 3. Lead coplanarity is 0.004" (0.102 mm) maximum.

10/04/01

2325 Orchard Parkway San Jose, CA 95131	TITLE 44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)	DRAWING NO. 44J	REV. B
--	--	---------------------------	------------------

M1-A



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	-	0.02	0.05	
A3	0.25 REF			
b	0.18	0.23	0.30	
D	6.90	7.00	7.10	
D2	5.00	5.20	5.40	
E	6.90	7.00	7.10	
E2	5.00	5.20	5.40	
e	0.50 BSC			
L	0.59	0.64	0.69	
K	0.20	0.26	0.41	

Note: JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-3.

5/27/06

AMEL 2325 Orchard Parkway
San Jose, CA 95131

TITLE
44M1, 44-pad, 7 x 7 x 1.0 mm Body, Lead Pitch 0.50 mm,
5.20 mm Exposed Pad, Micro Lead Frame Package (MLF)

DRAWING NO. 44M1
REV. G



The revision letter refer to the device revision.

- **First Analog Comparator conversion may be delayed**
- **Asynchronous Oscillator does not stop in Power-down**

1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising V_{CC} , the first Analog Comparator conversion will take longer than expected on some devices.

Problem Fix/Workaround

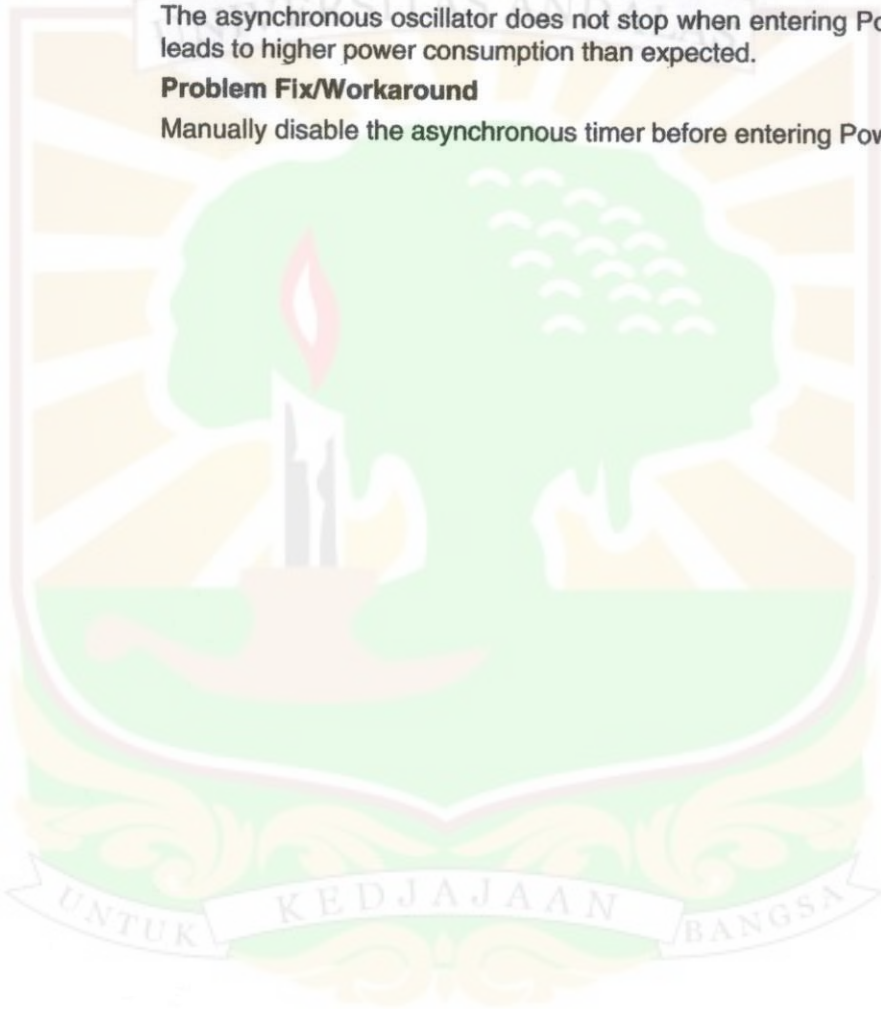
When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

2. Asynchronous Oscillator does not stop in Power-down

The asynchronous oscillator does not stop when entering Power-down mode. This leads to higher power consumption than expected.

Problem Fix/Workaround

Manually disable the asynchronous timer before entering Power-down.



datasheet Revision History

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

Changes from Rev.
02J- 08/06 to Rev.
02K- 10/06

1. Updated TOP/BOTTOM description for all Timer/Counters Fast PWM mode.
2. Updated "Errata" on page 18.

Changes from Rev.
02I- 06/06 to Rev.
02J- 08/06

1. Updated "Ordering Information" on page 13.

Changes from Rev.
02H- 04/06 to Rev.
02I- 06/06

1. Updated code example "USART Initialization" on page 150.

Changes from Rev.
02G- 04/05 to Rev.
02H- 04/06

1. Added "Resources" on page 6.
2. Updated Table 7 on page 29, Table 17 on page 42 and Table 111 on page 258.
3. Updated "Serial Peripheral Interface – SPI" on page 136.
4. Updated note in "Bit Rate Generator Unit" on page 180.

Changes from Rev.
02F- 06/04 to Rev.
02G- 04/05

1. Removed "Preliminary" and TBD's.
2. Updated Table 37 on page 69 and Table 113 on page 261.
3. Updated "Electrical Characteristics" on page 255.
4. Updated "Ordering Information" on page 13.

Changes from Rev.
02E-12/03 to Rev.
02G-06/04

1. MLF-package alternative changed to "Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF".

Changes from Rev.
02E-12/03 to Rev.
02F-06/04

1. Updated "Reset Characteristics" on page 37.
2. Updated SPH in "Stack Pointer" on page 12.
3. Updated C code in "USART Initialization" on page 150.
4. Updated "Errata" on page 18.

Changes from Rev.
02D-09/03 to Rev.
02E-12/03

1. Updated "Calibrated Internal RC Oscillator" on page 29.
2. Added section "Errata" on page 18.



anges from Rev.
2C-04/03 to Rev.
2D-09/03

1. Removed "Advance Information" and some TBD's from the datasheet.
2. Added note to "Pinout ATmega8535" on page 2.
3. Updated "Reset Characteristics" on page 37.
4. Updated "Absolute Maximum Ratings" and "DC Characteristics" in "Electrical Characteristics" on page 255.
5. Updated Table 111 on page 258.
6. Updated "ADC Characteristics" on page 263.
7. Updated "ATmega8535 Typical Characteristics" on page 266.
8. Removed CALL and JMP instructions from code examples and "Instruction Set Summary" on page 10.

anges from Rev.
2B-09/02 to Rev.
2C-04/03

1. Updated "Packaging Information" on page 14.
2. Updated Figure 1 on page 2, Figure 84 on page 179, Figure 85 on page 185, Figure 87 on page 191, Figure 98 on page 207.
3. Added the section "EEPROM Write During Power-down Sleep Mode" on page 22.
4. Removed the references to the application notes "Multi-purpose Oscillator" and "32 kHz Crystal Oscillator", which do not exist.
5. Updated code examples on page 44.
6. Removed ADHSM bit.
7. Renamed Port D pin ICP to ICP1. See "Alternate Functions of Port D" on page 64.
8. Added information about PWM symmetry for Timer 0 on page 79 and Timer 2 on page 126.
9. Updated Table 68 on page 169, Table 75 on page 190, Table 76 on page 193, Table 77 on page 196, Table 108 on page 253, Table 113 on page 261.
10. Updated description on "Bit 5 – TWSTA: TWI START Condition Bit" on page 182.
11. Updated the description in "Filling the Temporary Buffer (Page Loading)" and "Performing a Page Write" on page 231.
12. Removed the section description in "SPI Serial Programming Characteristics" on page 254.
13. Updated "Electrical Characteristics" on page 255.

14. Updated "ADC Characteristics" on page 263.
14. Updated "Register Summary" on page 8.
15. Various Timer 1 corrections.
16. Added WD_FUSE period in Table 108 on page 253.
1. Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.

anges from Rev.
02A-06/02 to Rev.
02B-09/02





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
Shinachem Golden Plaza
7 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
F, Tonetsu Shinkawa Bldg.
24-8 Shinkawa
Huo-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 Chantrerie
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 actual 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 LIABILITY 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 or product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Atmel's products are not authorized, or warranted for use as components in applications intended to support or sustain life.

© 2006 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, Everywhere You Are®, AVR®, and others are the trademarks or registered trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

PING)))™ Ultrasonic Distance Sensor (#28015)

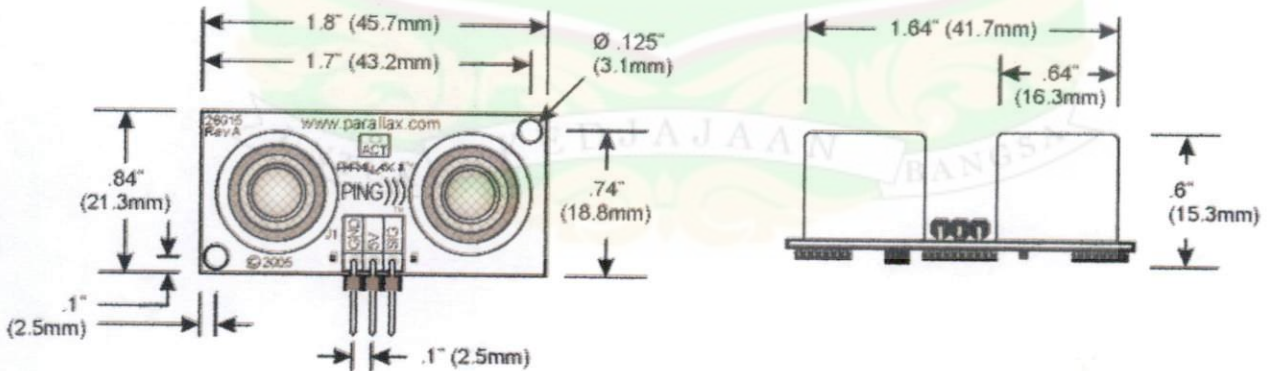
The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to BASIC Stamp® or Javelin Stamp microcontrollers, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width the distance to target can easily be calculated.

Features

- Supply Voltage – 5 VDC
- Supply Current – 30 mA typ; 35 mA max
- Range – 2 cm to 3 m (0.8 in to 3.3 yds)
- Input Trigger – positive TTL pulse, 2 μ s min, 5 μ s typ.
- Echo Pulse – positive TTL pulse, 115 μ s to 18.5 ms
- Echo Hold-off – 750 μ s from fall of Trigger pulse
- Burst Frequency – 40 kHz for 200 μ s
- *Burst Indicator LED shows sensor activity*
- Delay before next measurement – 200 μ s
- Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)

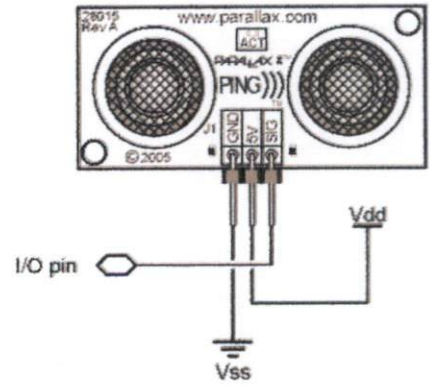
Dimensions



Pin Definitions

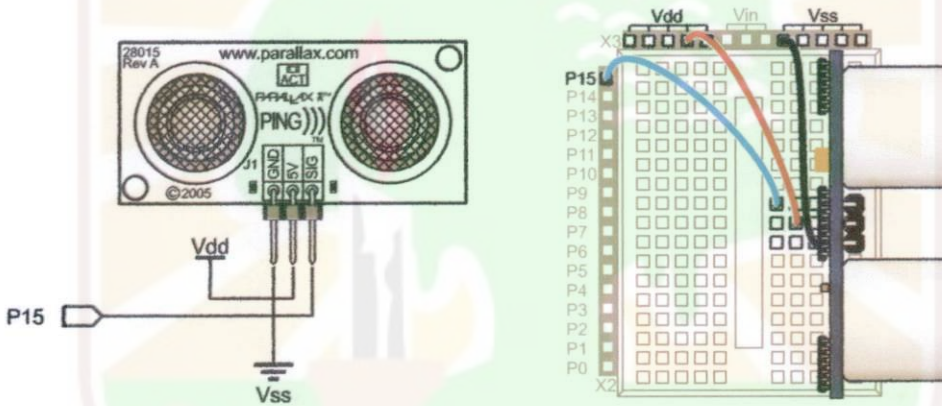
GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply power (5 VDC), ground, and signal. The header allows the sensor to be plugged into a solderless breadboard, or to be located remotely through the use of a standard servo extender cable (Parallax part #805-00002). Standard connections are shown in the diagram to the right.



Quick-Start Circuit

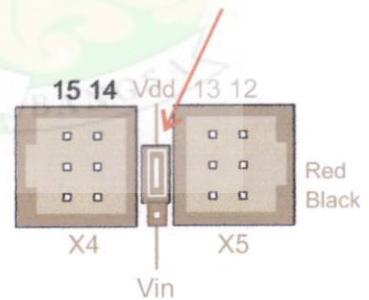
This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp[®] 2 via the Board of Education[®] breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example program Ping_Demo.BS2 listed on page 7.



Servo Cable and Port Cautions

If you want to connect your PING))) sensor to a Board of Education using a servo extension cable, follow these steps:

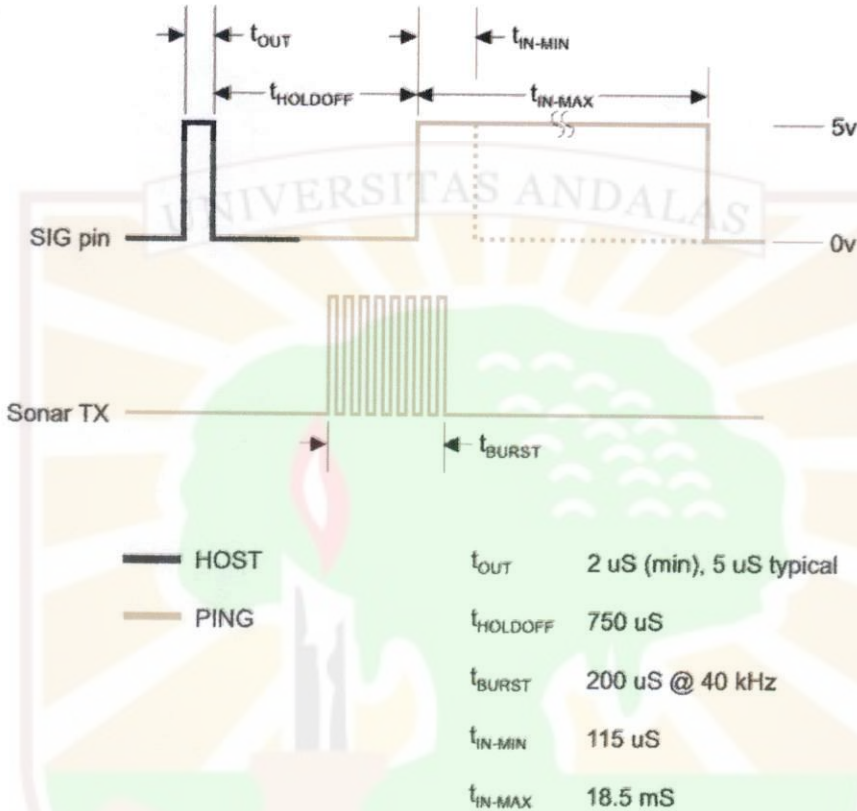
1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown.
4. If your Board of Education servo ports do not have a jumper, do not use them with the PING))) sensor. These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the servo cable directly to the breadboard with a 3-pin header. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.



Board of Education Servo Port Jumper, Set to Vdd

Theory of Operation

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air at about 1130 feet per second, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.



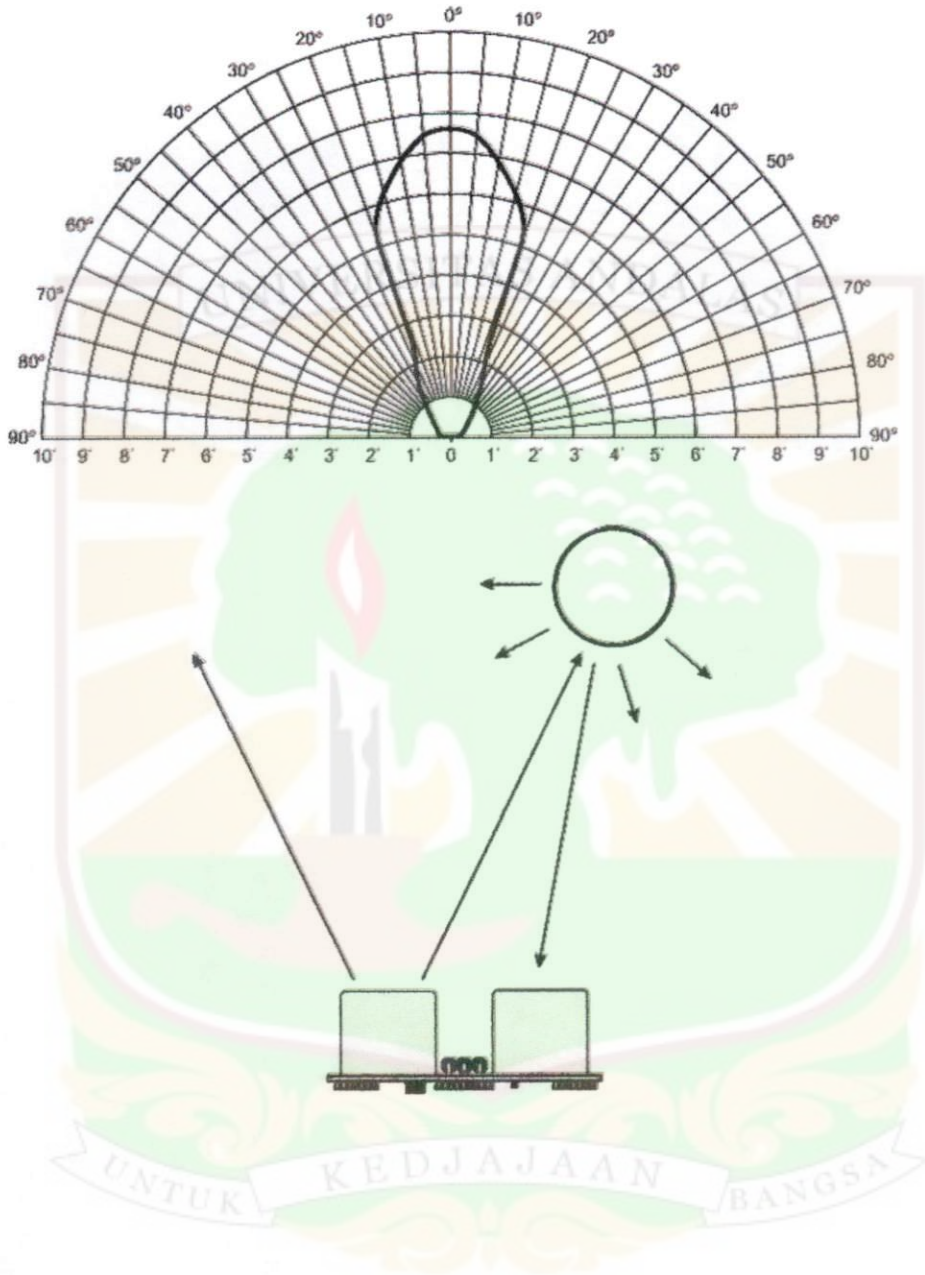
Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

Test 1

Sensor Elevation: 40 in. (101.6 cm)

Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation

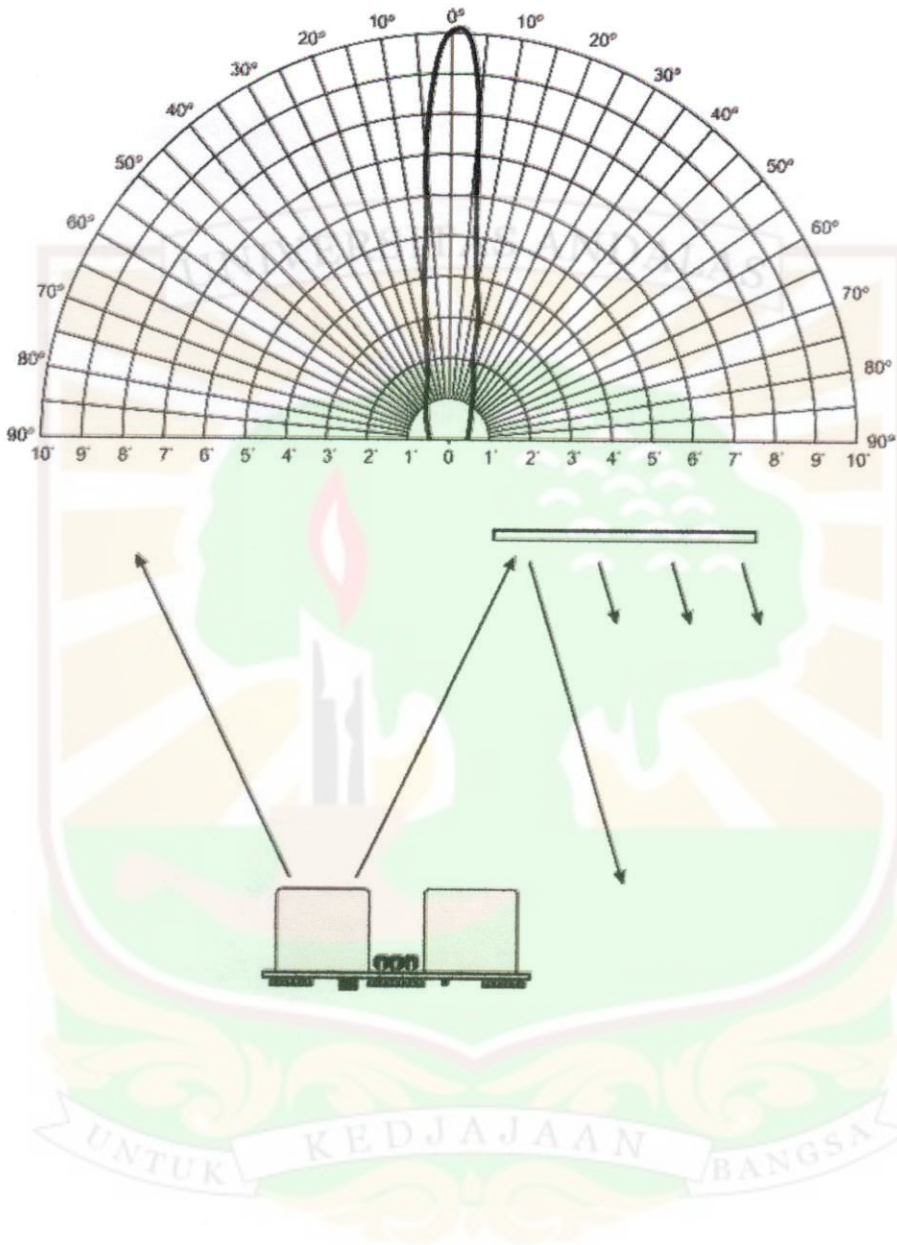


Test 2

Sensor Elevation: 40 in. (101.6 cm)

Target: 12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole

- target positioned parallel to backplane of sensor



Program Example: BASIC Stamp 2 Microcontroller

The following program demonstrates the use of the PING))) sensor with the BASIC Stamp 2 microcontroller. Any model of BASIC Stamp 2 module will work with this program as conditional compilation techniques are used to make adjustments based on the module that is connected.

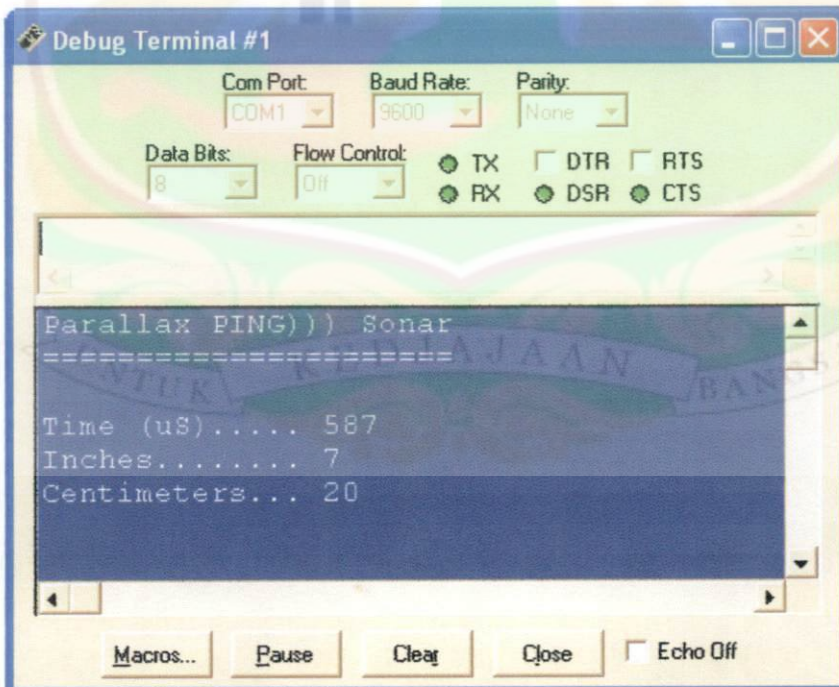
The heart of the program is the **Get_Sonar** subroutine. This routine starts by making the output bit of the selected IO pin zero – this will cause the successive **PULSOUT** to be low-high-low as required for triggering the PING))) sensor. After the trigger pulse falls the sensor will wait about 200 microseconds before transmitting the ultrasonic burst. This allows the BS2 to load and prepare the next instruction. That instruction, **PULSIN**, is used to measure the high-going pulse that corresponds to the distance to the target object.

The raw return value from **PULSIN** must be scaled due to resolution differences between the various members of the BS2 family. After the raw value is converted to microseconds, it is divided by two in order to remove the "return trip" of the echo pulse. The value now held in *rawDist* is the distance to the target in microseconds.

Conversion from microseconds to inches (or centimeters) is now a simple matter of math. The generally-accepted value for the speed-of-sound is 1130 feet per second. This works out to 13,560 inches per second or one inch in 73.746 microseconds. The question becomes, how do we divide our pulse measurement value by the floating-point number 73.746?

Another way to divide by 73.746 is to multiply by 0.01356. For new BASIC Stamp users this may seem a dilemma but in fact there is a special operator, ******, that allows us to do just that. The ****** operator has the affect of multiplying a value by units of 1/65,536. To find the parameter for ****** then, we simply multiply 0.01356 by 65,536; the result is 888.668 (we'll round up to 889).

Conversion to centimeters uses the same process and the result of the program is shown below:



```

=====
File..... Ping_Demo.BS2
Purpose.... Demo Code for Parallax PING))) Sonar Sensor
Author..... Parallax, Inc.
E-mail..... support@parallax.com
Started....
Updated.... 08 JUN 2005

{$STAMP BS2}
{$PBASIC 2.5}
=====

```

```

-----[ Program Description ]-----

```

```

This program demonstrates the use of the Parallax PING))) sensor and then
converting the raw measurement to English (inches) and Metric (cm) units.

```

```

Sonar Math:

```

```

At sea level sound travels through air at 1130 feet per second. This
equates to 1 inch in 73.746 uS, or 1 cm in 29.034 uS).

```

```

Since the PING))) sensor measures the time required for the sound wave to
travel from the sensor and back. The result -- after conversion to
microseconds for the BASIC Stamp module in use -- is divided by two to
remove the return portion of the echo pulse. The final raw result is
the duration from the front of the sensor to the target in microseconds.

```

```

-----[ I/O Definitions ]-----

```

```

Ping          PIN      15

```

```

-----[ Constants ]-----

```

```

#SELECT $STAMP
#CASE BS2, BS2E
  Trigger      CON      5           ' trigger pulse = 10 uS
  Scale        CON     $200        ' raw x 2.00 = uS
#CASE BS2SX, BS2P, BS2PX
  Trigger      CON     13
  Scale        CON     $0CD        ' raw x 0.80 = uS
#CASE BS2PE
  Trigger      CON      5
  Scale        CON     $1E1        ' raw x 1.88 = uS
#ENDSELECT

RawToIn        CON     889         ' 1 / 73.746 (with **)
RawToCm        CON    2257         ' 1 / 29.034 (with **)

IsHigh         CON      1         ' for PULSOUT
IsLow          CON      0

```

' -----[Variables]-----

```
rawDist      VAR      Word      ' raw measurement
inches       VAR      Word
cm           VAR      Word
```

' -----[Initialization]-----

```
Reset:
  DEBUG CLS,
    "Parallax PING))) Sonar", CR,      ' setup report screen
    "-----", CR,
    CR,
    "Time (uS).....", CR,
    "Inches.....", CR,
    "Centimeters..."
```

' -----[Program Code]-----

```
Main:
  DO
    GOSUB Get_Sonar      ' get sensor value
    inches = rawDist ** RawToIn  ' convert to inches
    cm = rawDist ** RawToCm      ' convert to centimeters

    DEBUG CRSRXY, 15, 3,      ' update report screen
      DEC rawDist, CLREOL,
      CRSRXY, 15, 4,
      DEC inches, CLREOL,
      CRSRXY, 15, 5,
      DEC cm, CLREOL

  PAUSE 100
  LOOP
  END
```

' -----[Subroutines]-----

' This subroutine triggers the PING))) sonar sensor and measures the echo pulse. The raw value from the sensor is converted to microseconds based on the Stamp module in use. This value is divided by two to remove the return trip -- the result value is the distance from the sensor to the target in microseconds.

```
Get_Sonar:
  Ping = IsLow      ' make trigger 0-1-0
  PULSOUT Ping, Trigger  ' activate sensor
  PULSIN Ping, IsHigh, rawDist  ' measure echo pulse
  rawDist = rawDist */ Scale  ' convert to uS
  rawDist = rawDist / 2      ' remove return trip
  RETURN
```

Program Example: BASIC Stamp 1 Microcontroller

```
' =====
'
' File..... Ping_Demo.BS1
' Purpose.... Demo Code for Parallax PING))) Sonar Sensor
' Author..... Parallax, Inc.
' E-mail..... support@parallax.com
' Started....
' Updated.... 06 JUN 2006
'
' {$STAMP BS1}
' {$PBASIC 1.0}
' =====
'
' ----- [ Program Description ] -----
'
' This program demonstrates the use of the Parallax PING))) sensor and then
' converting the raw measurement to English (inches) and Metric (cm) units.
'
' Sonar Math:
'
' At sea level sound travels through air at 1130 feet per second. This
' equates to 1 inch in 73.746 uS, or 1 cm in 29.034 uS).
'
' Since the PING))) sensor measures the time required for the sound wave to
' travel from the sensor and back. The result -- after conversion to
' microseconds for the BASIC Stamp module in use -- is divided by two to
' remove the return portion of the echo pulse. The final raw result is
' the duration from the front of the sensor to the target in microseconds.
'
' ----- [ I/O Definitions ] -----
'
SYMBOL Ping          = 7
'
' ----- [ Constants ] -----
'
SYMBOL Trigger       = 1           ' 10 uS trigger pulse
SYMBOL Scale         = 10          ' raw x 10.00 = uS
'
SYMBOL RawToIn       = 889         ' 1 / 73.746 (with **)
SYMBOL RawToCm       = 2257        ' 1 / 29.034 (with **)
'
SYMBOL IsHigh        = 1           ' for PULSOUT
SYMBOL IsLow         = 0
'
' ----- [ Variables ] -----
'
SYMBOL rawDist       = W1           ' raw measurement
SYMBOL inches        = W2
SYMBOL cm            = W3
```

' -----[Program Code]-----

```
Main:
GOSUB Get_Sonar          ' get sensor value
inches = rawDist ** RawToIn  ' convert to inches
cm = rawDist ** RawToCm    ' convert to centimeters

DEBUG CLS              ' report
DEBUG "Time (uS)..... ", #rawDist, CR
DEBUG "Inches..... ", #inches, CR
DEBUG "Centimeters... ", #cm

PAUSE 500
GOTO Main

END
```

' -----[Subroutines]-----

```
' This subroutine triggers the PING))) sonar sensor and measures
' the echo pulse. The raw value from the sensor is converted to
' microseconds based on the Stamp module in use. This value is
' divided by two to remove the return trip -- the result value is
' the distance from the sensor to the target in microseconds.

Get_Sonar:
LOW Ping                ' make trigger 0-1-0
PULSOUT Ping, Trigger  ' activate sensor
PULSIN Ping, IsHigh, rawDist  ' measure echo pulse
rawDist = rawDist * Scale  ' convert to uS
rawDist = rawDist / 2     ' remove return trip
RETURN
```



```

/*
 * The PING))) returns a pulse width of 73.746 uS per inch. Since the
 * Javelin pulseIn() round-trip echo time is in 8.68 uS units, this is the
 * same as a one-way trip in 4.34 uS units. Dividing 73.746 by 4.34 we
 * get a time-per-inch conversion factor of 16.9922 (x 0.058851).
 *
 * Values to derive conversion factors are selected to prevent roll-over
 * past the 15-bit positive values of Javelin Stamp integers.
 */

/**
 * @return PING))) distance value in inches
 */
public int getIn() {
    return (getRaw() * 3 / 51); // raw * 0.058824
}

/**
 * @return PING))) distance value in tenths of inches
 */
public int getIn10() {
    return (getRaw() * 3 / 5); // raw / 1.6667
}

/*
 * The PING))) returns a pulse width of 29.033 uS per centimeter. As the
 * Javelin pulseIn() round-trip echo time is in 8.68 uS units, this is the
 * same as a one-way trip in 4.34 uS units. Dividing 29.033 by 4.34 we
 * get a time-per-centimeter conversion factor of 6.6896.
 *
 * Values to derive conversion factors are selected to prevent roll-over
 * past the 15-bit positive values of Javelin Stamp integers.
 */

/**
 * @return PING))) distance value in centimeters
 */
public int getCm() {
    return (getRaw() * 3 / 20); // raw / 6.6667
}

/**
 * @return PING))) distance value in millimeters
 */
public int getMm() {
    return (getRaw() * 3 / 2); // raw / 0.6667
}
}

```

This simple demo illustrates the use of the PING))) ultrasonic range finder class with the Javelin Stamp:

```
import stamp.core.*;
import stamp.peripheral.sensor.Ping;

public class testPing {

    public static final char HOME = 0x01;

    public static void main() {

        Ping range = new Ping(CPU.pin0);
        StringBuffer msg = new StringBuffer();

        int distance;

        while (true) {
            // measure distance to target in inches
            distance = range.getIn();

            // create and display measurement message
            msg.clear();
            msg.append(HOME);
            msg.append(distance);
            msg.append(" \" \" \" \n");
            System.out.print(msg.toString());

            // wait 0.5 seconds between readings
            CPU.delay(5000);
        }
    }
}
```



Program Example: Javelin Stamp Microcontroller

This class file implements several methods for using the PING))) sensor:

```
package stamp.peripheral.sensor;

import stamp.core.*;

/**
 * This class provides an interface to the Parallax PING))) ultrasonic
 * range finder module.
 * <p>
 * <i>Usage:</i><br>
 * <code>
 *   Ping range = new Ping(CPU.pin0);           // trigger and echo on P0
 * </code>
 * <p>
 * Detailed documentation for the PING))) Sensor can be found at: <br>
 * http://www.parallax.com/detail.asp?product\_id=28015
 * <p>
 *
 * @version 1.0 03 FEB 2005
 */
public final class Ping {

    private int ioPin;

    /**
     * Creates PING))) range finder object
     *
     * @param ioPin PING))) trigger and echo return pin
     */
    public Ping (int ioPin) {
        this.ioPin = ioPin;
    }

    /**
     * Returns raw distance value from the PING))) sensor.
     *
     * @return Raw distance value from PING)))
     */
    public int getRaw() {

        int echoRaw = 0;

        CPU.writePin(ioPin, false);           // setup for high-going pulse
        CPU.pulseOut(1, ioPin);               // send trigger pulse
        echoRaw = CPU.pulseIn(2171, ioPin, true); // measure echo return

        // return echo pulse if in range; zero if out-of-range
        return (echoRaw < 2131) ? echoRaw : 0;
    }
}
```