

**RANCANG BANGUN SISTEM INSTRUMENTASI
UNTUK *MONITORING* KETINGGIAN AIR
BERBASIS SENSOR ULTRASONIK *SRF05***

SKRIPSI

Oleh:
ANGGA SETIYO PRAYOGO
0310930007



**JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG**

2009

**RANCANG BANGUN SISTEM INSTRUMENTASI
UNTUK *MONITORING* KETINGGIAN AIR
BERBASIS SENSOR ULTRASONIK *SRF05***

SKRIPSI

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Sains Dalam Bidang Fisika

Oleh:
ANGGA SETIYO PRAYOGO
0310930007



**JURUSAN FISIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG**

2009

LEMBAR PENGESAHAN SKRIPSI

**RANCANG BANGUN SISTEM INSTRUMENTASI
UNTUK *MONITORING* KETINGGIAN AIR
BERBASIS SENSOR ULTRASONIK *SRF05***

Oleh:
ANGGA SETIYO PRAYOGO
0310930007-93

Setelah dipertahankan di depan Majelis Penguji
pada tanggal
Dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Sains dalam bidang fisika

Pembimbing I

Pembimbing II

Dr. Eng., Didik R. Santoso, M.Si.
NIP. 132 086 158

Drs. Adi Susilo, M.Si., Ph.D.
NIP. 131 960 447

Mengetahui,
Ketua Jurusan Fisika
Fakultas MIPA Universitas Brawijaya

Drs. Adi Susilo, M.Si., Ph.D.
NIP. 131 960 447

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Angga Setiyo Prayogo

NIM : 0310930007

Bidang Minat : Instrumentasi

Jurusan : Fisika

Penulis skripsi berjudul :

**RANCANG BANGUN SISTEM INSTRUMENTASI UNTUK
MONITORING KETINGGIAN AIR BERBASIS SENSOR
ULTRASONIK SRF05**

Dengan ini menyatakan bahwa :

1. Skripsi ini adalah benar – benar karya saya sendiri, dan bukan hasil plagiat dari karya orang lain. Karya – karya yang tercantum dalam Daftar Pustaka skripsi ini, semata – mata digunakan sebagai acuan/referensi.
2. Apabila dikemudian hari diketahui bahwa isi skripsi saya merupakan hasil plagiat, maka saya bersedia menanggung akibat hukum dari keadaan tersebut.

Demikian pernyataan ini dibuat dengan segala kesadaran

Malang,.....

Yang menyatakan

(Angga Setiyo Prayogo)

NIM. 0310930007 – 93

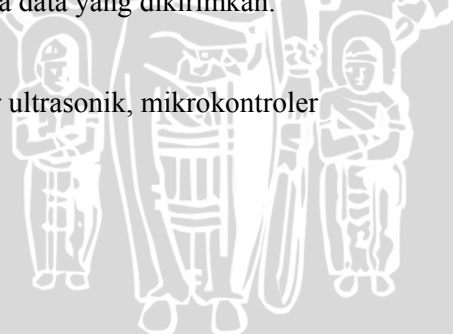
RANCANG BANGUN SISTEM INSTRUMENTASI UNTUK *MONITORING* KETINGGIAN AIR BERBASIS SENSOR ULTRASONIK *SRF05*

ABSTRAK

Penelitian tentang pembuatan alat pengukur ketinggian air dengan biaya murah dan efisien telah berhasil dilakukan. Sensor yang digunakan adalah sensor ultrasonik jenis SRF05 dengan sistem transfer data multidrop RS-485. Setelah pengujian, sensor dapat digunakan untuk mendeteksi jarak maksimum 380 cm, baik itu menjauhi maupun mendekati obyek (air). Pengaturan kerja sensor dilakukan oleh PC Master yang dilengkapi RS-232 To RS-485 *Converter*. Sedangkan *slave* berfungsi untuk mengirimkan data pengukuran yang terdiri dari sensor yang terhubung langsung ke mikrokontroler PIC16F876 yang dilengkapi dengan NIC RS-485.

Sensor diletakkan di ujung pipa setinggi 250 cm, sementara air naik dari dasar pipa sampai mendekati ujung pipa, kemudian air turun lagi, sehingga oleh sensor terdeteksi perubahan naik turunnya, hal ini bisa diaplikasikan sebagai pendeteksi adanya air pasang/surut. Nilai pulsa delay gelombang ultrasonik dan ketinggian air ditampilkan pada *PC* dengan menggunakan karakter kunci sebagai alamat dan pembeda data yang dikirimkan.

Kata kunci : Sensor ultrasonik, mikrokontroler



DEVELOPMENT OF INSTRUMENTATION SYSTEM FOR WATER LEVEL MONITORING BASED ON ULTRASONIC SENSOR SRF05

ABSTRACT

An experiment about development of water level measurement has been made successfully with low cost and efficient. By using ultrasonic sensor SRF05 and data transfer based on RS-485 multidrop network. Sensor can be used to detect 380 cm maximum length with object (water). PC is equipped with a RS-232 To RS-485 Converter as the master that is controlled by the sensor. The slave that the function is transfers the measurement data, consists of a sensor SRF05 that connected to a microcontroller PIC16F876 with an IC NIC RS-485.

The performance of system was tested with placed the sensor on 250 cm length water pipe, then water is given up and down from bottom of pipe, so the moving of water surface is detected by sensor. This phenomenon can be applied as a detector of water up and down. The results show that ultrasonic pulse delay and the water level can be displayed on a PC with a key characters as the address from slave.

Key Words : ultrasonic sensor, microcontroller

KATA PENGANTAR

Segala puji dan syukur ke hadirat Allah SWT atas segala rahmat, hidayah dan karuniaNya sehingga penulis dapat menyelesaikan Skripsi dengan judul **“RANCANG BANGUN SISTEM INSTRUMENTASI UNTUK MONITORING KETINGGIAN AIR BERBASIS SENSOR ULTRASONIK SRF05”** yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Sains dalam bidang Fisika.

Skripsi ini dapat diselesaikan karena dukungan dari beberapa pihak. Oleh karena itu, dengan segenap hati penulis ingin menyampaikan ucapan dan rasa terima kasih yang sebesar-besarnya, khususnya kepada :

1. Bapak Dr. Eng, Didik R. Santoso, M.Si. selaku pembimbing I atas segala kesabaran, arahan, dan atas pendanaan untuk penulis dalam menyelesaikan skripsi ini.
2. Bapak Adi Susilo, Ph.D. selaku Ketua Jurusan Fisika Universitas Brawijaya dan selaku pembimbing II atas segala bimbingan dan atas pendanaan untuk mengerjakan skripsi ini.
3. Orang tuaku Bapak Suwarno dan Ibu Mujiati, serta Kakakku Okta Wijayanti dan keluarga besarku yang selalu mendoakan, mendukung, dan membantu aktivitas penulis selama kuliah, “ini persembahanku buat cinta kalian..”
4. Seluruh dosen penguji dan Bapak/Ibu dosen Jurusan Fisika Universitas Brawijaya atas ilmu yang telah diberikan selama kuliah.
5. Seluruh karyawan dan staf Jurusan Fisika Universitas Brawijaya atas bantuannya dalam melancarkan administrasi perkuliahan.
6. Teman – teman dan keluarga besar Jurusan Fisika Universitas Brawijaya khususnya angkatan 2003, persahabatan kita adalah awal perjuangan untuk terus lebih baik.
7. Semua pihak yang telah memberikan dukungan kepada penulis sehingga penulis bisa menyelesaikan skripsi ini.

Penulis menyadari laporan skripsi ini masih jauh dari sempurna, karena itu penulis mengharapkan kritik dan saran yang bersifat membangun.

Akhirnya penulis mengharapkan semoga skripsi ini bisa memberikan manfaat yang baik kepada pembaca.

Malang, Juli 2009

Penulis

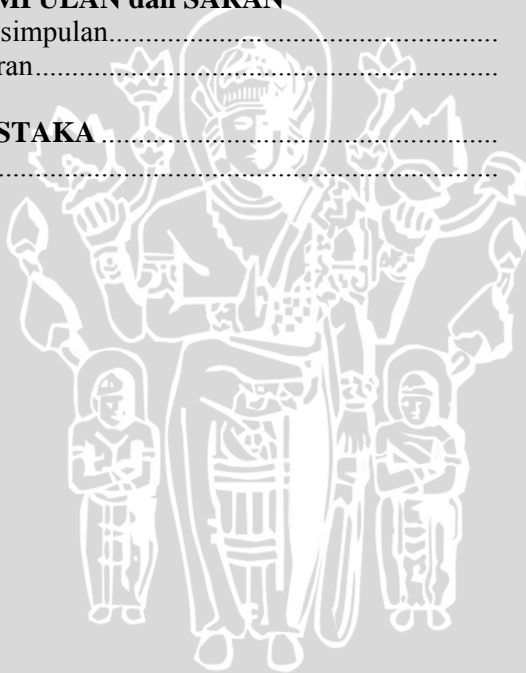
UNIVERSITAS BRAWIJAYA



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PENGESAHAN SKRIPSI	ii
LEMBAR PERNYATAAN	iii
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR LAMPIRAN	xi
DAFTAR GAMBAR	xii
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	2
BAB II TINJAUAN PUSTAKA	
2.1 Gelombang Ultrasonik.....	3
2.1.2 Sensor ultrasonik.....	3
2.1.2 <i>Sonic Range Finder</i> SRF05.....	4
2.2 <i>Device Network Interface Card(NIC)</i> RS-485	5
2.3 Mikrokontroler PIC16F876.....	6
2.4 Port Serial Komputer.....	8
BAB III METODE PENELITIAN	
3.1 Waktu dan Tempat Penelitian	11
3.2 Tahap Pengerjaan	11
3.3 Perancangan Perangkat Keras	11
3.3.1 Perancangan <i>master</i>	12
3.3.2 Perancangan <i>slave</i>	13
3.3.3 Perancangan komunikasi RS-485.....	14
3.3.4 Perancangan tabung pengujian.....	16
3.4 Perancangan Perangkat Lunak.....	18
3.4.1 Perancangan program keseluruhan.....	18

3.4.2 Inisialisasi dan <i>trigger</i>	19
3.4.3 Pengaktifan sensor ultrasonik.....	21
3.5 Pengujian Akuisisi Data.....	23
3.5.1 Pengujian <i>master</i>	23
3.5.2 Pengujian <i>slave</i>	24
3.5.3 Pengujian sensor ultrasonik.....	25
3.5.4 Pengujian aplikasi alat keseluruhan.....	26
BAB IV HASIL dan PEMBAHASAN	
4.1 Hasil Pengujian Sensor Ultrasonik.....	27
4.2 Hasil Pengujian <i>Software</i>	29
BAB V KESIMPULAN dan SARAN	
5.1 Kesimpulan.....	35
5.2 Saran.....	35
DAFTAR PUSTAKA	37
LAMPIRAN	39



DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Data Pengujian Sensor Ultrasonik	39
Lampiran 2 Data Pengujian <i>Software</i>	42
Lampiran 3 Gambar Alat	45
Lampiran 4 <i>Source Code Master (Delphi7)</i>	47
Lampiran 5 <i>Source Code Mikrokontroler (Assembly)</i>	57
Lampiran 6 <i>Data Sheet Sonic Range Finder SRF05</i>	64



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Ilustrasi kerja sensor	3
Gambar 2.2 Sensor ultrasonik <i>SRF05</i>	4
Gambar 2.3 Blok diagram <i>NIC RS-485</i>	6
Gambar 2.4 Susunan pin mikrokontroler <i>PIC16F876</i>	7
Gambar 2.5 Pengiriman data serial secara asinkron	8
Gambar 2.6 Konfigurasi pin <i>DB9</i>	9
Gambar 2.7 <i>DB9 Male</i>	9
Gambar 2.8 <i>DB9 Female</i>	9
Gambar 2.9 Pemasangan <i>DB9 Male</i> dan <i>DB9 Female</i>	10
Gambar 3.1 Perancangan perangkat keras meliputi sensor <i>SRF05, slave, RS-485 to RS-232 converter</i> , dan <i>PC</i>	12
Gambar 3.2 Blok diagram <i>RS-232 to RS-485 converter</i> yang diatur dari <i>software</i> dengan menggunakan jalur <i>RTS</i>	13
Gambar 3.3 Rangkaian <i>slave</i>	14
Gambar 3.4 Jaringan komunikasi multidrop	15
Gambar 3.5 Skema tabung pengujian	16
Gambar 3.6 Implementasi tabung pengujian yang dibuat	17
Gambar 3.7 Diagram transfer data keseluruhan	19
Gambar 3.8 Prosedur inisialisasi mikrokontroler	20
Gambar 3.9 Prosedur subrutin <i>trigger</i>	21
Gambar 3.10 Diagram <i>clock SRF05</i>	22
Gambar 3.11 Diagram pembacaan pulsa <i>delay</i> dan jarak oleh sensor ultrasonik	23
Gambar 3.12 Pengujian sensor ultrasonik	25
Gambar 4.1 Grafik pengujian sensor ultrasonik sebagai kalibrasi, menggunakan <i>rollmeter</i> sebagai pembanding ...	27
Gambar 4.2 Grafik hasil pengujian sensor terhadap <i>rollmeter</i>	28
Gambar 4.3 Tampilan software	29
Gambar 4.4 Perubahan <i>time delay/output</i> sensor	31
Gambar 4.5 Perubahan jarak sensor akibat naik turun permukaan air	31
Gambar 4.6 Proses <i>monitoring</i> ketinggian air	32

BAB I

PENDAHULUAN

1.1 Latar Belakang

Melakukan pengukuran dan *monitoring* parameter fisis yang ada di alam merupakan pekerjaan yang sering dilakukan oleh manusia. Hal ini bertujuan untuk mengetahui kondisi fisis suatu obyek baik dimensi (seperti jarak) maupun sifat fisis internal (seperti suhu) yang berhubungan obyek tersebut, serta untuk mengamati dan mengawasi segala bentuk perubahan kondisi fisis suatu obyek.

Salah satu aplikasi pekerjaan itu adalah pengukuran ketinggian air, yang ada kalanya perlu dilakukan pada banyak tempat sebagai perbandingan, dengan letak obyek saling berjauhan dan kadang membahayakan. Hal ini memerlukan pengamatan dari satu tempat saja karena penempatan sarana pengukuran ketinggian air yang letaknya jauh dan tersebar akan merepotkan jika tidak dapat diamati dari satu tempat. Selama ini, alat pengukur ketinggian air yang ada masih bersifat manual dan membutuhkan pengamatan langsung di lapangan. Sedangkan sistem canggih yang ada umumnya relatif mahal karena membeli dari luar negeri dan membutuhkan orang yang kompeten untuk menjalankannya.

Proses perekaman dan penyimpanan data ketinggian air secara manual langsung di lapangan memiliki kelemahan. Media pencatatan manual memiliki kekurangan dalam kapasitas dan masa pakai media penyimpanan tersebut. Perbedaan bentuk tulisan dari pencatatan secara manual juga menjadi masalah tersendiri pada saat dilakukan pengamatan ulang hasil rekaman pengukuran yang telah dicatat.

Melihat fenomena tersebut, pada penelitian ini kami memberikan suatu ide dengan membuat perangkat instrumentasi yang dapat memonitor perubahan ketinggian air yang bisa diaplikasikan sebagai sistem peringatan adanya air pasang, melalui sensor ultrasonik yang dikendalikan kerjanya oleh mikrokontroler serta dapat diatur penggunaannya melalui komputer. Dengan sistem terdistribusi, penempatan beberapa *slave* dan sensor ultrasonik pada beberapa tempat dapat memberikan informasi ketinggian air di beberapa titik. Hasil pengamatan dapat diamati dan disimpan pada *PC* untuk efisiensi pengolahan dan penggunaan lebih lanjut.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan, maka rumusan masalah yang dibahas dalam penelitian ini adalah:

1. Bagaimana merancang komunikasi dan transmisi data dari *master* menuju *slave*, atau sebaliknya untuk pengukuran jarak dengan sensor ultrasonik.
2. Bagaimana mengaktifkan dan menguji sensor ultrasonik secara *online* dengan mikrokontroler untuk mengukur ketinggian permukaan air serta menampilkan hasilnya pada komputer.

1.3 Batasan Masalah

Untuk menghindari penyimpangan dan kesalahpahaman penelitian, maka diberi batasan masalah, antara lain :

1. Perancangan hanya sebatas *prototipe* pengukur jarak.
2. Menggunakan bahasa mesin pada mikrokontroler *PIC16F876* dengan sensor ultrasonik *SRF05*.
3. Obyek ada di depan sensor dengan posisi tegak lurus permukaan.
4. Hanya dibahas sampai tampilan data di komputer berupa ketinggian air yang terukur dari sensor.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk membuat piranti pengukur ketinggian air secara online untuk mengetahui kenaikan atau penurunan posisi permukaan air dengan sensor ultrasonik yang efektif dengan biaya yang murah.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah dihasilkan piranti pengukur perubahan ketinggian air dengan sensor ultrasonik, yang bisa diaplikasikan secara efektif untuk mengetahui bagaimana posisi permukaan air.

BAB II TINJAUAN PUSTAKA

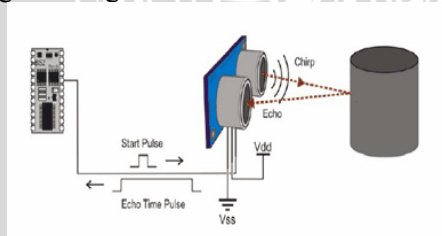
2.1 Gelombang Ultrasonik

Gelombang ultrasonik merupakan gelombang mekanik longitudinal dengan frekuensi lebih besar dari 20 kHz. Gelombang ini dapat merambat dalam medium padat, cair dan gas, karena gelombang ultrasonik merupakan rambatan energi dan momentum mekanik sehingga merambat sebagai interaksi dengan molekul dan sifat enersia medium yang dilalui.

Sifat gelombang ultrasonik melalui medium mengakibatkan getaran partikel dengan medium amplitudo sejajar dengan arah rambat secara longitudinal sehingga partikel medium akan membentuk rapatan dan regangan. Proses kontinyu yang menyebabkan terjadinya rapatan dan regangan di dalam medium disebabkan oleh getaran partikel secara periodik selama gelombang ultrasonik melaluinya.

2.1.1 Sensor ultrasonik

Sensor ultrasonik adalah sensor yang bekerja berdasar prinsip pantulan gelombang suara. Sensor menghasilkan gelombang suara yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar pengukuran. Perbedaan waktu antara gelombang suara yang dipancarkan dan yang diterima kembali adalah berbanding lurus dengan jarak atau tinggi objek yang memantulkannya. Jenis objek yang dapat diindranya adalah padat, cair dan butiran. Tanpa kontak, jarak beberapa meter dapat dengan mudah terdeteksi jika sensor dihubungkan dengan mikrokontroler melalui 2 pin I/O saja.



Gambar 2.1 Ilustrasi kerja sensor (Sahala,2004).

Pulsa yang dihasilkan dari keluaran sensor ultrasonik adalah dalam orde mikrodetik (μ s), yang jika dibagi 58 akan langsung

didapatkan jarak dengan orde centimeter, sedangkan jika dibagi 148 akan menghasilkan jarak dengan orde inchi (Anonymous, 2007).

Sedangkan secara teori, jarak antara sensor dengan objek yang direfleksikan dapat dihitung dengan menggunakan rumus:

$$L = \frac{1}{2} \cdot Tof \cdot v \quad (2.1)$$

dimana :

L = jarak ke objek

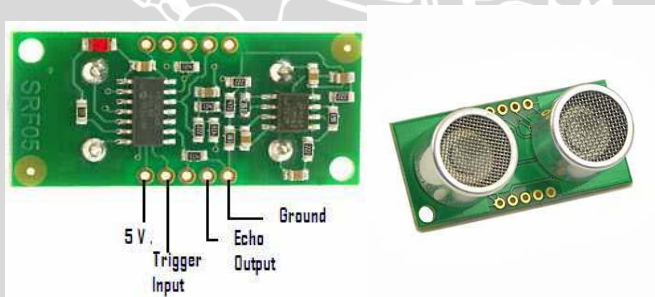
Tof = waktu pengukuran oleh sensor

v = cepat rambat suara (340 m/s)

(Sahala, 2004)

2.1.2 Sonic Range Finder SRF05

Sensor ultrasonik *SRF05* merupakan pengembangan dari *SRF04*, dan dibuat untuk memperbesar fleksibilitas dan jangkauan serta efisiensi karena lebih murah. *SRF05* dapat menjangkau jarak 3 cm - 4 m dengan suplai tegangan 5 volt, arus maksimum 50 mA, frekuensi 40 kHz dan *input trigger* 10 μ s. *SRF05* memiliki 5 pin penghubung, yaitu 5 volt *supply*, *no connection*, *trigger input*, *echo output*, *mode*, serta 0 *ground*. Seperti gambar dibawah ini.



Gambar 2.2 Sensor Ultrasonik *SRF05* (Anonymous, 2007)

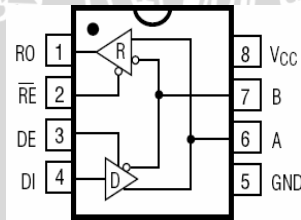
2.2 Device Network Interface Card (NIC) RS-485

RS-485 merupakan pengembangan komunikasi serial untuk menjawab kelemahan komunikasi serial standar *RS-232* yang terdapat pada *PC*. *RS-232* memiliki kelemahan pada jumlah

percabangan meskipun dapat mencapai jarak yang cukup jauh. RS-232 hanya dapat berkomunikasi satu titik ke satu titik yang lain (*point to point*). RS-485 dengan penyempurnaan perancangan dapat dipecah hingga mencapai 32 titik (*multipoint*) dengan jarak relatif lebih jauh dibanding RS-232 yang dapat mencapai jarak 1,2 km (Axelson, 1999).

Pengiriman data pada komunikasi serial RS-485 bersifat *half duplex*. Pengiriman data secara *half duplex* adalah proses pengiriman data yang tidak dapat dilakukan secara dua arah (mengirim dan menerima data) dalam satu waktu tetapi harus bergantian. Pengiriman data secara bergantian mengakibatkan adanya proses menunggu hingga proses pengiriman data yang berlangsung telah selesai.

NIC RS-485 memiliki delapan buah pin yang memiliki fungsi berbeda-beda. Pin nomor satu (*Receiver Output*) merupakan pin yang akan menerima data pada perangkat komunikasi lain. Pin nomor dua (*Receiver Output Enable*) merupakan pin yang mengatur kinerja dari pin RO. Pin RO akan berfungsi apabila pin RE memiliki logika rendah atau bernilai nol. Pin nomor tiga (*Driver Output Enable*) merupakan pin yang berfungsi untuk mengatur kinerja dari pin nomor empat (*Driver Input*) yang akan mengirimkan data menuju perangkat lain pada saat pin nomor tiga berlogika tinggi. Pin nomor lima memiliki fungsi sebagai *ground* dari IC RS-485. Pin nomor enam (A) dan tujuh (B) sebagai jalur data yang dalam bekerjanya bergantung pada pin RE dan DE. Apabila pin RE berlogika rendah maka proses penerimaan data akan berlangsung dan bila pin DE berlogika tinggi maka proses pengiriman data akan berlangsung. Pin A dan B dapat bekerja bila memiliki selisih tegangan antara A dan B sebesar 200mV.



Gambar 2.3 Blok diagram NIC RS-485 (Axelson, 1999).

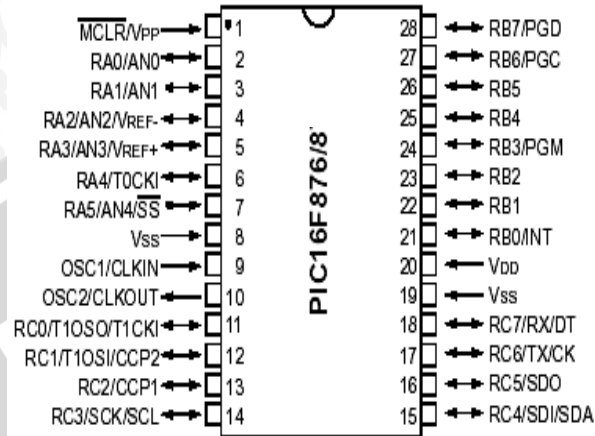
Bila antara jalur A dan B memiliki selisih tegangan kurang dari 200mV maka nilai ini tidak didefinisikan atau RS-485 tidak dapat bekerja sebagaimana mestinya. Pin nomor delapan (Vcc) memiliki fungsi sebagai masukan tegangan sebesar lima volt.

2.3 Mikrokontroler PIC16F876

Mikrokontroler merupakan suatu IC yang memiliki kemampuan proses bertindak berdasarkan program yang telah diberikan atau diisikan pada mikrokontroler tersebut. Mikrokontroler memiliki beberapa bagian untuk mendukung proses bertindak seperti CPU, ROM, RAM dan I/O. CPU (*Central Processing Unit*) berfungsi untuk membaca program yang tersimpan di dalam RAM dan melaksanakannya. ROM (*Read Only Memory*) merupakan memori yang hanya dapat dibaca saja untuk menyimpan program dari mikrokontroler itu sendiri. RAM (*Random Acces Memory*) merupakan memori yang dapat dibaca dan ditulis atau diisi berulang kali oleh pemrogram. I/O (*Input/Output*) merupakan bagian penghubung antara mikrokontroler dengan perangkat luar agar dapat berkomunikasi antara keduanya (Malik, 2003).

Mikrokontroler memiliki fitur tambahan untuk mendukung proses bertindak dari mikrokontroler. Salah satu kompoen tambahan tersebut adalah ADC (*Analog To Digital Converter*). ADC berfungsi untuk mengubah nilai analog yang masuk pada mikrokontroler menjadi nilai digital. Dengan pengubahan nilai analog menjadi nilai digital maka segala bentuk masukan analog dari luar mikrokontroler dapat diterjemahkan oleh mikrokontroler maupun perangkat lain yang membutuhkan masukan berupa nilai digital.

Mikrokontroler PIC16F876 merupakan mikrokontroler yang diproduksi oleh Microchip Inc. Mikrokontroler jenis ini memiliki 8kByte *flash memory*, tiga port I/O yaitu port A, B dan C, masukan analog sebanyak lima *channel* yang telah dilengkapi dengan ADC sebesar 10 bit pada port A. Mikrokontroler PIC16F876 membutuhkan tegangan sebesar 5 volt.



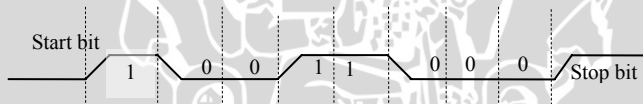
Gambar 2.4 Susunan pin mikrokontroler *PIC16F876* (Malik, 2003)

Mikrokontroler *PIC16F876* memiliki ukuran fisik kecil dengan jumlah kaki 28 pin, ditunjukkan pada Gambar 2.4. Masing-masing pin memiliki fungsi yang berbeda dan beberapa pin memiliki fungsi ganda. Pengaturan fungsi pin dilakukan pada saat inisialisasi mikrokontroler. Pin *MCLR* atau pin nomor 1 berfungsi untuk mengembalikan kondisi mikrokontroler pada keadaan semula (*reset*) pada saat diberi logika rendah. Pin RA0 hingga RA5 adalah *port A* yang berfungsi sebagai *ADC* pada saat menerima masukan nilai analog. Pin Vss sebagai masukan *ground* pada mikrokontroler. Pin nomor 9 dan 10 sebagai masukan pengatur *clock* berupa kristal yang dibutuhkan oleh mikrokontroler. Pin RC0 hingga RC7 merupakan *port C* yang berfungsi sebagai *input* atau *output* dari mikrokontroler. Pada pin nomor 17 dan 18 memiliki fungsi ganda sebagai jalur komunikasi serial antara mikrokontroler dengan perangkat luar. Pin nomor 17 dapat berfungsi sebagai jalur pengirim data (TX) dari mikrokontroler dan pin nomor 18 sebagai jalur penerima data (RX) bila fungsi khusus tersebut diaktifkan pada saat inisialisasi mikrokontroler. Pin Vss memiliki fungsi sebagai jalur masukan sinyal *ground* seperti pada pin nomor 8. Pin Vdd sebagai pin sumber tegangan 5 volt agar mikrokontroler bisa bekerja. Pin RB1 hingga RB7 berfungsi sebagai jalur *input* atau *output* dari mikrokontroler bergantung pada saat inisialisasi.

2.4 Port Serial Komputer

Komunikasi serial adalah proses pengiriman data secara serial atau data dikirim satu per satu secara berurutan. Komunikasi serial memiliki kelebihan pada jangkauan panjang kabel yang lebih jauh dibandingkan dengan paralel. Serial port mengirimkan logika 1 dengan tegangan -3 volt hingga -25 volt dan logika 0 sebagai +3 volt hingga +25 volt.

Komunikasi serial port bersifat asinkron. Komunikasi serial asinkron pada saat pengiriman data harus diawali dengan *start* bit dan diakhiri dengan stop bit. Sinyal *clock* yang merupakan *baud rate* dari komunikasi data dibangkitkan oleh penerima maupun pengirim data dengan frekuensi yang sama. Penerima akan mendeteksi adanya *start* bit sebagai awal pengiriman data dan setelah data dikirim oleh pengirim, penerima akan menunggu stop bit sebagai tanda bahwa data telah selesai dikirim. Pada aplikasinya proses komunikasi asinkron selalu digunakan untuk mengakses komponen-komponen yang memiliki fasilitas *UART* (*Universal Asynchronous Receiver/Transmitter*) seperti port serial *PC* maupun port serial mikrokontroler.

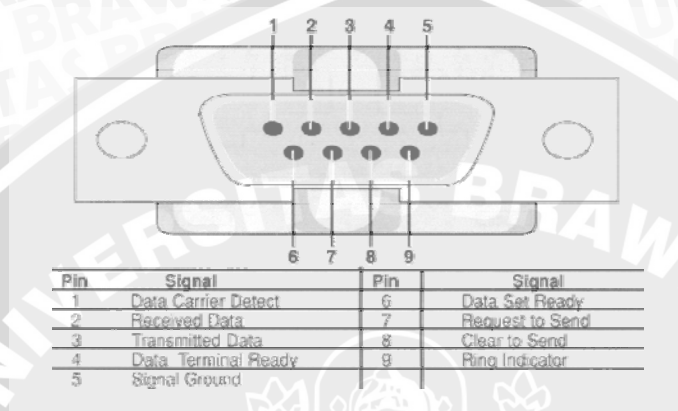


Gambar 2.5 Pengiriman data serial secara asinkron (Budiharto, 2004).

Perangkat keras pada komunikasi serial port dibagi menjadi dua kelompok yaitu *Data Communication Equipment* (*DCE*) dan *Data Terminal Equipment* (*DTE*). Contoh *DCE* adalah modem sedangkan *DTE* adalah terminal serial di komputer. Konektor port serial terdiri dari dua jenis yaitu konektor 25 pin atau biasa disebut dengan *DB25* dan 9 pin atau biasa disebut dengan *DB9* (Budiharto, 2004).

Pin pada *DB9* memiliki fungsi yang berbeda-beda seperti pada Gambar 2.6. Pin yang digunakan untuk komunikasi serial RS-485 antara PC dengan mikrokontroler adalah pin nomor 2, 3 dan 7. Pin *received data* berfungsi sebagai jalur penerima data yang dikirim oleh mikrokontroler. Pin *transmitted data* berfungsi sebagai jalur pengirim data menuju mikrokontroler dan pin *request to send*

sebagai jalur penanda kesiapan pengiriman data dari *PC* menuju mikrokontroler.



Gambar 2.6 Konfigurasi pin pada *DB9* dengan fungsi yang berbeda (Budiharto, 2004).

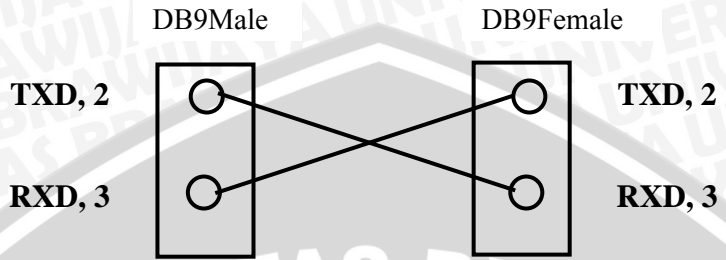


Gambar 2.7 *DB9Male*



Gambar 2.8 *DB9Female*

Pin *DB9Female* (betina) dengan *DB9Male* (jantan) memiliki kesamaan fungsi dari masing-masing pin tetapi perbedaan terjadi pada saat pemasangan keduanya. Fungsi-fungsi pin tersebut dipasang saling bersilangan seperti pin *transmit data* dari *DB9Female* dihubungkan dengan pin *received data* dari *DB9Male* dan pin *received data (RxD)* dari *DB9Female* dihubungkan dengan pin *transmit data (TxD)* dari *DB9Male*.



Gambar 2.9 Pemasangan *DB9 Female* dan *DB9 Male* secara bersilangan.



BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan di Laboratorium Instrumentasi dan Laboratorium Fisika Dasar, Jurusan Fisika, Fakultas MIPA, Universitas Brawijaya pada bulan Agustus 2008 – Juni 2009.

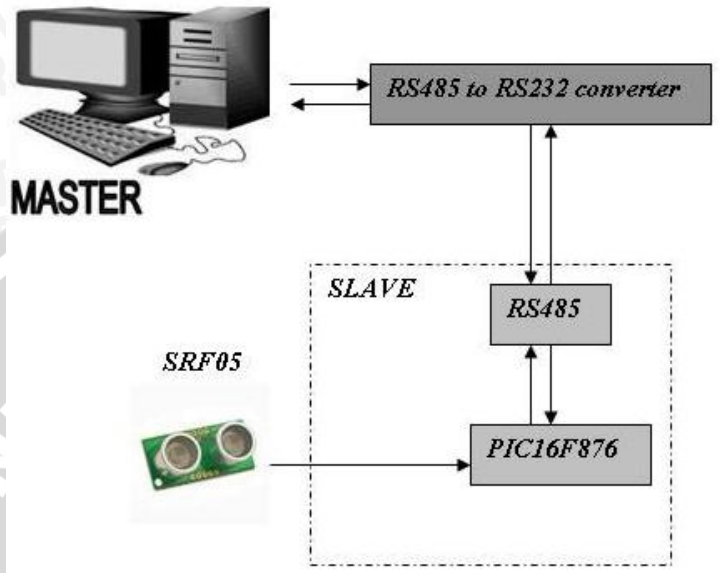
3.2 Tahapan Penelitian

Tahapan penelitian yang dikerjakan adalah sebagai berikut :

1. Studi literatur
2. Merancang dan membuat sistem akuisisi data
3. Merancang dan membuat software komunikasi
4. Penggabungan dan pengujian peranti
5. Proses analisis dan evaluasi kinerja piranti
6. Pembuatan Laporan

3.3 Perancangan Perangkat Keras

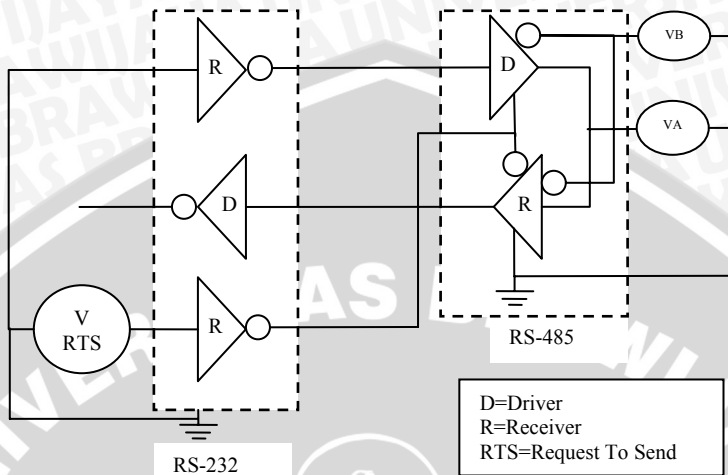
Perancangan alat yang dibuat pada tugas akhir ini, secara keseluruhan ditunjukkan pada Gambar 3.1. Untuk mengukur jarak ketinggian air di suatu tempat, digunakan sensor ultrasonik SRF05 dengan data keluaran langsung dalam bentuk digital dan memiliki waktu respons yang cukup cepat. Variabel perubahan data *time delay* yang diterima oleh sensor dalam bentuk digital akan langsung diterima oleh mikrokontroler PIC16F8786. Disini, mikrokontroler berfungsi sebagai perekam dan penyimpan data pada register yang ada di dalamnya, serta pengendali seluruh bagian dari sistem pada *slave*. Dari mikrokontroler menuju RS-485 sebagai media antar muka antara *slave* dan *master* untuk diteruskan menuju *PC*. RS-485 digunakan untuk memperpanjang jarak komunikasi *master* dan *slave*. Namun, sebelum menuju *PC* terlebih dahulu dilakukan perubahan komunikasi *interface* dari RS-485 menjadi RS-232 agar kompatibel dengan komunikasi *interface* yang ada pada *PC*. Setelah diterima oleh *PC*, data dapat diolah lebih lanjut oleh *PC* sehingga dapat disimpan dan ditampilkan untuk keperluan lebih lanjut.



Gambar 3.1 Perancangan perangkat keras meliputi sensor SRF05, slave, RS-485 To RS-232 Converter dan PC.

3.3.1 Perancangan master

Sistem komunikasi RS-485 membutuhkan pengendali untuk mengatur koneksi jalur pengiriman data. Pengendali koneksi ini dilakukan dengan memanfaatkan sinyal *request to send (RTS)* pada port serial. Jalur *RTS* yang terhubung pada pin *driver enable* memberikan logika tinggi agar *RS-485* dapat bekerja. Sedangkan pada logika rendah akan mengakibatkan kondisi *tristate* sehingga sebuah jalur transmisi akan terputus dan membolehkan jalur lain untuk mengirimkan data meskipun menggunakan jalur kabel yang sama. Sinyal *RTS* digunakan untuk mengeset logika tinggi pada saat data sebelum dikirim dan mengeset pada logika rendah sesudah data dikirim. Pengaturan logika tinggi dan rendah dari jalur *RTS* diatur melalui *software*, dengan blok diagram berikut ini.



Gambar 3.2 Blok diagram RS-232 to RS-485 Converter yang diatur dari software dengan memanfaatkan jalur RTS.

Nilai logika yang dikirimkan oleh *master* melalui *port* serial akan selalu dibalik oleh RS-232 karena jalur pengiriman data melalui *inverter* (pembalik) pada RS-232. *Inverter* akan membalik nilai logika tinggi yang dikirim oleh *master* menjadi logika rendah dan nilai logika rendah menjadi logika tinggi. Perancangan *software* pada *master* untuk membuka koneksi transmisi data dikirimkan pertama kali logika rendah agar RS-485 mengaktifkan *driver enable* yang berlogika tinggi dan segera mengirimkan logika tinggi untuk mengaktifkan *receiver enable* yang berlogika rendah sesudah data dikirim oleh *master*.

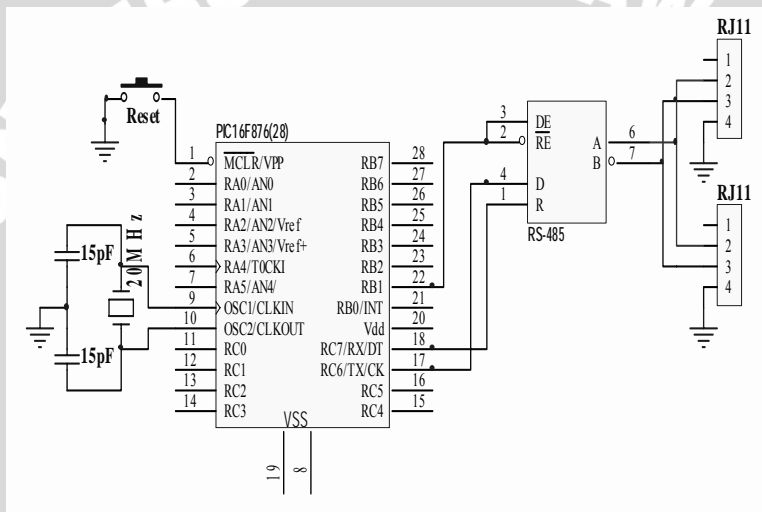
Master secara umum memberikan perintah permintaan pengiriman data dari tiap *slave*. Agar data dari tiap *slave* tidak tertukar, *master* akan mengirimkan karakter kunci yang berbeda pada tiap *slave*. Pada *master* telah ditempatkan kolom-kolom yang akan menyimpan data hasil pengukuran dari tiap *slave*. (Fahmi, 2008)

3.3.2 Perancangan *slave*

Slave terdiri dari sensor SRF05, mikrokontroler dan NIC RS-485. Sensor berfungsi sebagai pendeteksi ketinggian air dengan gelombang ultrasonik. Mikrokontroler berfungsi sebagai perekam data digital yang dihasilkan sensor dan juga sebagai pengendali komunikasi antara *slave* dan *master*. NIC RS-485 sebagai penunjang

komunikasi antara *master* dengan *slave*. *NIC RS-485* diatur oleh mikrokontroler dalam proses transmisi data baik dari *master* menuju *slave* maupun dari *slave* menuju *master*.

Mikrokontroler yang digunakan adalah mikrokontroler *PIC16F876*, seperti pada Gambar 3.3 dibawah ini. Sistem ini memanfaatkan *port B* sebagai masukan digital yaitu RB0 dan RB1. Kedua *channel* tersebut secara berurutan menjadi masukan dan keluaran dari dan ke sensor ultrasonik. Sebagai pengontrol aktifasi dari *RS-485* digunakan *port C*.

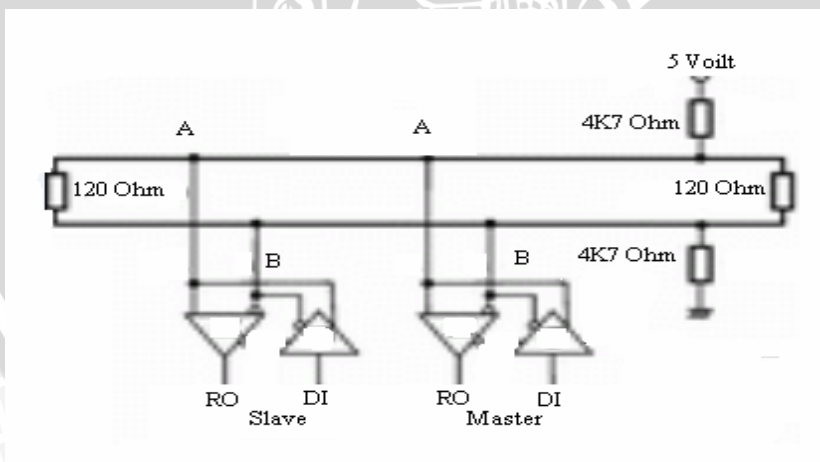


Gambar 3.3 Rangkaian *slave* berupa mikrokontroler dilengkapi *NIC485*.

3.3.3 Perancangan komunikasi *RS-485*

Pada komunikasi serial *RS-485*, output berupa perbedaan tegangan antara pin A dan pin B *IC max485*, dengan selisih tegangan minimal adalah 200 mV. Saluran komunikasi multi-drop adalah sepasang kabel yang panjangnya tidak lebih dari 1200 meter, pada kedua ujung saluran masing-masing dipasang resistor 120 Ohm yang menghubungkan kedua kabel. Resistor tersebut dimaksud untuk mengurangi terjadinya gelombang pantul dalam saluran, yang sering terjadi pada transmisi kecepatan tinggi. Selanjutnya pada saluran tersebut bisa dipasangkan maksimal 32 chip *NIC485 Multi-drop RS485 Transceiver*, kaki A (pin 6) dari masing-masing *IC* harus dihubungkan pada seutas kabel pembentuk saluran yang sama, dan

kaki B (pin 7) dihubungkan ke kabel yang lain. Karena saluran dipakai bersama oleh banyak *transceiver*, agar *output Line Driver* dari masing-masing *transceiver* tidak berbenturan, dalam rangkaian saluran komunikasi multidrop ditentukan semua *output Line driver* harus dalam keadaan non-aktif ($DE=0$), kecuali *Line Driver* dari *transceiver* yang berfungsi sebagai induk (*Master*) yang boleh aktif ($DE=1$). Saat beroperasi *Master* menghubungkan *Slave*, setelah itu *Master* menonaktifkan *Line Drivernya*, *Slave* akan mengaktifkan *Line drivernya* dan mengirimkan informasi kesaluran, setelah itu *Slave* tersebut menonaktifkan kembali *Line Drivernya* dan kembali *Master* mengaktifkan *Drivernya* untuk menghubungi *Slave* lagi. Dengan demikian *Master* berfungsi untuk mengendalikan saluran, dan komunikasi yang terjadi di saluran adalah komunikasi *half-duplex*, yakni komunikasi dua arah secara bergantian. Pada saat pergantian aktivitas *Line Driver*, *Master* dan *Slave*, bisa terjadi satu saat secara bersamaan semua *Line driver* tidak aktif, akibatnya saluran menjadi mengambang dan keadaan logika dari saluran tidak menentu. Untuk mencegah terjadinya hal tersebut, pada saluran ditambahkan 2 buah resistor masing-masing bernilai 4K7 Ohm, resistor yang terhubung ke A dihubungkan ke +5 Volt dan resistor yang terhubung ke B dihubungkan ke ground, dengan cara begini kalau semua *Line Generator* tidak aktif bisa dipastikan saluran dalam keadaan '1'. Berikut rangkaian jaringan multidrop RS 485.

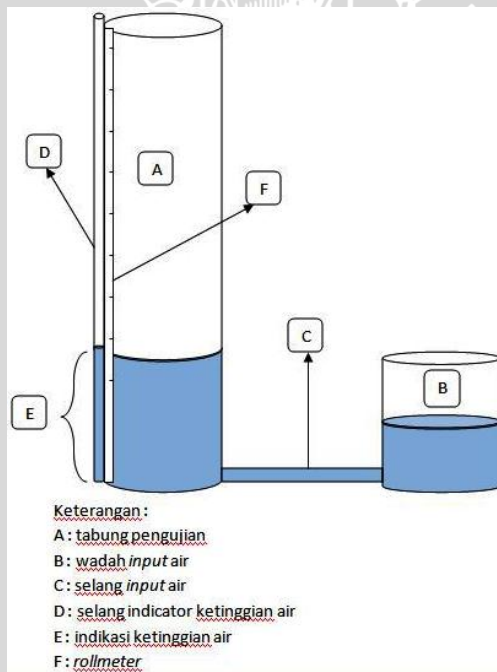


Gambar 3.4 Jaringan komunikasi multi-drop

Secara umum, *master* mengubah level komunikasi RS-485 menjadi RS-232 dan memberikan perintah kepada *slave* dan menerima data dari *slave*. Data yang diterima akan diolah oleh *master (PC)* dan ditampilkan.

3.3.4 Perancangan tabung pengujian

Tabung pengujian dibuat sebagai prototipe tempat naik turunnya air, yang dideteksi perubahan naik turunnya. Tabung terdiri dari pipa sepanjang 1 meter dengan diameter 10 cm, bagian bawahnya terhubung dengan selang transparan sepanjang 1 meter yang menempel di sisi pipa sebagai indikator posisi air di dalam pipa. Bagian bawah pipa juga terhubung dengan wadah air melalui selang transparan. Skema rancangannya dan implementasinya dapat dilihat pada gambar berikut ini.



Gambar 3.5 Skema tabung pengujian



Gambar 3.6 Implementasi tabung pengujian yang dibuat

Prinsip kerjanya adalah air dimasukkan melalui wadah tempat air yang terhubung ke bagian bawah pipa melalui selang transparan semakin banyak air yang masuk ke wadah, maka volume air dalam pipa akan semakin besar dan akibatnya permukaan air dalam pipa bergerak naik. Posisi permukaan air dalam pipa diindikasikan dengan selang transparan yang menempel di sisi lain pipa. Dengan air yang masih ada dalam wadah, posisi permukaan air dalam pipa dapat diubah-ubah dengan menaikturunkan wadah sebatas ketinggian pipa, jika wadah naik, maka air yang berpindah ke dalam pipa semakin banyak, sehingga permukaan air juga semakin naik, dan sebaliknya

jika wadah diturunkan, maka air dalam pipa akan bergerak menuju wadah sehingga posisi permukaan air dalam pipa akan turun.

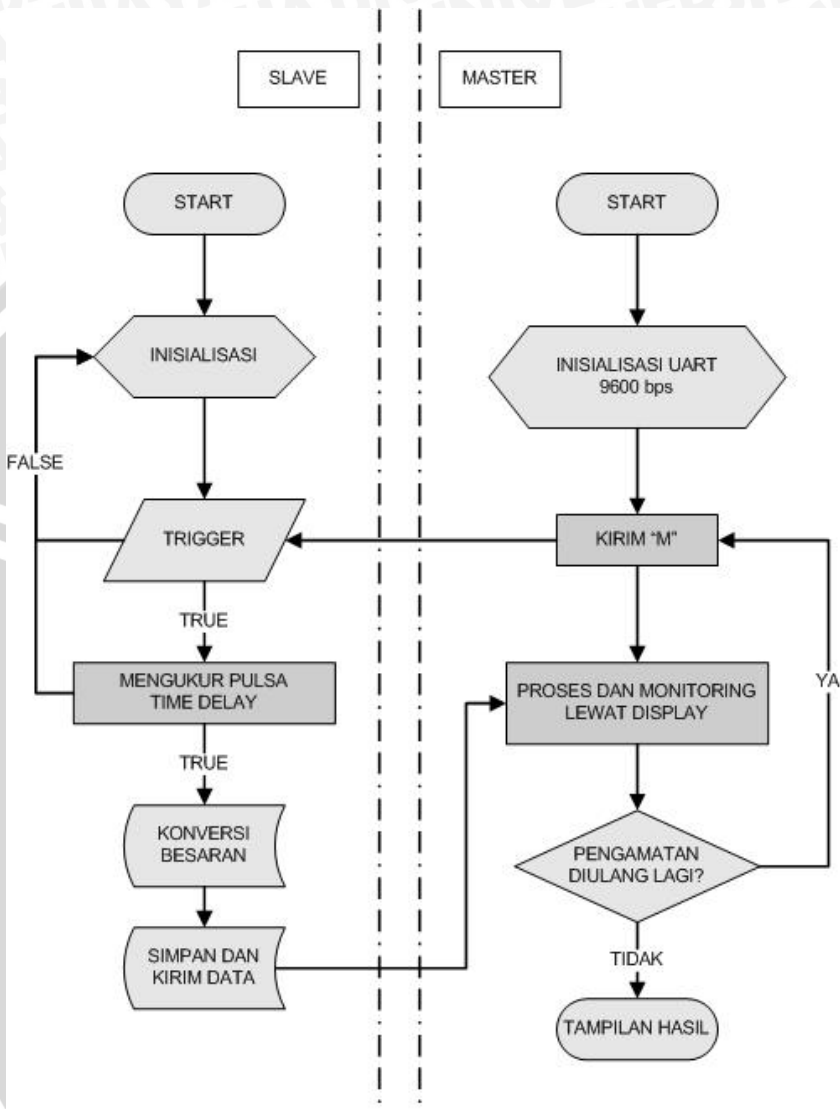
Proses naik dan turunnya permukaan air ini dapat terdeteksi oleh sensor ultrasonik yang terpasang di atasnya, sehingga data tentang perubahan ketinggian air dalam pipa akibat gerakan naik dan turun, akan ditampilkan di *PC*.

3.4 Perancangan Perangkat Lunak

3.4.1 Perancangan program keseluruhan

Program keseluruhan berisi aliran data dari *master*, *slave*, dan sensor. Dimulai dari subrutin inialisasi dan *trigger* mikrokontroler, serta pengaktifan sensor ultrasonik.

Prosedur pada Gambar 3.7 dibawah ini dijalankan dengan *software Delphi7*. Program dimulai ketika kita mulai menginisialisasi master dan mikrokontroler, kemudian membuka koneksi *serial port* dari PC menuju *converter* dan *slave* dengan memilih *baudrate* dan *port* yang akan mengambil data dari *slave*. Pada saat program dijalankan dilakukan pengiriman karakter kunci "M" sebagai alamat pada *slave*. Apabila karakter kunci yang dikirim sesuai dengan *slave* penerima maka *slave* tersebut akan mengaktifkan sensor ultrasonik. Apabila tidak sesuai, maka dilanjutkan dengan karakter berikutnya hingga karakter yang diterima oleh *slave* sesuai dengan yang dikirimkan. Sensor aktif jika *trigger* mengirimkan pulsa dan hasil monitoring dikonversi menjadi jarak. Sesudah data diambil, maka data direkam oleh mikrokontroler, yang kemudian dengan perintah program *Delphi7* akan mengirimkan data ke *PC* untuk menunjukkan proses *monitoring*. Jika *monitoring* tidak dilanjutkan, maka data hasilnya akan ditampilkan, yang bisa dikirim dalam format MS Excel. Jika *monitoring* diulang, maka pengukuran dimulai lagi dari pengiriman karakter kunci sampai *monitoring* dihentikan.



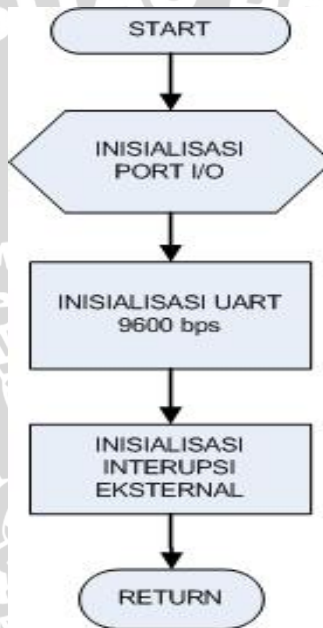
Gambar 3.7 Diagram transfer data keseluruhan

3.4.2 Inisialisasi dan *trigger*

Perangkat lunak yang dibuat adalah untuk menjalankan mikrokontroler dengan bahasa *assembly* yang dikendalikan dari PC menggunakan bahasa pemrograman *Delphi7*. Pada mikrokontroler terdiri dari beberapa subrutin yaitu inisialisasi awal, *trigger*,

pengukuran *time delay*, dan interupsi eksternal untuk memecah berapa kali gelombang ultrasonik memancar dan diterima oleh sensor.

Tahap awal perancangan perangkat lunak adalah menginisialisasi mikrokontroler agar dapat bekerja dan berfungsi sebagai pengendali komunikasi antara *slave* dengan *master*. Inisialisasi disini meliputi inisialisasi *baudrate* yang digunakan, jumlah masukan digital, port keluaran pengatur komunikasi RS-485, dan jalur pengiriman data.

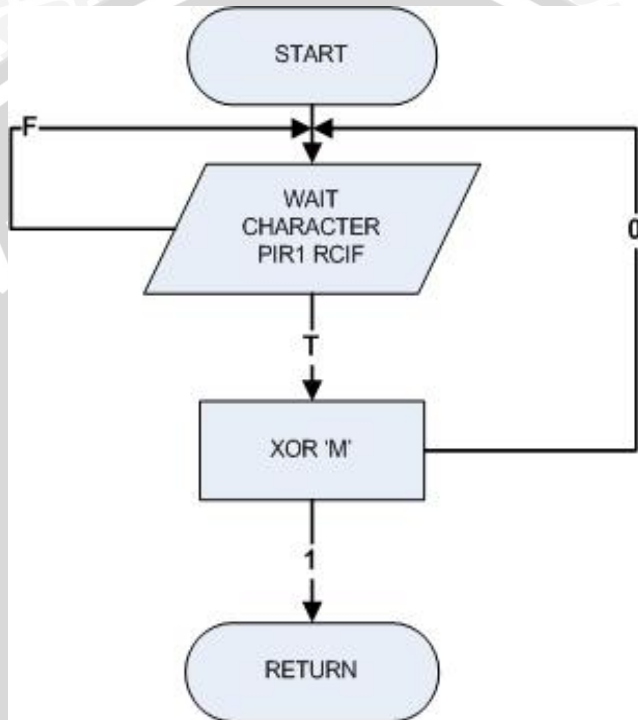


Gambar 3.8 Prosedur inisialisasi mikrokontroler.

Prosedur inisialisasi mikrokontroler ditunjukkan pada gambar 3.8, diawali dengan pengesetan *I/O* pada mikrokontroler, yaitu pengesetan dua masukan digital pada port B, port C sebagai keluaran serta pengesetan fungsi *RX/TX* untuk proses pengiriman maupun penerimaan data, mengatur komunikasi serial dengan kecepatan transfer 9600 *bps* dan pengaturan interupsi eksternal.

Setelah prosedur inisialisasi selesai, mikrokontroler akan menunggu karakter kunci dari *PC* yaitu 'M', hal ini agar mikrokontroler bisa menjalankan perintah selanjutnya. Pengiriman

karakter dari *PC* diatur menggunakan timer agar pengambilan data bisa dilakukan secara kontinu dengan interval pengukuran yang dapat diatur. Proses ini ditunjukkan pada Gambar 3.9.

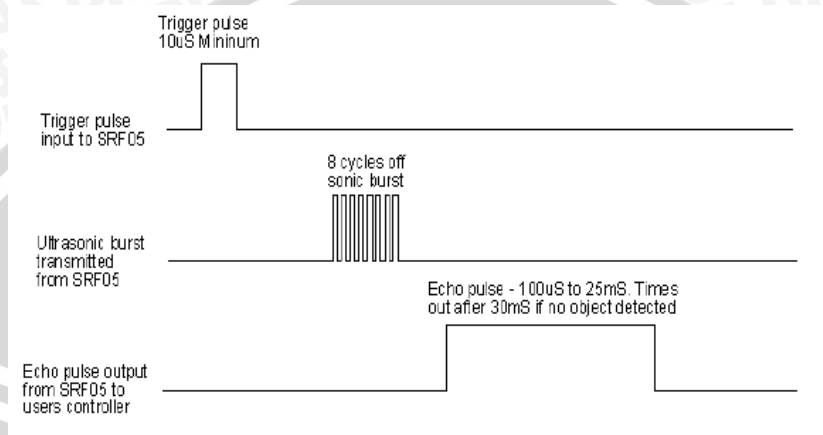


Gambar 3.9 Prosedur subrutin *trigger*.

3.4.3 Pengaktifan sensor ultrasonik

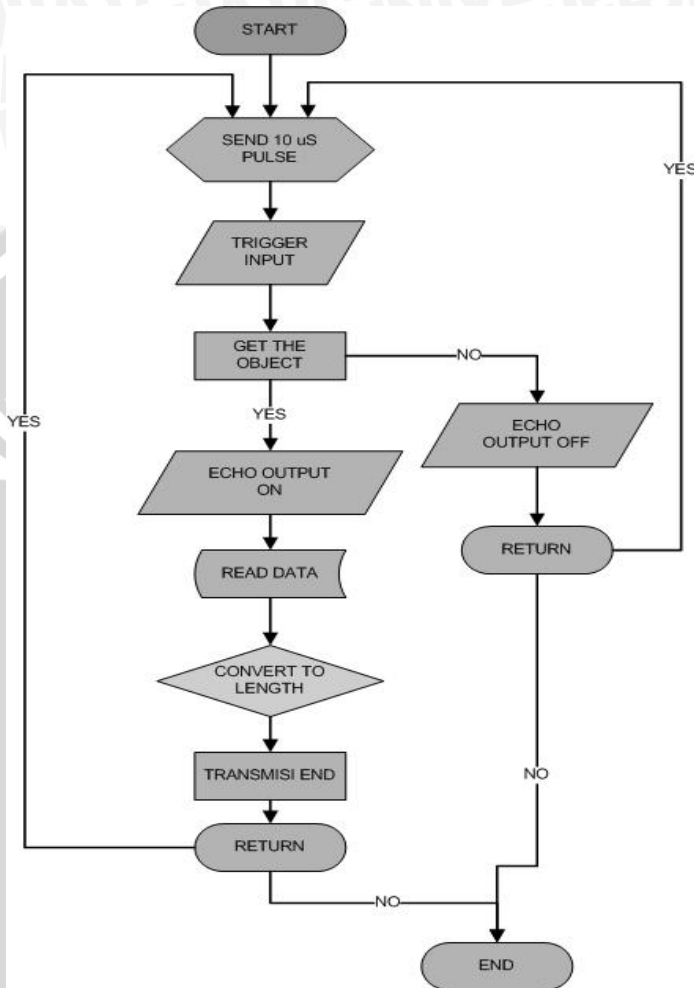
Sensor ultrasonik terhubung ke mikrokontroler pada *slave* dengan konfigurasi pin tersendiri. *Trigger input* dihubungkan ke pin RB7 pada mikrokontroler dan *echo output* dihubungkan ke pin RB6. Pulsa sebesar 10 μ s dikirim ke trigger input untuk mulai pengukuran. Sensor memancarkan 8 pulsa ultrasonik pada frekuensi 40 kHz dan membuat pin echo berlogika tinggi, sehingga sensor menghasilkan waktu *delay* pulsa gelombang ultrasonik yang kembali setelah memantul obyek penghalang. Sinyal *echo output* ini dikirim ke mikrokontroler, yaitu waktu dalam bentuk heksadesimal. Kemudian

keluaran dikonversi menjadi jarak yang kemudian keseluruhan data tersebut ditampilkan pada PC.



Gambar 3.10 Diagram *clock* SRF05

Jika tidak mendeteksi apapun, maka sensor akan mematikan sinyal echo setelah 30 ms. Pulsa gelombang ultrasonik dari SRF05 bisa dipicu setiap 50 ms atau 20 kali setiap detik. Urutan aktivasi sensor ultrasonik diatas ditunjukkan pada Gambar 3.11.



Gambar 3.11 Diagram pembacaan pulsa *delay* dan jarak oleh sensor ultrasonik

3.5 Pengujian Akuisisi Data

3.5.1 Pengujian *master*

Pengujian master dilakukan untuk mengetahui apakah rangkaian pengubah komunikasi serial antara PC dengan mikrokontroler dapat bekerja dengan baik. Hal ini dapat dilihat melalui transfer datanya, apakah data dari *slave* dapat diterima dan ditempatkan secara tepat

oleh *master*. Pengujian dilakukan dengan menghubungkan rangkaian *converter* pada mikrokontroler dan port serial *PC*. Mikrokontroler terlebih dulu diisi dengan beberapa karakter sebagai data serta satu karakter kunci, yaitu 'M'. Data yang telah diisikan pada mikrokontroler akan dikirimkan pada *PC*, bila ada kesesuaian antara karakter kunci yang dikirim oleh *PC* dengan yang diterima oleh mikrokontroler. Tampilnya data pada *PC* dari mikrokontroler menunjukkan bahwa perubahan komunikasi serial menggunakan *RS-485 to RS-232 converter* dapat berfungsi dengan baik.

3.5.2 Pengujian *slave*

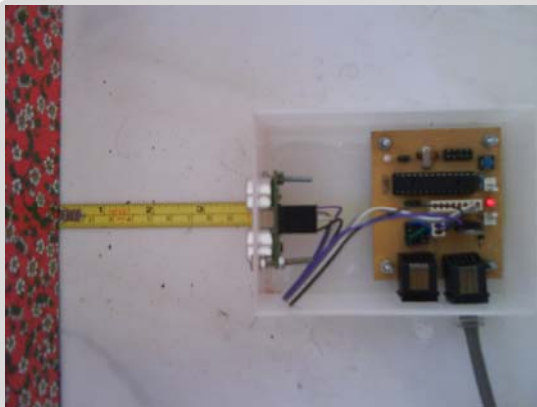
Pengujian ini bertujuan untuk mengetahui apakah sistem dari *slave* yang telah dirancang dapat bekerja dengan baik. Pengujian diawali dengan pengujian *port A* sebagai masukan dan *port B* sebagai keluaran. Sebelumnya terlebih dahulu dilakukan konfigurasi port mikrokontroler. Konfigurasi disini dilakukan dengan terlebih dahulu mengkonfigurasi *port A* sebagai masukan dan *port B* sebagai keluaran. Masukan yang diberikan pada *port A* berupa nilai tegangan antara 0-5 volt yang nilainya dapat diubah-ubah. Keluaran pada *port B* berupa *LED* yang akan mengalami perubahan menyala dan mati apabila diberikan nilai masukan yang berubah-ubah. *LED* menyala berarti bernilai 1 atau logika tinggi dan *LED* mati berarti bernilai 0 atau logika rendah.

Pengujian dilanjutkan dengan komunikasi serial melalui pin *RX/TX* mikrokontroler dengan *PC*. *Software Delphi* digunakan sebagai pendukung komunikasi serial ini. Sebelum dilakukan pengujian komunikasi serial, dibuat *software* penampil data pada *PC* dengan memanfaatkan komponen-komponen pendukung pada *Delphi* yang sesuai. Pengujian komunikasi dilakukan dengan mengisi mikrokontroler beberapa karakter sebagai data dan satu karakter sebagai karakter kunci, untuk mengambil data yang telah diisikan pada mikrokontroler, diberikan karakter kunci dari *PC* pada mikrokontroler. Data yang telah diisikan pada mikrokontroler akan dikirimkan pada *PC* bila ada kesamaan karakter kunci yang dikirim oleh *PC* dengan yang ada pada mikrokontroler. Data yang dikirim oleh mikrokontroler akan ditampilkan oleh *PC* sebagai tanda bahwa komunikasi serial mikrokontroler dengan *PC* dapat dilakukan.

3.5.3 Pengujian sensor ultrasonik

Pengujian sensor dilakukan untuk mengetahui apakah sensor ultrasonik SRF05 dapat bekerja mengukur *time delay* gelombang yang dipancarkan, sehingga bisa didapatkan jarak yang sesuai dengan nilai sebenarnya, dalam hal ini menggunakan pembanding rollmeter sepanjang 300 cm.

Pengujian sensor meliputi pengaturan *start* dan *stop* nilai pulsa pada masukan, sehingga sensor bisa bekerja, yaitu dengan memancarkan dan menerima gelombang ultrasonik setiap kali diberikan pulsa dari mikrokontroler yang dikendalikan melalui *PC*.



Gambar 3.12 Pengujian sensor ultrasonik

Pengujian sensor dilakukan dengan cara meletakkan penghalang di depan sensor ultrasonik seperti pada Gambar 3.12, di permukaan penghalang tersebut dipasang rollmeter tegak lurus permukaannya, dan sensor diletakkan sejajar dengan rollmeter sebagai pembanding hasil pengukuran sensor. Pengujian jarak sensor ultrasonik dilakukan dengan mendekatkan dan menjauhkan posisi sensor terhadap penghalang setiap 5 cm, hal ini dilakukan untuk mengetahui respons sensor ketika diberikan objek yang dengan perubahan jarak bervariasi, selain itu juga untuk mengetahui sejauh mana sensor dapat melakukan pengukuran.

Data *time delay* dari pengukuran pemancaran dan penerimaan gelombang ultrasonik oleh sensor dikonversi menjadi data perubahan jarak sensor ke penghalang dengan persamaan 2.1. Nilai jarak yang

dihasilkan dari konversi persamaan tersebut ditampilkan di PC dan dibandingkan dengan pengukuran manual menggunakan *rollmeter*.

3.5.4 Pengujian aplikasi alat keseluruhan

Pengujian ini dilakukan untuk mengetahui apakah sistem secara keseluruhan dapat bekerja sesuai dengan perancangan. Pengujian sistem dilakukan secara bersamaan mulai dari pengukuran dan monitoring oleh *master* dan *slave*. Pada *slave* yang terdiri dari sensor dan mikrokontroler dihubungkan dengan *master* yang terdiri dari rangkaian *RS-232 To RS-485 converter* yang telah terhubung secara serial dari *PC*. Pengujian dilakukan dengan cara membuka koneksi awal antara *master* dengan *slave* melalui *master*. Bila koneksi berhasil maka waktu dan jarak yang terukur oleh sensor akan ditampilkan oleh *PC*.

Pengambilan data dilakukan dengan memasang sensor ultrasonik diatas pipa setinggi 2,5 meter yang terbuka menghadap ke dalam pipa tersebut, di sisi pipa tersebut terpasang selang transparan yang mengindikasikan posisi air dalam pipa, serta rollmeter untuk mengetahui ketinggian air yang sebenarnya sebagai pembanding. Air dimasukkan ke dalam pipa melalui bagian bawah pipa, sehingga permukaan air dalam pipa akan bergerak naik, bersamaan dengan itu, alat memonitoring kenaikan air setiap.

Apabila sistem dapat bekerja, maka data *time delay* yang terukur oleh sensor akan disimpan dalam register pada mikrokontroler, kemudian dikirimkan oleh *slave* menuju *master* akan ditampilkan pada *PC* berupa nilai waktu dan jarak, dalam hal ini ketinggian air. Data hasil monitoring yang telah diambil dapat diakses ulang oleh *master* yang berfungsi sebagai media penyimpanan data hasil monitoring untuk keperluan lebih lanjut.

BAB IV

HASIL DAN PEMBAHASAN

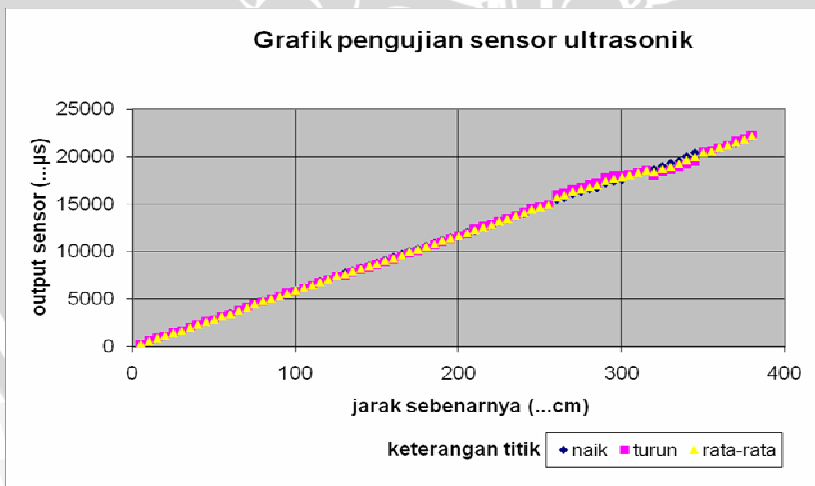
Pengujian dilakukan untuk mengetahui sejauh mana sistem dari sensor ultrasonik SRF05 hingga pendukung komunikasi dapat bekerja sesuai dengan perancangan hardware dan software.

Pengujian alat disini meliputi pengujian sensor ultrasonik dan pengujian software akuisisi data oleh mikrokontroler sebagai pengambilan data.

4.1 Hasil Pengujian Sensor Ultrasonik

Berdasarkan hasil beberapa pengujian yang telah dilakukan untuk kalibrasi, sensor ultrasonik dapat bekerja sesuai dengan yang diharapkan meskipun pada data didapat nilai yang tidak stabil.

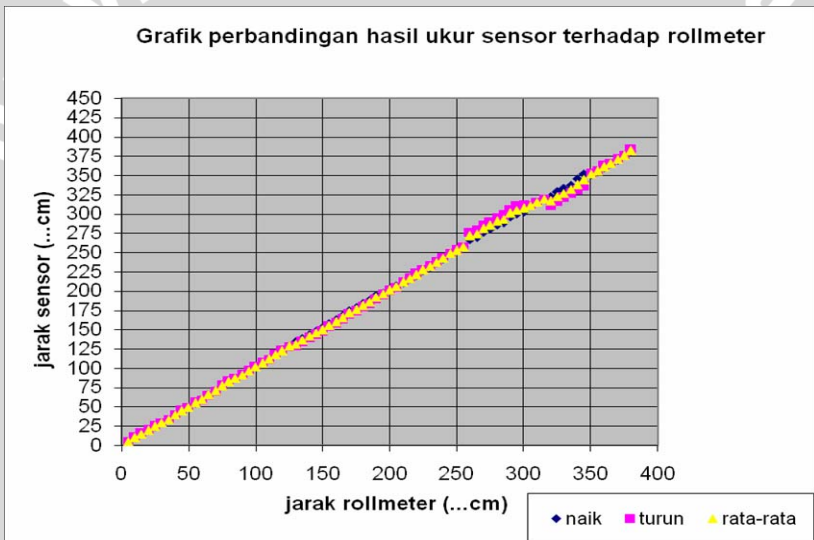
Waktu respon sensor yang kadang masih terlambat menyebabkan perubahan data yang tidak stabil, hal ini dapat dilihat pada grafik 4.1 tentang perbandingan output sensor terhadap hasil ukur rollmeter,



Gambar 4.1 Grafik pengujian sensor ultrasonik sebagai kalibrasi, menggunakan rollmeter sebagai pembandingan.

Data output sensor ultrasonik yang didapatkan dalam orde μs selanjutnya dibagi dengan 58 untuk langsung mendapatkan variabel jarak dalam orde cm. Atau dimasukkan dalam persamaan 2.1 sehingga bisa didapatkan jarak yang terukur oleh sensor ultrasonik.

Data hasil pengolahan tersebut kemudian dibandingkan dengan hasil pengukuran rollmeter dalam Gambar 4.2 dibawah ini untuk mengetahui sejauh mana tingkat akurasi pengukuran oleh sensor tersebut.



Gambar 4.2 Grafik perbandingan hasil pengujian sensor terhadap pengukuran rollmeter

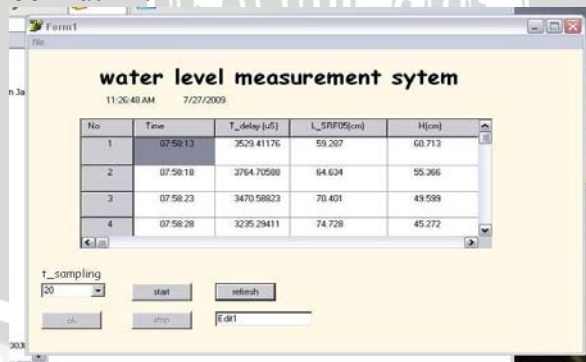
Pengaturan pada sensor ultrasonik dilakukan secara tetap karena sensor bekerja secara konstan memancarkan gelombang ultrasonik yang memiliki kecepatan tetap ($v = 340 \text{ m/s}$). Mulai pemancaran gelombang oleh *transmitter* sensor hingga mengenai permukaan penghalang, kemudian memantul dan diterima kembali oleh *receiver* sensor diperoleh waktu perambatan gelombang (*pulsa/time delay*) yang dikonversi sebagai jarak jarak dari sensor ke penghalang.

Jarak sensor ke penghalang diubah menjauhi penghalang setiap 5 cm (ditunjukkan oleh rollmeter sebagai pembanding) sampai sensor mencapai batas maksimal kerja monitoring jarak, dan itu diperoleh pada jarak 380 cm, setelah itu sensor tidak dapat mengukur dan monitoring dihentikan. Hal ini diasumsikan bahwa sensor bisa mendeteksi obyek yang bergerak naik. Selanjutnya dari jarak maksimum tersebut, sensor kembali digerakkan mendekati penghalang setiap 5 cm, dan hal ini diasumsikan bahwa sensor bisa mendeteksi jarak ke obyek yang bergerak turun dibawahnya.

Grafik tersebut menunjukkan kerja sensor masih memerlukan waktu respons yang belum stabil, terutama untuk jarak yang semakin jauh terhadap penghalang. Sedangkan untuk jarak yang lebih dekat, kerja sensor cukup akurat sesuai hasil yang sebenarnya yang didapat dari rollmeter.

4.2 Hasil Pengujian *Software*

Pengujian *software* meliputi pengujian komunikasi antara *PC* dengan *slave* serta tampilan data ketinggian air, di komputer. Dari pengujian *software* yang telah dilakukan, komunikasi antara tiap *slave* dengan *PC* dapat dilakukan. Hal ini ditunjukkan dari tampilnya nilai pengukuran oleh sensor pada bagian *slave*. *PC* juga dapat merekam dan menyimpan nilai pengukuran tersebut. Proses monitoring dilakukan dengan *software* berikut ini.

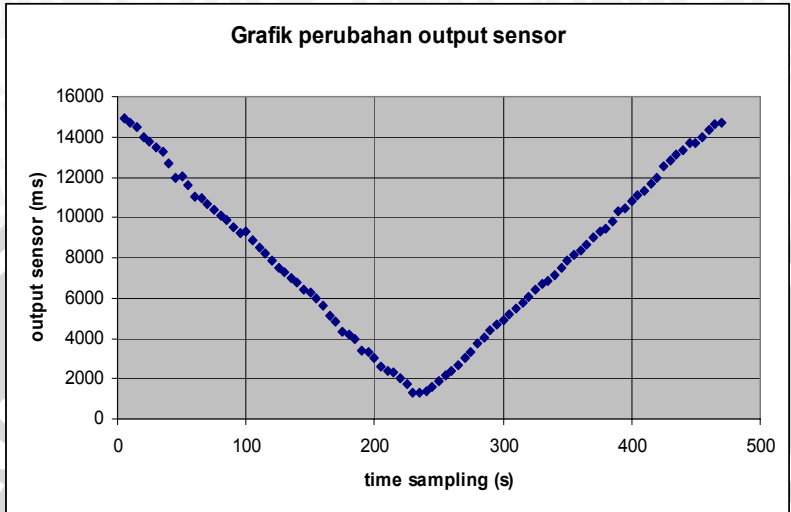


Gambar 4.3 Tampilan *software*

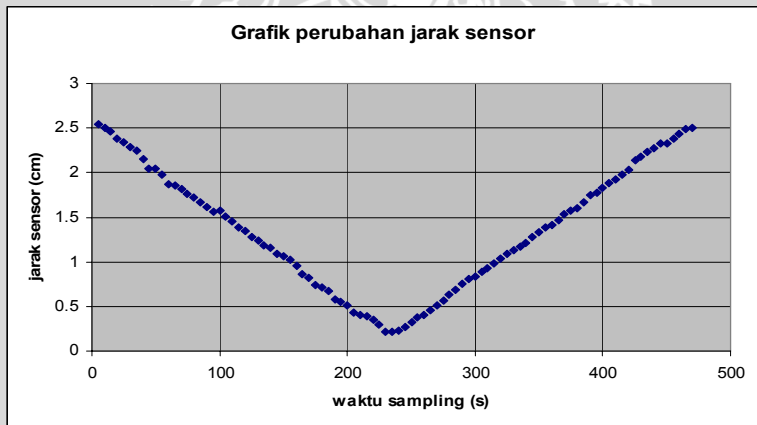
Pemilihan *serial port* pada *PC* untuk membuat komunikasi serial yang dilakukan antara *master* dengan *slave* lebih fleksibel sehingga tidak bergantung pada satu *COM*. Adanya pengaturan *baudrate* pada tampilan merupakan gambaran diperlukannya kesesuaian *baudrate* yang harus dimiliki oleh mikrokontroler dengan *PC* dalam berkomunikasi. *Baudrate* yang dibutuhkan antara keduanya adalah sebesar 9600 bps, karena telah mampu mendukung komunikasi antara *master* dengan *slave* tanpa adanya keterlambatan pengiriman data dari *slave* menuju *master*.

PC mengatur kerja RS-485 melalui pin *RTS Control* dari *port serial*. Dimana untuk melakukan pengiriman data maka *RTS* diset berlogika rendah atau bernilai 0 sehingga RS-485 memiliki logika tinggi, sedangkan saat pengiriman selesai maka *RTS* diset berlogika tinggi atau bernilai 1 sehingga RS-485 berlogika rendah. Hal ini dilakukan pula oleh *slave* melalui mikrokontroler, pada saat *slave* melakukan pengiriman data maka mikrokontroler akan mengaktifkan pin *DE* pada RS-485 dengan memberikan logika tinggi pada RS-485 melalui port B. Bila pengiriman telah selesai maka mikrokontroler akan memberikan nilai logika rendah pada RS-485 (Fahmi, 2008).

Hasil pengujian *software* ultrasonik yang ditampilkan oleh *PC* berupa grafik yang ditunjukkan pada Gambar 4.4 dan Gambar 4.5. Pengujian *software* disini dalam bentuk output sensor berupa waktu dan juga jarak hasil konversi yang pengukurannya dibandingkan dengan rollmeter.



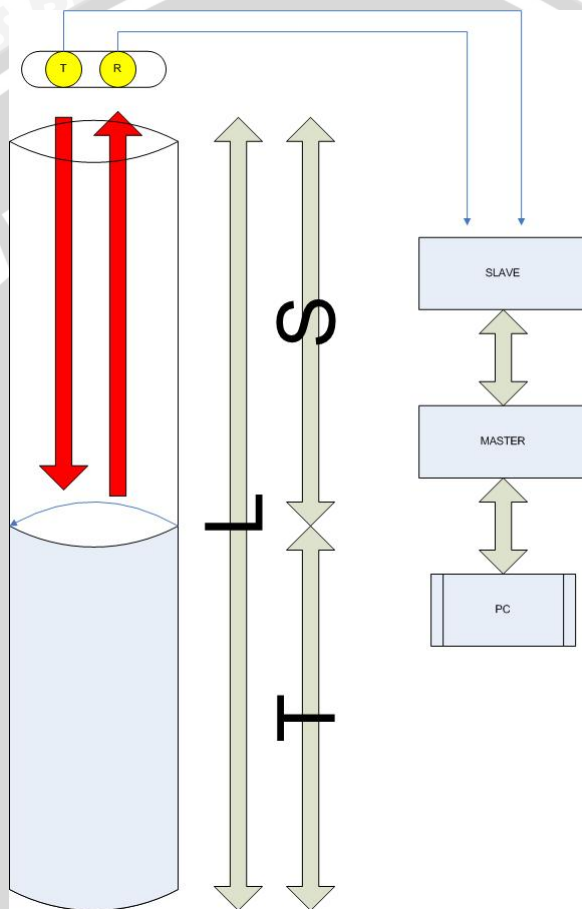
Gambar 4.4 Perubahan time delay/output sensor



Gambar 4.5 Perubahan jarak sensor akibat naik turun permukaan air

Gambar 4.4 merupakan grafik waktu *delay* sensor, sedangkan gambar 4.5 merupakan grafik perubahan jarak yang terukur oleh sensor. Titik-titik yang bergerak turun kemudian naik menunjukkan perubahan jarak dari sensor ke permukaan air, sehingga apabila diketahui jarak dasar air ke sensor, dapat

diketahui tinggi air yang sebenarnya yang berubahannya berbanding terbalik dengan grafik tersebut. Lebih jelasnya, proses monitoring ditunjukkan pada gambar berikut ini.



Keterangan :

L : Ketinggian pipa pengujian keseluruhan

S : Jarak yang terdeteksi oleh sensor

T : Ketinggian air sebenarnya

Gambar 4.6 Proses monitoring ketinggian air

Hasil pengujian pada gambar 4.4 menunjukkan adanya kesesuaian nilai ketinggian air yang terukur oleh sensor terhadap jarak yang sebenarnya. Kesesuaian ditunjukkan oleh grafik berupa perubahan titik-titik pengukuran yang membentuk pola linear meskipun ada sedikit titik yang kurang stabil, terutama untuk pengukuran jarak yang mendekati jangkauan maksimal sensor. Hal ini menunjukkan keterlambatan respon dari *slave* dibandingkan dengan pengukuran *rollmeter* meskipun setelah beberapa saat memiliki kesesuaian.

Selisih waktu yang dimiliki oleh *slave* hasil perancangan dengan hasil pengukuran sebenarnya diakibatkan oleh waktu yang dibutuhkan oleh sensor untuk mencapai kondisi ideal pada saat dimulainya monitoring. *Datasheet* sensor ultrasonik SRF05 memberikan informasi bahwa sensor SRF05 membutuhkan waktu 30 ms untuk mencapai kondisi ideal tersebut. Respon waktu tersebut menyebabkan adanya keterlambatan perubahan antara hasil pengukuran yang ditampilkan dari *slave* dengan hasil pengukuran melalui *rollmeter*.

Recommended Standard-232 (RS-232) merupakan standar komunikasi serial yang dikeluarkan *Electronic Industries Association (EIA)* untuk menghubungkan perangkat serial secara *point to point*, yaitu antara 1 master dan 1 slave dengan menggunakan 1 port saja. Standar komunikasi serial memiliki jangkauan yang jauh dan kecepatan komunikasi data relatif rendah. Sinyal RS-232 dikirimkan dalam bentuk level tegangan yang bergantung pada *ground* dan bekerja dengan baik untuk komunikasi *point to point* atau satu *port* hanya dapat berkomunikasi dengan satu perangkat seperti *COM1* hanya untuk *mouse* dan *COM2* untuk modem. RS-232 membutuhkan sebuah *ground* bersama antara *PC* dengan perangkat yang terhubung. Panjang kabel terbatas 100-200 *feet* untuk komunikasi asinkron dan 50 *feet* untuk komunikasi sinkron (www.hw-server.com/docs/rs485.html).

Standar RS-232 memiliki beberapa keterbatasan seperti adanya *ground* bersama antara *Data Terminal Equipment (DTE)* dan *Data Circuit-Terminating Equipment (DCE)* sehingga sinyal pada jalur yang sama cukup mustahil untuk terhindar dari noise. Selain itu mengakibatkan keterbatasan panjang kabel yang kurang lebih terbatas hanya 30 sampai 60 meter. Level tegangan yang bergantung pada *ground* hanya sesuai untuk komunikasi *point to point* atau 1 *transmitter* untuk 1 *receiver*.

Hasil prototipe untuk monitoring ketinggian air ini terbukti bisa mendeteksi bagaimana perubahan naik turunnya permukaan air yang menjadi obyek yang terdeteksi oleh sensor, hal ini bisa dilihat dari grafik data yang didapatkan. Sehingga alat ini bisa diaplikasikan untuk mengetahui adanya gelombang yang besar sejauh jangkauan sensor yang dipakai.



BAB V KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah piranti pengamatan ketinggian air yang dihasilkan dapat bekerja dengan baik, hal ini dibuktikan dengan dapat dilakukannya *monitoring* secara *online* untuk mengamati gerak perubahan naik turun permukaan air oleh sensor ultrasonik. Sensor ultrasonik SRF05 dapat melakukan pengukuran dengan jangkauan maksimal yang dapat terbaca yaitu 380 cm.

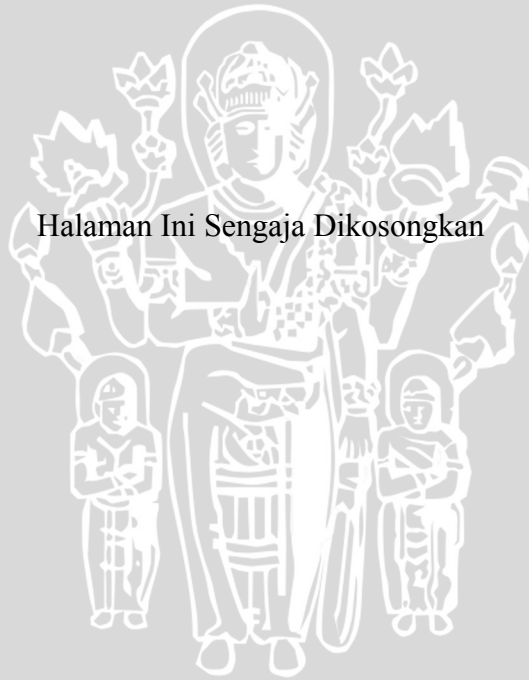
Piranti ini memiliki 3 komponen utama, yaitu *PC* sebagai *master* yang berfungsi sebagai media pengontrol *datalogger* keseluruhan sistem, mikrokontroler sebagai *slave* yang berfungsi sebagai pengendali keseluruhan sistem instrumentasi dengan program yang dimasukkan ke dalamnya dan juga sebagai media pengambil data pada objek yang dimonitor kemudian disimpan sementara, serta sensor ultrasonik SRF05 sebagai sensor pendeteksi jarak yang dapat bekerja dengan baik untuk monitoring ketinggian air secara kontinyu dengan sinyal digital yang dikirim ke *PC* melalui komunikasi antara *slave* dan *master*.

5.2. Saran

Saran yang diberikan setelah melakukan penelitian adalah perlunya penyempurnaan lebih lanjut untuk *software* pendukung segala bentuk kebutuhan aplikasi sistem, khususnya sebagai peringatan adanya tsunami. Penyempurnaan juga dilakukan pada kesesuaian antara hasil perancangan dengan sarana kalibrasi yang dipakai pada saat proses pengujian.

Perlu dilakukan penggantian sensor ultrasonik yang lebih baik. Sensor SRF05 bisa digantikan dengan sensor ultrasonik lain yang memiliki waktu respon lebih cepat serta jangkauan yang lebih besar.

UNIVERSITAS BRAWIJAYA



Halaman Ini Sengaja Dikosongkan

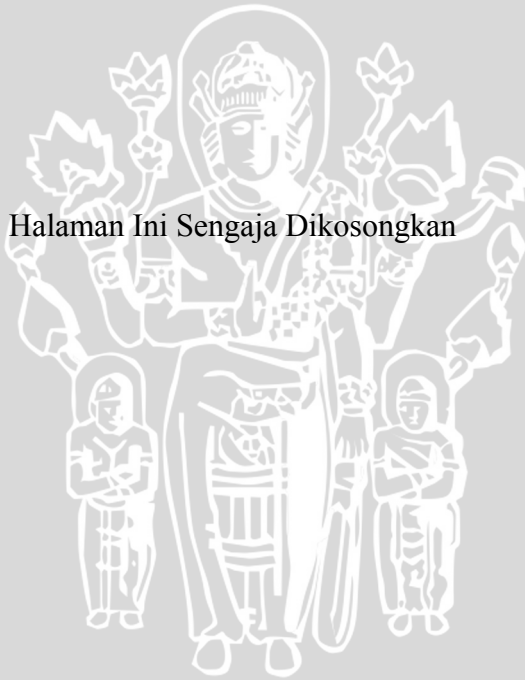
DAFTAR PUSTAKA

- Anonymous. 2007. **SRF-05 Ultrasonic Range Finder**. www.hobbyengineering.com. Diakses tanggal 26 Juli 2008
- Axelson, J. 1999. **Serial Port Complete : Programming and Circuits for RS-232 and RS-485 Links and Networks**. Lakeview Research. Madison
- Budiharto, Widodo S.Si, Mkom.2004.**Interfacing Komputer dan Mikrokontroler**. PT. Elex Media Komputindo.Jakarta
- Fahmi, M. Dimas Khoirul. 2008. **Rancang Bangun Sistem Pengukuran Terdistribusi Berbasis Jaringan Multidrop RS-485**. Jurusan Fisika Universitas Brawijaya. Malang
- Lyons, Jeff. 1999. **Distributed Monitoring And Control Modelling For A Laboratory Three Tank System**. Electrical Engineering Vanderbilt University. Tennessee
- Malik, Moh.Ibnu. 2003. **Belajar Mikrokontroler PIC 16F84**. Gaya Media. Yogyakarta
- Sahala, Stepanus. 2004. **Gelombang Ultrasonik Dan Terapannya**. UNAIR. Surabaya
- Santoso, D. R. 2005. **Development of Piezoelectric Sensor Module And System For SHM Instrumentation**. Indonesia Scientific Meeting. Nagoya-Japan
- Zarkasi, Ahmad. 2009. **Analisis Pengaturan Jarak Sensor Ultrasonik - Dengan Bahasa Pemrograman C Menggunakan MCU AT89C51**. ElectronicLab
- <http://www.hw-server.com/docs/rs485.html>,

tanggal akses : 14 April 2007

UNIVERSITAS BRAWIJAYA

Halaman Ini Sengaja Dikosongkan



LAMPIRAN

Lampiran 1 Data Pengujian Sensor Ultrasonik

Jarak (...cm)	t delay (... μ s)		rata-rata t delay (... μ s)
	naik	turun	
5	286.03	286.03	286.03
10	572.42	572.42	572.42
15	856.26	866.53	861.395
20	1118.44	1128.71	1123.575
25	1412.56	1422.83	1417.695
30	1697.63	1707.9	1702.765
35	1950.77	1960.97	1955.87
40	2304.86	2315.13	2309.995
45	2585.98	2596.25	2591.115
50	2845.16	2855.37	2850.265
55	3178.21	3188.48	3183.345
60	3447.53	3422.6	3435.065
65	3809.65	3784.72	3797.185
70	4109.77	4084.84	4097.305
75	4491.78	4466.85	4479.315
80	4794.34	4769.41	4781.875
85	5070.39	5045.37	5057.88
90	5302.42	5267.22	5284.82
95	5612.53	5577.33	5594.93
100	5907.74	5872.54	5890.14
105	6208.95	6173.75	6191.35
110	6496.17	6460.97	6478.57
115	6831.92	6796.72	6814.32
120	7119.21	7084.01	7101.61
125	7470.63	7435.43	7453.03
130	7760.51	7485.31	7622.91
135	8051.63	7776.43	7914.03
140	8348.79	8073.54	8211.165
145	8601.65	8326.56	8464.105
150	8800.59	8560.68	8680.635

155	9105.69	8915.65	9010.67
160	9408.87	9168.87	9288.87
165	9724.56	9484.98	9604.77
170	10074.12	9834.15	9954.135
175	10325.19	10085.31	10205.25
180	10629.39	10389.43	10509.41
185	10913.62	10673.55	10793.585
190	11269.75	11029.81	11149.78
195	11444.97	11314.87	11379.92
200	11730.01	11599.91	11664.96
205	12010.34	11880.29	11945.315
210	12213.06	12322.96	12268.01
215	12514.18	12624.08	12569.13
220	12777.36	12887.26	12832.31
225	13125.11	13235.01	13180.06
230	13417.53	13527.43	13472.48
235	13730.69	13840.55	13785.62
240	14023.84	14133.73	14078.785
245	14428.18	14428.18	14428.18
250	14665.51	14665.51	14665.51
255	14945.24	14945.24	14945.24
260	15470.22	16030.22	15750.22
265	15635.32	16195.32	15915.32
270	16032.43	16592.43	16312.43
275	16275.55	16835.55	16555.55
280	16567.38	17127.38	16847.38
285	16723.61	17283.61	17003.61
290	17168.66	17728.66	17448.66
295	17391.78	17951.78	17671.78
300	17534.22	18094.26	17814.24
305	18032.02	18032.02	18032.02
310	18287.84	18287.84	18287.84
315	18517.11	18517.11	18517.11
320	18709.23	18029.23	18369.23
325	19033.35	18353.35	18693.35
330	19297.41	18617.41	18957.41
335	19574.52	18914.52	19244.52
340	20033.66	19173.66	19603.66

345	20433.98	19473.98	19953.98
350	20353.24	20473.15	20413.195
355	20557.52	20677.58	20617.55
360	20883.29	21003.29	20943.29
365	21187.64	21207.64	21197.64
370	21455.69	21575.76	21515.725
375	21794.43	21804.77	21799.6
380	22114.01	22214.01	22164.01

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA

Lampiran 2 Data Pengujian Software

time sampling	time delay (... μ s)	L sensor (m)
5	14942.592	2.54024064
10	14680.175	2.49562975
15	14493.657	2.46392169
20	13978.539	2.37635163
25	13791.422	2.34454174
30	13466.952	2.28938184
35	13241.186	2.25100162
40	12664.369	2.15294273
45	11994.951	2.03914167
50	12048.834	2.04830178
55	11619.706	1.97535002
60	11025.588	1.87434996
65	10935.471	1.85903007
70	10702.353	1.81940001
75	10390.235	1.76633995
80	10104.118	1.71770006
85	9851.11	1.6746887
90	9522.882	1.61888994
95	9208.765	1.56549005
100	9279.317	1.57748389
105	8867.529	1.50747993
110	8525.412	1.44932004
115	8189.294	1.39217998
120	7879.176	1.33945992
125	7528.059	1.27977003
130	7309.041	1.24253697
135	6972.424	1.18531208
140	6787.706	1.15391002
145	6389.588	1.08622996
150	6272.471	1.06632007
155	5980.353	1.01666001
160	5620.235	0.95543995
165	5093.002	0.86581034
170	4811.21	0.8179057

175	4350.782	0.73963294
180	4209.985	0.71569745
185	3966.647	0.67432999
190	3397.437	0.57756429
195	3279.32	0.5574844
200	3001.202	0.51020434
205	2571.084	0.43708428
210	2357.967	0.40085439
215	2285.849	0.38859433
220	2040.732	0.34692444
225	1706.614	0.29012438
230	1300.466	0.22107922
235	1304.471	0.22176007
240	1346.5	0.228905
245	1620.6	0.275502
250	1904.7	0.323799
255	2187.78	0.3719226
260	2411.88	0.4100196
265	2688.18	0.4569906
270	3032.18	0.5154706
275	3329.28	0.5659776
280	3723.4	0.632978
285	4028.5	0.684845
290	4422.6	0.751842
295	4715.8	0.801686
300	4908.88	0.8345096
305	5205.48	0.8849316
310	5496.08	0.9343336
315	5792.18	0.9846706
320	6088.38	1.0350246
325	6385.48	1.0855316
330	6670.58	1.1339986
335	6858.88	1.1660096
340	7151.78	1.2158026
345	7512.9	1.277193
350	7852.1	1.334857
355	8149.2	1.385364
360	8339.28	1.4176776

365	8636.38	1.4681846
370	9031.48	1.5353516
375	9279.58	1.5775286
380	9409.78	1.5996626
385	9809.88	1.6676796
390	10291	1.74947
395	10484.1	1.782297
400	10788	1.83396
405	11078.4	1.883328
410	11340.5	1.927885
415	11667.6	1.983492
420	11960.7	2.033319
425	12553.8	2.134146
430	12848.9	2.184313
435	13142.1	2.234157
440	13341.2	2.268004
445	13685.3	2.326501
450	13699.4	2.328898
455	14003.5	2.380595
460	14310.6	2.432802
465	14601.8	2.482306
470	14697.5	2.498575



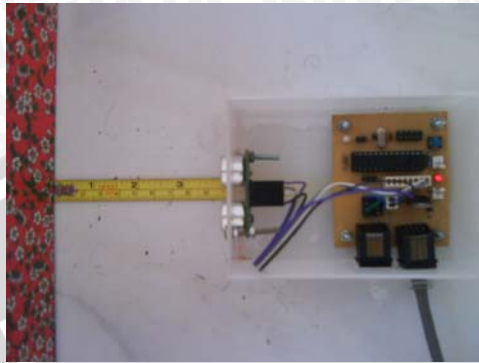
Lampiran 3
Gambar Alat



Rangkaian *slave*



Rangkaian *master*



Pengujian kalibrasi sensor



Pengujian alat

Lampiran 4

Source Code Master (Delphi 7)

```
unit Unit1;  
  
interface  
  
uses  
    Windows, Messages, SysUtils, Variants, Classes,  
    Graphics, Controls, Forms,  
    Dialogs, Grids, StdCtrls, Menus, Buttons,  
    ExtCtrls, CPort;  
  
type  
    TForm1 = class(TForm)  
        StringGrid1: TStringGrid;  
        OpenDialog1: TOpenDialog;  
        MainMenu: TMainMenu;  
        file1: TMenuItem;  
        open1: TMenuItem;  
        save1: TMenuItem;  
        SaveDialog1: TSaveDialog;  
        exit1: TMenuItem;  
        ComboBox1: TComboBox;  
        BitBtn1: TBitBtn;  
        Timer1: TTimer;  
        Label1: TLabel;  
        Label2: TLabel;  
        Label3: TLabel;  
        Button1: TButton;  
        Button2: TButton;  
        Button3: TButton;  
        Edit1: TEdit;  
        Timer2: TTimer;  
        Label4: TLabel;  
        procedure FormCreate(Sender: TObject);  
        procedure Delay(lama:cardinal);  
        procedure ClearBuffer;  
        procedure open1Click(Sender: TObject);  
        procedure save1Click(Sender: TObject);  
        procedure exit1Click(Sender: TObject);  
        procedure BitBtn1Click(Sender: TObject);  
        procedure Timer1Timer(Sender: TObject);  
    end;  
end;
```

```

procedure ComPort1RxChar(Sender: TObject;
Count: Integer);
procedure save;
procedure open;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure excellClick(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
i,t,n:integer;
Form1: TForm1;
RSBuffer:string;
ChInBuf:integer;
a,b,c:integer;
d,e,f:integer;

```

```

implementation
{$R *.dfm}

```

```

procedure TForm1.FormCreate(Sender: TObject);
begin
i:=0;
stringgrid1.Cells[0,0]:='  No';
stringgrid1.Cells[1,0]:='  Time';
stringgrid1.Cells[2,0]:='  T_delay (uS)';
stringgrid1.Cells[3,0]:='  L_SRF05 (cm)';
stringgrid1.Cells[4,0]:='  H (cm)';
BitBtn1.Enabled:=true;
end;

```

```

procedure TForm1.open1Click(Sender: TObject);
var
datanya:textfile;
kanal0,kanal3:string[6];
kanal4,kanal5:string[10];
kanal1,kanal2,kanal6:string[11];
jumdat:string[6];
t_sampling:string[6];

```

```

sigdata:integer;
k:integer;
begin
if opendialog1.Execute then
begin
assignFile(datanya,opendialog1.FileName);
reset(datanya);
readln(datanya,t_samplings);
readln(datanya,jumdat);
sigdata:=strtoint(jumdat);
stringgrid1.RowCount:=sigdata+1;
for k:=1 to (sigdata+1) do
begin
readln(datanya,kanal0);
stringgrid1.Cells[0,k]:=kanal0;
readln(datanya,kanal1);
stringgrid1.Cells[1,k]:=kanal1;
readln(datanya,kanal2);
stringgrid1.Cells[2,k]:=kanal2;
readln(datanya,kanal3);
stringgrid1.Cells[3,k]:=kanal3;
readln(datanya,kanal4);
stringgrid1.Cells[4,k]:=kanal4;
end;
end;
end;

procedure TForm1.Delay(lama:cardinal);
var
ref:cardinal;
begin
ref:=GetTickCount;
repeat
Application.ProcessMessages;
until
((GetTickCount-Ref)>=lama);
end;

procedure TForm1.sav1Click(Sender: TObject);
var
datanya:TextFile;
jumdat,bar:integer;
begin
if savedialog1.Execute then
begin

```

```
assignfile(datanya,savedialog1.FileName);
rewrite(datanya);
for bar:=1 to (jumdat+1) do
begin
writeln(datanya,copy(form1.StringGrid1.Cells[0,bar]
,0,6));
writeln(datanya,copy(form1.StringGrid1.Cells[1,bar]
,0,11));
writeln(datanya,copy(form1.StringGrid1.Cells[2,bar]
,0,11));
writeln(datanya,copy(form1.StringGrid1.Cells[3,bar]
,0,6));
writeln(datanya,copy(form1.StringGrid1.Cells[4,bar]
,0,10));
end;
closefile(datanya);
end;
end;
```

```
procedure TForm1.exit1Click(Sender: TObject);
begin
beep;
if messagedlg('Anda ingin
menyimpan',mtconfirmation,[mbyes,mbno],0)=mryes
then
begin
application.Terminate;
end;
if messagedlg('Anda yakin
Keluar',mtconfirmation,[mbyes,mbcancel],0)=mryes
then application.Terminate;
end;
```

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
BitBtn1.Enabled:=false;
stringgrid1.RowCount:=(10);
timer1.Interval:=1000*60;
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
label2.Caption:=Timetostr(time);
label3.Caption:=datetostr(date);
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  delay(5);
  delay(5);
  timer1.Enabled:=true;
  button1.Enabled:=false;
  button2.Enabled:=true;
  BitBtn1.Enabled:=false;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  timer1.Enabled:=false;
  BitBtn1.Enabled:=false;
  button1.Enabled:=true;
  button2.Enabled:=false;
end;
```

```
procedure TForm1.ClearBuffer;
begin
  RSBuffer:='';
  ChInBuf:=0;
end;
```

```
procedure TForm1.ComPort1RxChar(Sender: TObject;
Count: Integer);
var
  hasil:string;
begin
  RSBuffer:=RSbuffer+hasil;
  ChInBuf:=length(RSBuffer);
  a:=ord(RSBuffer[1]);
  b:=ord(RSBuffer[2]);
  c:=ord(RSBuffer[3]);
  d:=ord(RSBuffer[4]);
  e:=ord(RSBuffer[5]);
  f:=ord(RSBuffer[6]);
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
var
  kolom,baris: integer;
  j:integer;
begin
```

```
i:=0;
j:=strtoint(edit1.Text)+1;
for kolom:=0 to 6 do
begin
    for baris:=1 to j do
        StringGrid1.Cells[kolom,baris]:= '';
    end;
stringgrid1.RowCount:=10;
edit1.Text:='0';
BitBtn1.Enabled:=true;
button2.Enabled:=false;
end;

procedure TForm1.save;
var
datanya:TextFile;
jumdat,bar:integer;
begin
jumdat:=strtoint(edit1.Text);
if savedialog1.Execute then
begin
    assignfile(datanya,savedialog1.FileName);
    rewrite(datanya);
    writeln(datanya,copy (combobox1.Text,1,6));
    writeln(datanya,copy (edit1.Text,1,6));
    for bar:=1 to (jumdat+1) do
        begin
writeln(datanya,copy(form1.StringGrid1.Cells[0,bar]
,0,6));
writeln(datanya,copy(form1.StringGrid1.Cells[1,bar]
,0,11));
writeln(datanya,copy(form1.StringGrid1.Cells[2,bar]
,0,11));
writeln(datanya,copy(form1.StringGrid1.Cells[3,bar]
,0,6));
writeln(datanya,copy(form1.StringGrid1.Cells[4,bar]
,0,10));
writeln(datanya,copy(form1.StringGrid1.Cells[5,bar]
,0,10));
writeln(datanya,copy(form1.StringGrid1.Cells[6,bar]
,0,20));
        end;
    closefile(datanya);
end;
end;
```



```
procedure TForm1.Open;
var
  datanya:textfile;
  kanal0,kanal3:string[6];
  kanal4,kanal5:string[10];
  kanal1,kanal2,kanal6:string[11];
  jumdat:string[6];
  t_sampling:string[6];
  sigdata:integer;
  k:integer;
begin
  if opendialog1.Execute then
    begin
      assignFile(datanya,opendialog1.FileName);
      reset(datanya);
      readln(datanya,t_sampling);
      combobox1.Text:=t_sampling;
      readln(datanya,jumdat);
      edit1.Text:=jumdat;
      sigdata:=strtoint(jumdat);
      stringgrid1.RowCount:=sigdata+1;
      for k:=1 to (sigdata+1) do
        begin
          readln(datanya,kanal0);
          stringgrid1.Cells[0,k]:=kanal0;
          readln(datanya,kanal1);
          stringgrid1.Cells[1,k]:=kanal1;
          readln(datanya,kanal2);
          stringgrid1.Cells[2,k]:=kanal2;
          readln(datanya,kanal3);
          stringgrid1.Cells[3,k]:=kanal3;
          readln(datanya,kanal4);
          stringgrid1.Cells[4,k]:=kanal4;
          readln(datanya,kanal5);
          stringgrid1.Cells[5,k]:=kanal5;
        end;
      end;
    end;
end;

procedure TForm1.excel1Click(Sender: TObject);
var
  fa,g:variant;
  bar,F:integer;
begin
```

```

fa:='Excel.Application';
fa.visible:=true;
fa.workbooks.add;
fa.workbooks[1].worksheets[1].name:='File';
g:=fa.workbooks[1].worksheets['File'];
g.cells[1,1]:='No.';
g.cells[1,2]:='Time';
g.cells[1,3]:='T_Delay (uS)';
g.cells[1,4]:='L_SRF05(cm)';
g.cells[1,5]:='H(cm)';
F:=strtoint(edit1.Text);
for bar:=2 to F+1 do
begin
    g.cells[bar,1]:=bar-1;
    g.cells[bar,2]:=stringgrid1.Cells[1,bar-1];
    g.cells[bar,3]:=stringgrid1.Cells[2,bar-1];
    g.cells[bar,4]:=stringgrid1.Cells[3,bar-1];
    g.cells[bar,5]:=stringgrid1.Cells[4,bar-1];
    g.cells[bar,6]:=stringgrid1.Cells[5,bar-1];
end;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
ClearBuffer;
delay(10);
delay(10);
    repeat
        application.ProcessMessages;
    until
        ChInBuf>=1;
    if a=254 then
        begin
            timer1.Enabled:=false;
            messagedlg('Reset Communication, Data not
Ready',mtconfirmation,[mbok],0);
            application.Terminate;
        end;
    if a=255 then
        begin
            timer1.Enabled:=false;
            messagedlg('Reset Communication, Data not
Ready',mtconfirmation,[mbok],0);
            application.Terminate;
        end;
end;

```

```
repeat
application.ProcessMessages;
until
  ChInBuf>=6;
begin
d:=a;
e:=b;
f:=c;
t:=strtoint(combobox1.Text);
end;
inc(i);
stringgrid1.Cells[1,i]:=timetostr (time);
stringgrid1.Cells[2,i]:=floattostr (d);
stringgrid1.Cells[3,i]:=floattostr (e);
stringgrid1.Cells[4,i]:=floattostr (f);
if f=0 then
begin
stringgrid1.Cells[4,i]:='40';
end;
if f=1 then
begin
stringgrid1.Cells[4,i]:='50';
end;
if f=2 then
begin
stringgrid1.Cells[4,i]:='60';
end;
if f=3 then
begin
stringgrid1.Cells[4,i]:='70';
end;
if f=4 then
begin
stringgrid1.Cells[4,i]:='80';
end;
if f=5 then
begin
stringgrid1.Cells[4,i]:='90';
end;
end;
end.
```

Lampiran 5

Source Code Mikrokontroler (Assembly)

```
#include "p16F876.inc"
;SOFTWARE MEASURE WATER LEVEL BY ULTRASONIC SENSOR
;25 MEI 2009
;Angga S P
```

```
INDF EQU 00H
STATUS EQU 03H
RP0 EQU 05H
RP1 EQU 06H
ADRESL EQU 9EH
ADRESH EQU 1EH
TXREG EQU 19H
TXSTA EQU 98H
TRISA EQU 85H
PORTA EQU 05H
TRISB EQU 86H
PORTB EQU 06H
TRISC EQU 87H
PORTC EQU 07H
FSR EQU 04H
CNTAD EQU 20H
AIR EQU 21H ;register hasil konversi delay
TDELAY EQU 22H
JARAK EQU 26H ;
```

```
ORG 0X0000
MAIN
```

```
CALL INIT
CLRF TDELAY
CLRF JARAK
BCF PORTC,2
BCF PORTC,5 ; RECEIVE MODE
BSF PORTB,7
BCF PORTB,6
CALL TRIGGER
BSF PORTC,2
BSF PORTC,5 ; TRANSMIT MODE
```

```
CALL COMMUNICATION_RESET
MOVLW B'00000011' ;TDELAY
MOVWF COMMAND
```

```

CALL WRITE_SRF
      BTFSC STATUS,C
      GOTO ERROR
CALL WAITFORDATA
      XORLW 1 ; 1=TIMEOUT
      BTFSC STATUS,Z
      GOTO ERRORTIMEOUT ;W=1
CALL READ
      MOVFW UKUR
      MOVWF TDELAY
CALL READ
      MOVFW UKUR
      MOVWF JARAK
CALL DELAY_10 uS
CALL DELAY
CALL DELAY

      MOVFW TDELAY
      CALL SEND
      CALL DELAY
      CALL DELAY

      MOVFW JARAK
      CALL SEND
      CALL DELAY
      CALL DELAY

      MOVFW PORTA
      CALL SEND
      CALL DELAY
      GOTO START
ERROR
      MOVLW D'254'
      GOTO DOERROR
ERRORTIMEOUT
      MOVLW D'255'
      GOTO DOERRORNOACK
DOERROR
      CALL SEND
      CALL DELAY
      CALL DELAY
      GOTO START
;
INIT: ;-----SERIAL-----

```

```

CLRF    STATUS
BSF     STATUS,RP0    ;BANK1
MOVLW  B'10000000'
MOVWF   TRISC
BCF     STATUS,RP0
MOVLW  0X00
MOVWF   T1CON
MOVLW  0X90
MOVWF   RCSTA
BSF     STATUS,RP0    ;BANK1
MOVLW  0X24           ;BRGH=1
MOVWF   TXSTA
MOVLW  D'119'        ;9.600bps AT 18.432 MHZ
MOVWF   SPBRG

```

```

;-----PORT-----
BCF     STATUS, RP0
CLRF    PORTA
CLRF    PORTB
BSF     STATUS,RP0    ;BANK1
MOVLW  0X01
MOVWF   TRISB
MOVLW  0X06
MOVLW  B'11000000'
MOVWF   OPTION_REG
BCF     STATUS,RP0    ;BANK0
;-----INTERUPS-----
BSF     INTCON,7
BSF     INTCON,4
BCF     INTCON,1
BCF     STATUS,RP0    ;BANK0
CLRF    PIR1          ;CLEAR ALL INTERRUPT
RETURN

```

```

;PROSES SRF05
TRIGGER:

```

```

BSF PORTB,7 ;
NOP
BSF PORTB,6 ;
NOP
CALL TRIGGER
CALL DELAY_10US
RETURN

```

```

COMMUNICATION_RESET:

```

```
BCF STATUS,RP0
CLRF PORTB
BSF STATUS,RP0
MOVLW B'00000001'
MOVWF TRISB
BCF STATUS,RP0 ;
BSF PORTB,7;DATA=1
BCF PORTB,6;
MOVLW D'9'
RETURN
```

; writes a byte to the sensor

WRITE:

```
MOVLW D'8'
MOVWF CONT
```

WRITE1

```
RLF COMMAND,F
BTSS STATUS,C
BCF PORTB,7;DATA=0
BTFS STATUS,C
BSF PORTB,7;DATA=1
BSF PORTB,6;SCK=1
NOP
NOP
NOP
BCF PORTB,6;SCK=0
DECFSZ CONT,F
GOTO WRITE1
```

```
BCF STATUS,RP0
CLRF PORTB
BSF STATUS,RP0
MOVLW B'10000001'
MOVWF TRISB
BCF STATUS,RP0 ;D
```

ECHO

```
BSF PORTB,6
BCF STATUS,C
BTFS PORTB,7
BSF STATUS,C
BCF PORTB,6 ; WAIT HIGH LOGIC
CALL DELAY_30MS
RETURN ;SIGNAL ERROR CARRY=1
```

```

BCF PORTB,6 ; TDELAY RECEIVE

WAITFORDATA: ; TIMEOUT OF 30 MS
MOV LW 4 ; =4 FOR XTAL=20MHZ
MOVWF COUNT1

DDC1
CLRF COUNT2

DDC2
CLRF COUNT3

DDC3
BTFSZ PORTB,7
RETLW 0
DECFSZ COUNT3
GOTO DDC3
DECFSZ COUNT2
GOTO DDC2
DECFSZ COUNT1
GOTO DDC1
RETLW 1 ; TIMEOUT

READ:
MOV LW D'8'
MOVWF CONT

READ1
BSF PORTB,6,SCK=1
BCF STATUS,C
BTFSZ PORTB,7
BSF STATUS,C
RLF UKUR
BCF PORTB,6,SCK=0
DECFSZ CONT
GOTO READ1
BCF STATUS,RP0
CLRF PORTB
BSF STATUS,RP0
MOV LW B'00000001'
MOVWF TRISB
BCF STATUS,RP0 ;SRF WRITE
BSF PORTB,7;DATA=1
RETURN

SEND:
BCF STATUS,RP0
BTFSZ PIR1,TXIF

```



```
GOTO $-1
MOVWF TXREG
RETURN
```

TRIGGER:

```
BTSS PIR1, RCIF
GOTO $-1
MOVWF RCREG
BCF STATUS, Z
XORLW 'M'
BTSS STATUS, Z
GOTO START ; IF NOT 'M' IF RESTART
CALL DELAY_100MS
RETURN
```

DELAY_1MS:

```
BCF STATUS, RP0
BCF PIR1, TMR1IF
MOVLW 0XEE
MOVWF TMR1H
MOVLW 0X32
MOVWF TMR1L
BSF T1CON, TMR1ON ; START TO COUNT
BTSS PIR1, TMR1IF
GOTO $-1
RETURN
```

DELAY_10MS:

```
MOVLW D'10'
MOVWF COUNT1
LOOP10 BCF STATUS, RP0
BCF PIR1, TMR1IF
MOVLW 0XEE
MOVWF TMR1H
MOVLW 0X32
MOVWF TMR1L
BSF T1CON, TMR1ON ; START TO COUNT
BTSS PIR1, TMR1IF
GOTO $-1
DECFSZ COUNT1, F
GOTO LOOP10
RETURN
```

```
DELAY_100MS:
    MOVLW D'10'
    MOVWF COUNT2
LOOP25 CALL DELAY_10MS
    DECFSZ COUNT2,F
    GOTO LOOP25
    RETURN
```

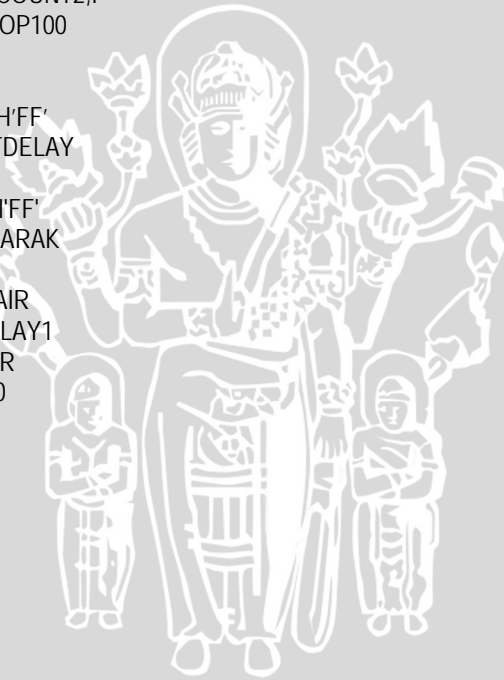
```
DELAY_1000MS
    MOVLW D'100'
    MOVWF COUNT2
LOOP100
    CALL DELAY_10MS
    DECFSZ COUNT2,F
    GOTO LOOP100
    RETURN
```

```
DELAY:
    MOVLW H'FF'
    MOVWF TDELAY
```

```
DELAY0
    MOVLW H'FF'
    MOVWF JARAK
```

```
DELAY1
    DECFSZ AIR
    GOTO DELAY1
    DECFSZ AIR
    GOTO DELAY0
    RETURN
```

```
END.
```



Lampiran 6

Datasheet Sonic Range Finder SRF05

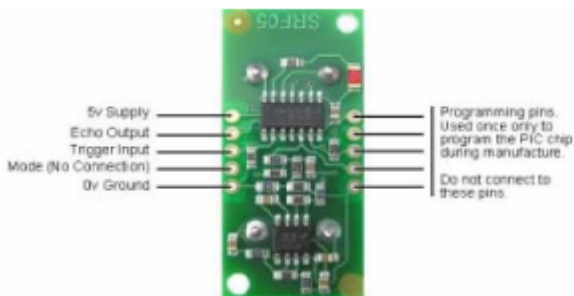
ULTRASONIC RANGE SENSOR

Specification:

The ultrasonic range sensor detects objects in its path and can be used to calculate the range to the object. It is sensitive enough to detect a 3cm diameter broom handle at a distance of over 3m.

Voltage	- 5v
Current	- 30mA Typ. 50mA Max.
Frequency	- 40KHz
Max Range	- 4 m
Min Range	- 3 cm
Sensitivity	- Detect 3cm diameter broom handle at > 3 m
Input Trigger	- 10µs Min. TTL level pulse
Echo Pulse	- Positive TTL level signal, width proportional to range.
Small Size	- 43mm x 20mm x 17mm height

Electrical connection (Mode 1 - separate trigger / echo pins):



Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)

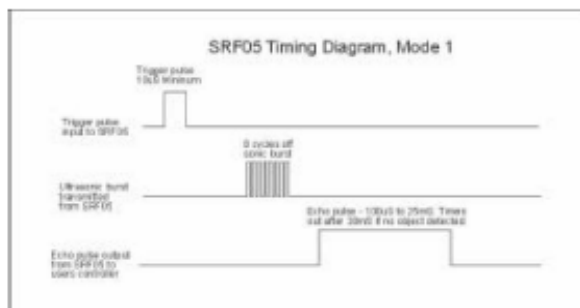
The SRF005 ultrasonic range finder has 5 connections pins. The power supply is connected to the 5V and 0V ground connections on the SRF005.

Important - Note that the 'Mode' (hole 4) connection **MUST NOT** be connected for correct operation in this mode. This is a different layout to the discontinued SRF004 module, where the mode pin was connected to 0V when used with the PICAXE system. If using the SRF005 module on a PCB designed originally for the SRF004 (e.g. the Micro-robot) cut off the mode pin from the 5 pin connector, so that no connection is made on this pin when inserted onto the main PCB.

Take care not to overheat, and therefore damage, the solder connection pads whilst making connections.

The SRF005 Trigger Input is connected to a PICAXE output pin.
The SRF005 Echo Output is connected to a PICAXE input pin.

Operation with the PICAXE microcontroller:



The following program gives an example of how to use the SRF005 module with a PICAXE-18 microcontroller. Output 3 is used to trigger the SRF005 module via a 'pulsout' command. The SRF005 module then sends out the sonic burst, and sets the Echo Output connection high for the time it takes the sonic burst to be returned. Therefore the PICAXE Input (input 6) is used to receive and time this echo pulse via a 'pulsin' command.

The length of the echo pulse is then divided by 5.8 to give a value in cm, and displayed on the computer screen via the 'debug' command. Note that a word variable, w1, is used for the echo timing, as the echo pulse will be a value greater than 255 (maximum value of a byte variable). Word variables are made up of two byte variables and so have a maximum value of 65535 (in this case w1 is made up of b2 and b3, so these two byte variables must not be used anywhere else in the program).

Sample PICAXE Program:

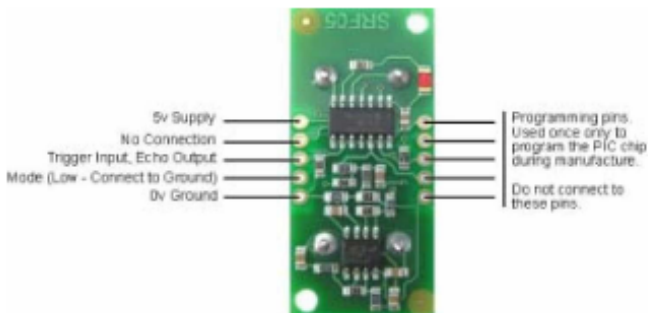
```

symbol trig = 3      ' Define output pin for Trigger pulse
symbol echo = 6     ' Define input pin for Echo pulse
symbol range = w1   ' 16 bit word variable for range

main:
  pulsetrig trig, 2 ' produce 20µs trigger pulse (must be minimum of 10µs)
  pulsinecho echo, 1, range ' measures the range in 10µs steps
  pause 10          ' recharge period after ranging completes
  ' now convert range to cm (divide by 5.8) or inches (divide by 14.9)
  ' as picaxe cannot use 5.8, multiply by 10 then divide by 58 instead
  let range = range * 10 / 58 ' multiply by 10 then divide by 58
  debug range       ' display range via debug command
  goto main        ' and around forever
  
```

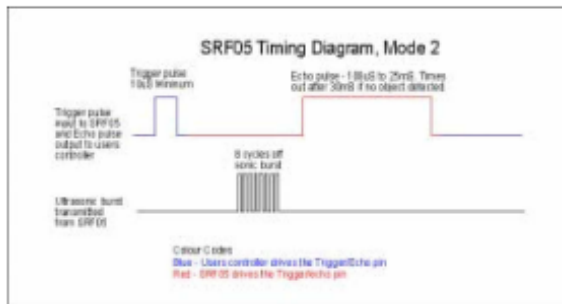
Special Mode 2 - For use with PICAXE-08/08M microcontroller:

As the PICAXE-08 and 08M have limited input/output pins, the SRF05 has a special mode for use with these chips. In this mode a single pin is used for both input and output (not possible with 18 and 28 pin PICAXE chips).



Connections for single pin Trigger/Echo Mode

This PICAXE pin used can be pin1, pin2 or pin4.



The following program gives an example of how to use the SRF05 module with a PICAXE-08 microcontroller.

Sample PICAXE Program:

```

symbol trig = 1          ' Define output pin for Trigger pulse

main:
  pulseout trig, 2       ' produce 20µs trigger pulse (must be minima of 10µs)
  pulsein trig, 1, range ' measures the range in 10µs steps
  ' now convert range to cm (divide by 5.8) or inches (divide by 14.8)
  ' as picaxe cannot use 5.8, multiply by 10 then divide by 42 instead
  let range = range * 10 / 58 ' multiply by 10 then divide by 42
  debug range             ' display range via debug command
  pause 50               ' short delay
  goto main              ' and around forever
  
```

Calculating the Distance

The SRF05 Timing diagrams are shown above for each mode. You only need to supply a short 10uS pulse to the trigger input to start the ranging. The SRF05 will send out an 8 cycle burst of ultrasound at 40kHz and raise its echo line high (or trigger line in mode 2). It then listens for an echo, and as soon as it detects one it lowers the echo line again. The echo line is therefore a pulse whose width is proportional to the distance to the object. By timing the pulse it is possible to calculate the range in inches/centimeters or anything else. If nothing is detected then the SRF05 will lower its echo line anyway after about 30mS.

The SRF04 provides an echo pulse proportional to distance. If the width of the pulse is measured in uS, then dividing by 58 will give you the distance in cm, or dividing by 148 will give the distance in inches. $uS/58=cm$ or $uS/148=inches$.

The SRF05 can be triggered as fast as every 50mS, or 20 times each second. You should wait 50ms before the next trigger, even if the SRF05 detects a close object and the echo pulse is shorter. This is to ensure the ultrasonic "beep" has faded away and will not cause a false echo on the next ranging.

The other set of 5 pins

The 5 pins marked "programming pins" are used once only during manufacture to program the Flash memory on the PIC16F630 chip. The PIC16F630's programming pins are also used for other functions on the SRF05, so make sure you don't connect anything to these pins, or you will disrupt the modules operation.

Changing beam pattern and beam width

You can't! This is a question which crops up regularly, however there is no easy way to reduce or change the beam width that I'm aware of. The beam pattern of the SRF05 is conical with the width of the beam being a function of the surface area of the transducers and is fixed. The beam pattern of the transducers used on the SRF05, taken from the manufacturers data sheet, is shown below.

