

**Rancang Bangun Alat Monitoring Ketinggian Cairan Infus  
Menggunakan *Arduino UNO***

**SKRIPSI**

**OLEH:**

**CANON LEE HAMONANGAN MARPAUNG**

**15.812.0037**



**FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2018**

**Rancang Bangun Alat Monitoring Ketinggian Cairan Infus  
Menggunakan *Arduino UNO***

**SKRIPSI**

**OLEH:**

**CANON LEE HAMONANGAN MARPAUNG  
15.812.0037**

Skripsi adalah Salah Satu Syarat untuk Mendapatkan

Gelar Sarjana di Fakultas Teknik

Universitas Medan Area

**FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA**

**MEDAN**

**2018**

## **Lembar Pengesahan**

**Judul Skripsi : Rancang Bangun Alat Monitoring  
Ketinggian Cairan Infus Menggunakan  
*Arduino UNO***

**Nama : Canon Lee Hamonangan Marpaung**

**NPM : 15.812.0037**

**Fakultas : Teknik Elektro**

**Disetujui Oleh  
Komisi Pembimbing**

Prof. Dr. Dadan Ramdan, M.Sc,M.Eng  
Pembimbing 1

Moranian Mungkin, ST.Msi  
Pembimbing 2

Prof. Dr. Dadan Ramdan, M.Sc,M.Eng  
DEKAN

Syarifah Muthia Putri, ST.MT  
Ketua Jurusan

## **LEMBAR PERNYATAAN**

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri dan bantuan orang lain. Adapun bagian - bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi – sanksi lainnya dengan peraturan yang berlaku, apabila dikemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 12 Januari 2018

Canon Lee Hamonangan Marpaung  
15.812.0037

## **ABSTRAK**

Untuk memudahkan pemantauan keberadaan cairan infus pada tabung oleh perawat dari jarak jauh maka diperlukan suatu alat untuk memonitoring ketinggian cairan infus tersebut. Alat monitor ketinggian cairan infus menggunakan *Arduino Uno* sebagai pengendali. Alat ini dilengkapi dengan sensor ketinggian HY-SRF05 sebagai ultrasonik yang nilainya ditunjukkan pada *LCD* 2x16. Kemudian indikator jika terisi penuh, setengah dan kosong menggunakan *LED* dan *Buzzer* untuk peringatan. Alat ini disuplay dengan 7-12 Volt DC. Hasil penyuplaian diperoleh kesalahan alat sebesar  $\pm 4,41$  %

**Kata Kunci:** *Infus, Arduino Uno, Ultrasonik HY-SRF05, LED, LCD, Buzzer.*

## ***Abstract***

*To facilitate the monitoring of the presence of infusion fluid in the tube by the nurse from a distance it is necessary a tool to monitor the height of the infusion fluid. Infusion fluid level monitor using Arduino Uno as controller. This tool is equipped with a HY-SRF05 height sensor as ultrasonic whose value is shown on 2x16 LCD. Then the indicator if fully charged, half and blank uses LED and Buzzer for warning. This tool is supplied with 7-12 Volts DC. The supply results obtained by the tool error of  $\pm 4,41$  %*

***Keywords: Infusion, Arduino Uno, Ultrasonic HY-SRF05, LED, LCD, Buzzer.***

## **RIWAYAT HIDUP**

Penulis bernama Canon Lee Hamonangan Marpaung dilahirkan pada tanggal 29 Mei 1994 di Medan, Anak dari Pasangan Bapak Ir. Jan Slamet dan Ibu Dra. Lindaria Lumbantoruan. Pada Tahun 2006 lulus dari SD Free Methodist – 2 Medan, Tahun 2009 Lulus dari SMP Teladan Sumatera Utara, Tahun 2012 Lulus dari SMK Negeri – 5 Medan, dan Tahun 2015 Lulus dari Politeknik Negeri Medan dengan gelar Amd. Pada Tahun 2015 Penulis masuk di Universitas Medan Area (UMA) sampai tahun 2018 mengantarkan penulis untuk mendapatkan gelar Sarjana Teknik.

Demikian Riwayat hidup penulis untuk sekedar diketahui.

Terima Kasih

Penulis

## KATA PENGANTAR

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Kuasa atas berkat, rahmat dan karunia-Nya sehingga penulis diberikan kesehatan, kekuatan, pengetahuan dan kesempatan sehingga dapat menyelesaikan Skripsi ini dengan baik dan tepat waktu.

Tema yang dipilih dalam penelitian ini adalah “RANCANGAN BANGUN ALAT MONITORING KETINGGIAN CAIRAN INFUS MENGGUNAKAN *ARDUINO UNO*“. Skripsi ini disusun guna menyelesaikan Program Pendidikan Strata 1 Program Studi Teknik Elektro Universitas Medan Area.

Dalam penyelesaian penulisan Skripsi ini penulis banyak mendapatkan bantuan, baik moral maupun materil dari berbagai pihak, dan pada kesempatan ini penulis mengucapkan terimakasih kepada:

1. Orang tua penulis yang selalu memberi do'a dan dukungan secara moril maupun material.
2. Bapak Prof. Dr. H. A. Ya'kub Matondang, MA, selaku Rektor Universitas Medan Area.
3. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc, selaku Dekan Fakultas Teknik, dan sekaligus dosen pembimbing untuk Skripsi ini, yang telah banyak meluangkan waktu, tenaga dan pikiran dalam penyusunan Skripsi ini hingga selesai.
3. Bapak Moranian Mungkin, ST, MSi, selaku dosen pembimbing untuk Skripsi ini, yang telah memberikan saran dan kritik yang membangun dalam penyusunan Skripsi sampai selesai.
4. Ibu Syarifah Muthia Putri, ST, MT, selaku Ketua Jurusan Teknik Elektro.



5. Seluruh staff pengajar Universitas Medan Area khususnya Program Studi Teknik Elektro.
6. Maria Selvina Simarmata yang tercinta telah senantiasa memberi dukungan, semangat dan motivasi sehingga penulis mampu berjuang kembali untuk menyelesaikan Skripsi ini.
7. Rekan-rekan kelas terkhususnya buat teknik elektro angkatan 2015 yang telah banyak memberikan kenangan manis dan persahabatan yang baik.
8. Semua rekan-rekan yang tidak disebutkan namanya yang telah membantu dalam doa dan memberi semangat kepada penulis dalam menyelesaikan pembuatan Skripsi ini.
9. Kepada semua pihak yang membantu penulis untuk menyelesaikan Skripsi ini.

Penulis menyadari bahwa masih ada kekurangan dalam pembuatan Skripsi ini, untuk itu dengan segala kerendahan hati penulis menerima kritik dan saran yang membangun demi kesempurnaan Skripsi ini nantinya. Semoga Skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan maupun bagi dunia usaha dan pemerintahan.

Akhirnya penulis kembali mengucapkan terimakasih kepada semua pihak yang telah membantu penulis dalam menyelesaikan Skripsi ini. Sehingga dapat bermanfaat bagi siapapun yang membacanya.

Medan, 12 Januari 2018

Hormat Penulis

Penulis

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERSETUJUAN.....	ii
DAFTAR ISI.....	viii
DAFTAR TABLE .....	xi
DAFTAR GAMBAR .....	xii
DAFTAR LAMPIRAN .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah .....	4
1.5 Manfaat Pembuatan Alat .....	4
1.6 Sistematika Pembahasan .....	5
BAB II TEORI PENUNJANG .....	6
2.1 <i>Arduino</i> .....	6
2.2 <i>Arsitektur Microcontroller ATmega</i> .....	9
2.3 Definisi Program .....	12
2.4 <i>Software Arduino IDE</i> .....	13
2.5 Sensor Ultrasonik .....	14
2.6 <i>LCD</i> .....	17

2.7	<i>LED</i> .....	19
2.8	<i>Alarm/Buzzer</i> .....	20
2.9	Catu Daya .....	21
2.10	Cairan Intravena (Infus) .....	21
BAB III METODOLOGI PENELITIAN .....		24
3.1	Tempat dan Waktu Penelitian .....	24
3.1.1	Tempat Penelitian .....	24
3.1.2	Waktu Penelitian .....	24
3.2	Metode Penelitian .....	24
3.2.1	Alat dan Bahan .....	26
3.2.2	Rancangan Struktural .....	26
3.2.3	Rancangan Sistem Elektrikal .....	31
	A. Sistem <i>AC-DC</i> Adaptor.....	32
	B. Sistem Sensor Ultrasonik HY-SRF-05.....	33
	C. Sistem <i>Indikator LED</i> .....	34
	D. Sistem Penampilan Data ( <i>LCD2x16</i> ). .....	36
	E. Sistem <i>Buzzer</i> . .....	37
	F. Sistem <i>Minimum Arduino Uno</i> . .....	38
	G. Sistem Secara Keseluruhan.....	39
3.3	<i>Flowchart</i> Program Monitoring Infus Menggunakan Arduino ...	40
3.4	<i>Flowchart</i> Sistem Kerja Alat.....	41

BAB IV PENGUJIAN DAN ANALISA .....	42
4.1 Hasil Pembuatan Alat .....	42
4.2 Pengujian Alat dan Analisa .....	43
4.2.1 Pengujian Modul <i>Arduino Uno</i> .....	44
4.2.2 Pengujian Rangkaian Sensor Ultrasonik dengan <i>Arduino</i> .	46
4.2.3 Pengujian <i>LCD 16x2</i> .....	49
4.2.4 Pengujian Indikator <i>LED</i> .....	51
4.2.5 Pengujian <i>Buzzer</i> .....	52
4.2.6 Pengujian Alat Secara Keseluruhan .....	54
4.2.7 Kurva Hubungan Antara Volume dengan Jarak Sensor ..	58
4.2.8 Perhitungan Volume Ketinggian Cairan dalam Program ..	59
BAB V KESIMPULAN DAN SARAN.....	60
5.1 Kesimpulan .....	60
5.2 Saran .....	60
DAFTAR PUSTAKA .....	61
LAMPIRAN .....	62

## DAFTAR TABEL

Tabel 2.1 Indeks <i>Board Arduino Uno</i> .....	9
Tabel 3.1 Penetapan Komponen Bahan .....	26
Tabel 4.1 Hasil Pengujian <i>LED</i> .....	51
Tabel 4.2 Hasil Pengujian <i>Buzzer</i> .....	53
Tabel 4.3 Hasil pengujian Secara Keseluruhan .....	57

## DAFTAR GAMBAR

Gambar 2.1	<i>Microcontroller</i> ATmega 328 .....	6
Gambar 2.2	<i>Arduino Uno</i> dan <i>Lily Pad</i> .....	7
Gambar 2.3	<i>Led-led</i> di <i>Arduino</i> .....	8
Gambar 2.4	Blok diagram <i>microcontroller</i> .....	10
	(a). ATmega 328 .....	10
	(b). ATmega 1280 .....	11
Gambar 2.5	Tampilan <i>Arduino IDE</i> Versi 1.6.4.....	14
Gambar 2.6	Cara Kerja Sensor Ultrasonik .....	15
Gambar 2.7	Sensor Jarak Ultrasonik HY-SRF05 .....	16
Gambar 2.8	Blok Diagram <i>LCD</i> .....	18
Gambar 2.9	Konfigurasi <i>pin</i> pada <i>LCD</i> .....	18
Gambar 2.10	Bentuk Fisik <i>LED</i> dan Simbol .....	19
Gambar 2.11	<i>LEDarray</i> .....	20
Gambar 2.12	<i>Buzzer</i> .....	20
Gambar 2.13	Regulator Tegangan .....	21
Gambar 2.14	Botol Infus .....	23
Gambar 3.1	<i>Flowchart</i> Kerangka Berfikir .....	25
Gambar 3.2	Sketsa Dudukan Seluruh Sistem.....	28
Gambar 3.3	Desain dan Dimensi Dudukan Secara Elektrikal .....	29
Gambar 3.4	Sketsa Penempatan Sensor Ultrasonik dengan Botol Infus .....	30
Gambar 3.5	Desain Tata Letak Sistem .....	31
Gambar 3.6	<i>AC-DC</i> Adaptor .....	32
Gambar 3.7	Skema Rangkaian <i>AC-DC</i> Adaptor .....	32
Gambar 3.8	Pola Penguinstalasian Sensor Ultrasonik HY-SRF05 .....	33
Gambar 3.9	Rangkaian Sistem Indikator <i>LED</i> .....	35

Gambar 3.10	Pola Instalasi <i>LCD</i> 2x16 pada <i>Arduino Uno</i> .....	36
Gambar 3.11	Pola Instalasi <i>Buzzer</i> pada <i>Arduino Uno</i> .....	37
Gambar 3.12	Sistem Minimum <i>Arduino Uno</i> .....	38
Gambar 3.13	Skema Rangkaian Seluruh Sistem.....	39
Gambar 3.14	<i>Flowchart</i> Program Monitoring Infus .....	40
Gambar 3.15	<i>Flowchart</i> Sistem Kerja Alat .....	41
Gambar 4.1	Sistem Monitoring Volume Cairan Infus .....	42
Gambar 4.2	Rangkaian Pengujian Model <i>Arduino</i> dengan Indikator <i>LED</i> ..	44
Gambar 4.3	Listing Program Pengujian Lampu <i>LED</i> .....	45
Gambar 4.4	Listing Program Pengujian Lampu <i>LED</i> .....	46
Gambar 4.5	Rangkaian Uji Coba Sensor Ultrasonik dengan <i>Arduino Uno</i> .	46
Gambar 4.6	Rangkaian nyata Sensor Ultrasonik dengan <i>Arduino Uno</i> .....	48
Gambar 4.7	Tampilan Hasil Pengukuran Sensor Ultrasonik .....	49
Gambar 4.8	Pengujian Rangkaian <i>LCD</i> 16x2 <i>Character</i> .....	50
Gambar 4.9	Listing Program Pengujian Rangkaian <i>LCD</i> 16x2 <i>Character</i> ..	51
Gambar 4.10	Rangkaian nyata <i>Buzzer</i> dengan <i>Arduino Uno</i> .....	52
Gambar 4.11	Listing Program Pengujian <i>Buzzer</i> .....	53
Gambar 4.12	Kondisi Volume Cairan Infus Berada Pada 334,64 mL.....	55
Gambar 4.13	Kondisi Volume Cairan Infus Berada Pada 223,85 mL.....	56
Gambar 4.14	Kondisi Volume Cairan Infus Berada Pada 2,89 mL.....	57
Gambar 4.15	Kurva Kondisi Volume dengan Jarak Sensor.....	58

## DAFTAR LAMPIRAN

1. Program <i>Microcontroller ATmega 328</i> pada <i>Arduino</i> .....	62
2. Table Hasil Pengujian Volume dan Jarak .....	67
3. Datasheet <i>Arduino Uno</i> dan Datasheet Sensor HY-SRF05 .....	68



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Infus adalah suatu piranti kesehatan yang dalam kondisi tertentu digunakan untuk menggantikan cairan yang hilang dan menyeimbangkan elektrolit tubuh. (Handaya, 2010)

Pada kondisi emergensi misalnya pada pasien dehidrasi, stress metabolic berat yang menyebabkan *syok hipovolemik*, *asidosis*, *gastroenteritis* akut, demam berdarah *dengue* (DBD), luka bakar, *syok hemoragik* serta trauma, infus dibutuhkan dengan segera untuk menggantikan cairan tubuh yang hilang. (Handaya, 2010)

Infus juga digunakan sebagai larutan awal bila status elektrolit pasien belum diketahui, misal pada kasus dehidrasi karena asupan oral tidak memadai, demam, dan lain-lain.

Karena fungsinya yang sangat penting, proses pemasangan infus harus dilakukan dengan benar yakni sesuai dengan prosedur yang telah ditetapkan untuk menghindari timbulnya komplikasi yang dapat memperburuk kondisi pasien. Selain proses pemasangan infus, proses lain yang sering disepelekan yaitu proses penggantian kantong infus saat cairan infus mendekati habis juga sangat berpengaruh pada proses terapi pasien.

Pada kenyataannya, perawat atau tenaga medis terkadang lalai terhadap tugasnya untuk mengganti kantong cairan infus pasien karena keterbatasan waktu dan tenaga. Padahal hal ini juga dapat menyebabkan timbulnya komplikasi lain

antara lain darah dari pasien dapat tersedot naik ke selang infus sehingga mengganggu kelancaran aliran cairan infus.

Selain itu, jika tekanan pada infus tidak stabil, darah yang membeku pada selang infus dapat tersedot kembali masuk ke dalam pembuluh darah. Darah yang membeku (*Blood Clot*) tersebut dapat beredar ke seluruh tubuh dapat menyumbat kapiler darah di paru sehingga menyebabkan emboli di paru. (Wait et al, 2004)

Jika berbagai hal tersebut terjadi maka tempat pemasangan infus harus dipindahkan dan dipasang pembuluh darah vena lain, yang tidak menutup kemungkinan dapat menyebabkan timbulnya berbagai komplikasi yang jauh lebih berbahaya akibat pemasangan yang tidak dilakukan dengan benar. (Dokudok's Team, 2017)

Berdasarkan latar belakang tersebut maka dalam penelitian ini akan dirancang suatu sistem pendeteksian kondisi cairan infus yang secara *real time* dimonitoring di ruang perawat. Harapannya adalah dengan diterapkannya alat ini maka permasalahan yang akan timbul karena kelalaian petugas dapat diminimalisir.

## 1.2 Rumusan Masalah

Adapun rumusan masalah yang dapat diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang alat monitoring ketinggian cairan infus menggunakan *Arduino Uno*?
2. Apa yang menjadi keunggulan alat monitoring ketinggian cairan infus menggunakan *Arduino Uno* dibandingkan dengan alat infus biasa?
3. Variabel apa yang diukur sehingga perawat dapat mengetahui kondisi cairan infus untuk segera diganti?

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Merancang alat pendeteksi ketinggian cairan infus yang menggunakan Sensor Ultrasonik HY-SRF05.
2. Memprogram *Arduino Uno* agar dapat menunjukkan volume cairan infus adalah baik.

#### **1.4 Batasan Masalah**

Agar tujuan ini mencapai hasil yang diharapkan, maka penulis membatasi masalah dalam laporan ini antara lain:

1. Warna cairan infus yang dideteksi oleh alat rancangan adalah warna bening.
2. Tidak membahas hubungan kesehatan pasien terhadap alat yang dirancang.
3. Jika nilai angka mulai minus yang ditampilkan oleh *LCD* adalah menunjukkan bahwa volume cairan infus harus segera diganti.

#### **1.5 Manfaat Pembuatan Alat**

Pembuatan Skripsi ini diharapkan bermanfaat bagi:

1. Bagi Rumah Sakit/Perawat  
Sebagai solusi alternatif yang dapat membantu dan mempermudah perawat dalam menjalankan tugasnya.
2. Bagi Masyarakat/Pasien  
Dapat mengurangi kecemasan bagi pihak pasien atau pihak rumah sakit jika cairan infus lupa digantikan.

## **1.6 Sistematika Pembahasan**

Sistematika penulisan pada masing-masing bab adalah sebagai berikut:

### **1. BAB I PENDAHULUAN**

Menjelaskan secara singkat tentang latar belakang penelitian, rumusan masalah penelitian, batasan masalah, tujuan dan manfaat penelitian serta sistematika pembahasan.

### **2. BAB II TEORI PENUNJANG**

Bab ini membahas tentang teori-teori pendukung dalam penelitian ini sehingga hasil yang didapatkan lebih optimal.

### **3. BAB III METODOLOGI PENELITIAN**

Berisi tentang bagaimana metode penelitian dilakukan, yang meliputi bagaimana cara pengambilan data, cara perancangan dan pembuatan alat, serta cara pengujiannya.

### **4. BAB IV HASIL DAN PEMBAHASAN**

Berisi penjelasan tentang hasil perancangan alat, pengujian dan pembahasannya.

### **5. BAB V PENUTUP**

Berisi tentang kesimpulan dan saran yang didapatkan dari penelitian yang dilakukan.

## BAB II

### TEORI PENUNJANG

#### 2.1 *Arduino*

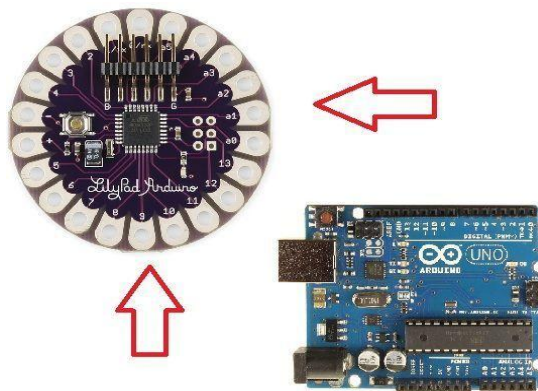
*Arduino* menyatakan perangkat lunak dan perangkat keras yang digunakan untuk mengontrol sejumlah perangkat elektronik seperti sensor suhu, penampil *LCD*, dan motor. *Arduino Integrated Development Equipment (Arduino IDE)* adalah perangkat lunak yang digunakan untuk mengembangkan program dan memungkinkan program diunggah ke papan *Arduino*. Program adalah kumpulan instruksi yang ditujukan untuk mengendalikan komputer. Papan *Arduino* merupakan elektronis yang berisi *microcontroller* berbasis *ATMega*, sedangkan *microcontroller* adalah sebuah keping (*integrated circuit*) yang mengandung prosesor atau memproses dan sekaligus memori yang berguna untuk menyimpan data. (Kadir, 2016)

Berikut ini adalah Gambar 2.1 yang memperlihatkan contoh bentuk fisik sebuah *microcontroller*:



**Gambar 2.1 : Microcontroller ATMega 328**  
**Sumber: <https://lapakrobotika.com/atmega-328A>**

Papan *Arduino* sendiri ada beragam ukuran. Sebagai contoh, *Arduino Uno* berukuran sebesar kartu kredit, sedangkan *LilyPad* berbentuk lingkaran berdiameter sekitar 5,5 cm. Gambar 2.2 menunjukkan contoh *Arduino Uno* dan *LilyPad*. Berdasarkan informasi di halaman *webS4A.cat*, *Arduino Uno* merupakan jenis *Arduino* yang telah diuji dapat bekerja dengan *Scratch for Arduino*.(Kadir, 2016)



**Gambar 2.2 : Arduino Uno dan LilyPad**  
 Sumber: <https://fr.pinterest.com/explore/arduino-lilypad/>

Papan *Arduino* mengandung empat *LED* yang masing-masing diberi kode *L*, *TX*, *RX*, dan *ON*. Gambar 2.3 menunjukkan keempat tombol tersebut. Peran setiap *LED* adalah seperti berikut:

1. *L*: *LED* ini terhubung ke pin bernomor 13.
2. *TX*: *LED* ini menandakan terdapat pengiriman data dari *Arduino*.
3. *RX*: *LED* ini menandakan terdapat penerimaan data.
4. *ON*: *LED* ini menyala sekiranya *Arduino* mendapatkan pasokan listrik.(Kadir, 2016)



**Gambar 2.3 : Led-led di Arduino**  
 Sumber: <https://arduino-info.wikispaces.com/GettingStarted-Software>

*Arduino Uno* mengandung 13 pin digital dan 6 pin analog. Pin digital berarti pin-pin yang mempunyai nilai digital, yang berarti kemungkinannya hanya ada dua buah, yaitu 1 atau 0. Pin analog adalah pin yang nilainya sangat bervariasi, yaitu berupa bilangan bulat antara 0 dan 1023. Sering kali kita menggunakan pin digital, tetapi kadang-kadang kita memerlukan pin analog.

Di *Arduino*, nilai digital 0 menyatakan 0 volt (0V) dan nilai digital 1 menyatakan nilai 5 volt (5V). Nilai analog menyatakan nilai yang identik dengan tegangan yang berkisar antara 0V dan 5V. (Kadir, 2016)



Berikut ini adalah Tabel 2.1 yang menampilkan *Index Board Arduino*:

**Tabel 2.1: Indeks Board Arduino Uno**

a. Mikrokontroler	ATmega328
b. Tegangan Operasi	5V
c. Tegangan <i>Input</i> (disarankan)	7-12V
d. Batas Tegangan <i>Input</i>	6-20V
e. pin Digital I/O	14 pin
f. Pin Analog <i>Input</i>	6 pin
g. Arus DC per I/O pin	40 mA
h. Arus DC untuk pin 3.3V	50 mA
i. Flash Memory	32 KB
j. SRAM	2 KB
k. EEPROM	1 KB
l. <i>Clock</i>	16 MHz

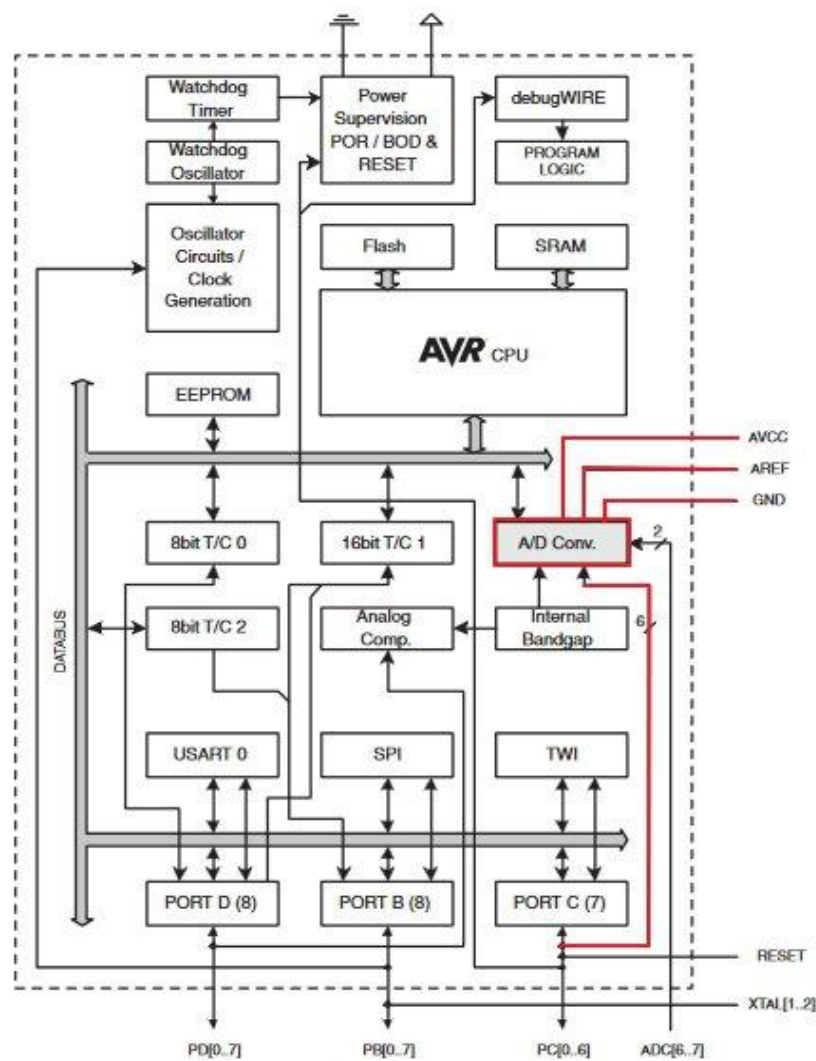
Sumber: <http://widuri.raharja.info/index.php/SI1233472982>

## 2.2 Arsitektur *Microcontroller ATmega*

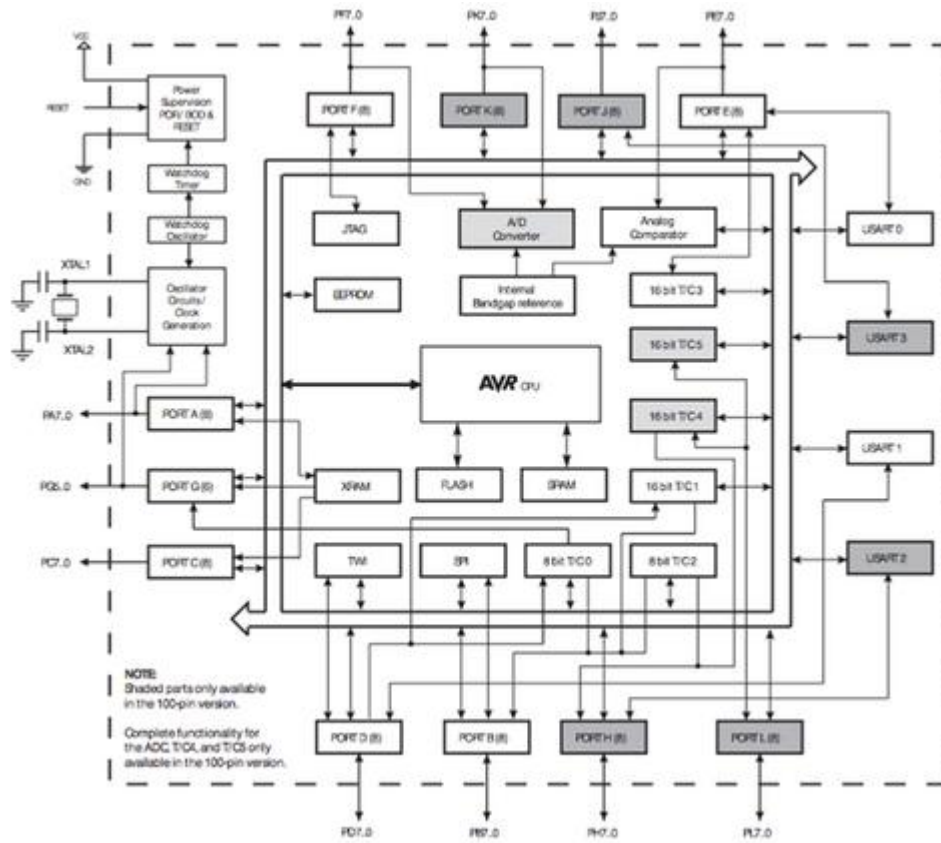
Seluruh *microcontroller* yang diimplementasikan pada produk *Arduino* menggunakan *ATmega* keluarga *AVR*. Salah satunya seri *ATmega328* (Gambar 2.4a) dengan sejumlah fitur diantaranya *ON-Chip System Debug*, 5 ragam tidur (*Mode Sleep*), 6 saluran *ADC* yang mendukung reduksi derau, ragam hemat daya (*Power-save Mode, Power-down*), dan ragam siaga (*Standby Mode*). (Istiyanto, 2014)

*Microcontroller ATmega328* paling umumnya digunakan pada *board Arduino* seperti *UNO, Duemilanove, Nano*, dan lain-lain. Sedangkan *ATmega1280* (Gambar 2.4b) dipakai pada tipe *ArduinoMega*, baik *ATmega328* maupun *ATmega 1280*, keduanya menggunakan kristal 16 MHz sebagai

pembangkit *clock*. Keduanya juga memiliki blok memori *Flash* untuk penyimpanan instruksi program, *SRAM* untuk penyimpanan variabel data sementara, dan *EEPROM* sebagai media penyimpanan data yang tetap tersimpan meskipun *microcontroller* dalam kondisi tidak dicatu. Fitur *microcontroller AVR* seri lainnya, seperti *ATMega168* atau *2560* tidak jauh berbeda dengan 328 atau 1280, kecuali pada ukuran kapasitas blok memori *EEPROM*, *Flash*, dan *SRAM*. (Istiyanto, 2014).



(a)



(b)

**Gambar 2.4:**  
**Blok diagram microcontroller (a) ATmega 328 dan (b) ATmega 1280**

*Sumber:*

<https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/>

<http://readingrat.net/arduino-mega-2560-circuit-diagram/arduino-mega-2560-circuit-diagram-the-wiring-diagram/>

### 2.3 Definisi Program

Menurut Kadir dan Heriyanto(2005:2) mengemukakan bahwa “suatu program ditulis dengan menguti kaidah bahasa pemrograman tertentu. Bahasa pemrograman dapat dianalogikan dengan bahasa yang digunakan manusia (bahasa manusia)”.

Sebagaimana diketahui, ada bermacam-macam bahasa manusia, seperti bahasa Inggris, bahasa Indonesia, ataupun bahasa Batak. Kumpulan instruksi dalam bahasa manusia yang berupa sejumlah kalimat dapat anda analogikan dengan suatu program. Manusia dapat mengerjakan suatu intruksi berdasarkan kalimat-kalimat dan komputer dapat menjalankan suatu instruksi program.

Dalam konteks pemrograman, terdapat sejumlah bahasa pemrograman, seperti *Pascal*, *C*, *C++*, dan *BASIC*. Secara garis besar, bahasa-bahasa pemrograman dapat dikelompokan menjadi:

1. Bahasa beraras-tinggi (*high-level language*)
2. Bahasa beraras-rendah (*low-level language*)

Bahasa beraras tinggi adalah bahasa pemrograman yang berorientasi kepada bahasa manusia. Program dibuat dengan menggunakan bahasa pemrograman yang mudah dipahami oleh manusia, biasanya menggunakan kata-kata bahasa Inggris; misalnya *IF* untuk menyatakan “jika” dan *AND* untuk menyatakan “dan”. Yang termasuk dalam kelompok bahasa ini adalah bahasa *C*, *C++*, *Pascal* dan *BASIC*.

Bahasa beraras rendah adalah bahasa pemrograman yang berorientasi kepada mesin. Bahasa ini menggunakan kode biner (yang hanya mengenal kode 0 dan 1) atau suatu kode sederhana untuk menggantikan kode-kode tertentu dalam sistem biner. Yang tergolong dalam kelompok bahasa ini adalah bahasa mesin dan

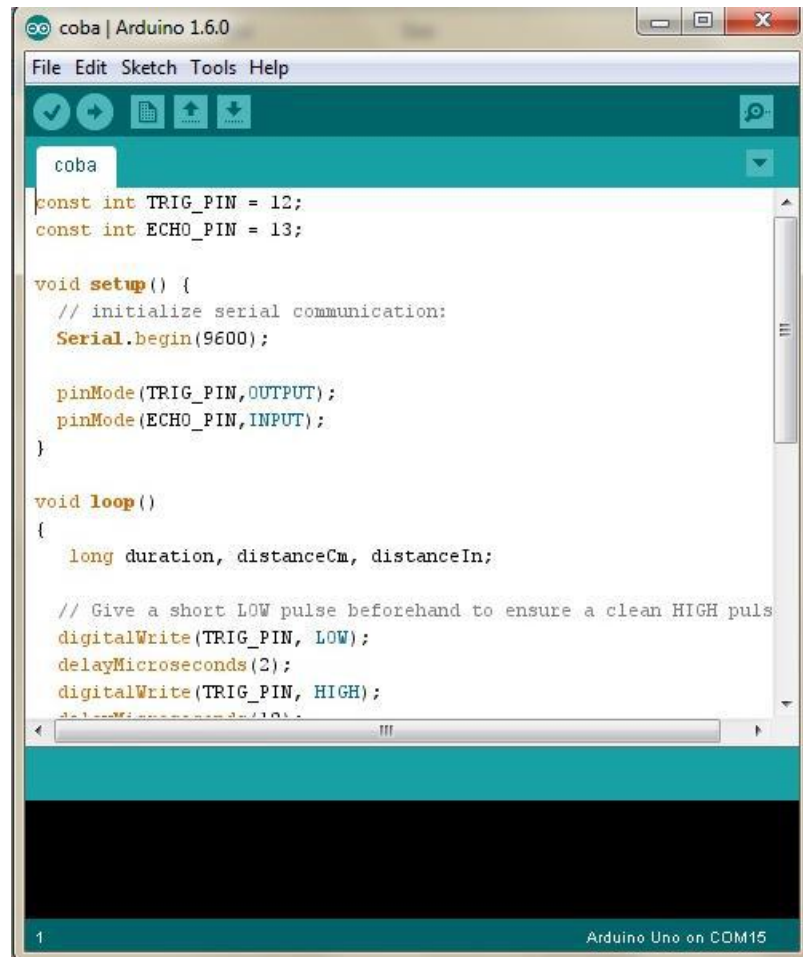
bahasa rakitan. Bahasa-bahasa itu ini sangat sulit untuk dipahami oleh orang awam dan sangat membosankan bagi pemrogram yang sudah terbiasa dengan bahasa beraras tinggi. Pemrogram harus benar-benar menguasai operasi komputer secara teknis. Namun bahasa generasi ini memberikan eksekusi program yang sangat cepat. Selain itu bahasa mesin sangat bergantung pada mesin (*machinedependent*) artinya, bahasa mesin antara satu mesin dengan mesin yang lain jauh berbeda.

#### **2.4 *Software Arduino IDE***

*IDE (Integrated Development Environment)* adalah sebuah program spesial yang berjalan di komputer yang mengizinkan *user* menulis *sketch* untuk *board Arduino* dalam bentuk bahasa pemrograman yang mudah menggunakan Bahasa *Processing*. *Software Arduino* ini dapat diinstal di berbagai *OS (operatingsystem)* seperti: *LINUX, Mac OS, Windows*. *Software Arduino IDE* terdiri dari 3 bagian yaitu:

1. *Editor* program, untuk menulis dan mengedit program dalam bahasa *Processing*. *Listing* program *Arduino* disebut *sketch*.
2. *Compiler*, modul yang berfungsi mengubah bahasa *Processing* ke dalam kode biner, karena kode biner adalah satu-satunya bahasa program yang dipahami oleh *microcontroller*.
3. *Uploader*, modul yang berfungsi memasukkan kode biner ke dalam memori *microcontroller*.(Feri, 2011)

Gambar 2.5 di bawah ini adalah tampilan *Arduino IDE*:



```

coba | Arduino 1.6.0
File Edit Sketch Tools Help
coba
const int TRIG_PIN = 12;
const int ECHO_PIN = 13;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop()
{
  long duration, distanceCm, distanceIn;

  // Give a short LOW pulse beforehand to ensure a clean HIGH puls
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
}
1 Arduino Uno on COM15

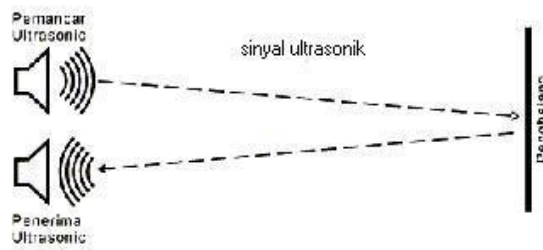
```

**Gambar 2.5: Tampilan *Arduino IDE* Versi 1.6.4**

Sumber: <https://learn.adafruit.com/add-boards-arduino-v164/overview>

## 2.5 Sensor Ultrasonik

Sensor *Ultrasonik* adalah sensor yang bekerja dengan cara memancarkan suatu gelombang dan kemudian menghitung waktu pantulan gelombang tersebut. Gelombang ultrasonik bekerja pada frekuensi mulai 20 kHz hingga sekitar 20 MHz. Frekuensi kerja yang digunakan dalam gelombang ultrasonik bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan rendah pada fasa gas, cair hingga padat. (Prawiroredjo dan Nyssa, 2008)



**Gambar 2.6: Cara kerja sensor ultrasonik**

*Sumber:*

<http://roboticbasics.blogspot.co.id/2015/01/membuat-robot-menggunakan-sensor-jarak-ultrasonik-srf04.html>

Pada Gambar 2.6 digambarkan cara kerja sensor ultrasonik. Sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan, dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh rangkaian *receiver* dikirimkan ke rangkaian *microcontroller* untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul). (Prawiroredjo dan Nyssa, 2008)

Secara matematis gelombang ultrasonik dapat dirumuskan sebagai:

$$x = v.t/2 \quad (2.1)$$

Pada rumus 2.1,  $x$  adalah jarak dalam satuan meter,  $v$  adalah kecepatan suara yaitu 340 m/detik dan  $t$  adalah waktu tempuh dalam satuan detik. Dijabarkan:

$$x = (34.000 * \text{waktu}) / 1.000.000 * 2 \quad (2.2)$$

Pada rumus 2.2,  $x$  adalah jarak tempuh dalam satuan cm, waktu adalah waktu tempuh dua kali lintasan ultrasonik hingga permukaan objek dalam satuan mikrosekon maka dikonversi menjadi sekon dibagi dengan 1.000.000 dan 34.000 adalah kecepatan suara dalam satuan cm/s. (Prawiroedjo dan Nyssa, 2008)

Ketika gelombang ultrasonik menumbuk suatu penghalang maka sebagian gelombang tersebut akan dipantulkan sebagian diserap dan sebagian yang lain akan diteruskan. Gelombang yang diserap akan dihitung oleh komparator dan diteruskan menjadi bilangan binary. (Prawiroedjo dan Nyssa, 2008)



**Gambar 2.7: Sensor jarak Ultrasonic HY-SRF05**  
 Sumber : <https://widuri.raharja.info/index.php/SII233473132>

Pada Gambar 2.7 merupakan gambaran fisik sensor ultrasonik jenis *HY-SRF05*. Sensor ultrasonik memiliki 4 pin yang digunakan yaitu, *pin vcc*, *pin ground*, *pin trigger* sebagai *input* dan *pin echo* sebagai *output*. Secara umum sensor ultrasonik digunakan untuk menghitung jarak dari suatu objek yang berada

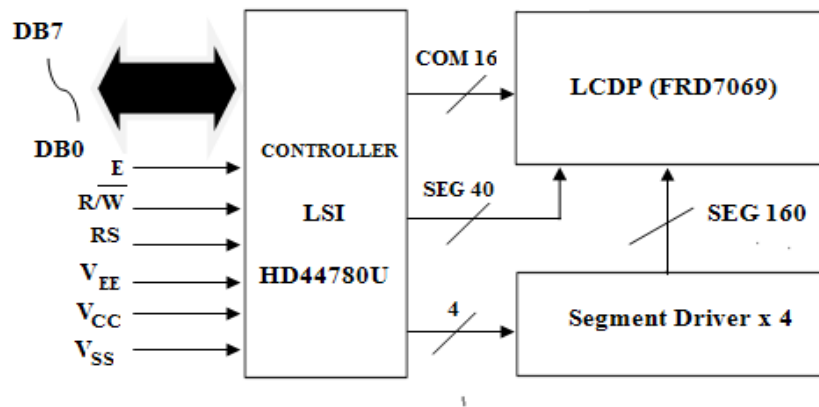


di depan sensor tersebut. Sehingga dengan fungsinya tersebut, sensor ultrasonik biasa digunakan pada perangkat yang membutuhkan perhitungan jarak. Contoh: pendeteksi ketinggian air, pemantauan ketersediaan air pada bak penampung, memantau kecepatan kendaraan bermotor dan pendeteksi dini banjir berdasarkan ketinggian dan kecepatan air. Spesifikasi sensor ultrasonik HY-SRF05 sebagai berikut:

1. Bekerja pada tegangan DC 5 volt.
2. Beban arus sebesar 30 mA – 50 mA.
3. Menghasilkan gelombang dengan frekuensi 40 KHz.
4. Jangkauan jarak yang dapat dideteksi 3 cm – 400 cm.
5. Membutuhkan *trigger input* minimal sebesar 10 uS.
6. Dapat digunakan dalam dua pilihan mode yaitu *input trigger* dan *output echo* terpasang pada pin yang berbeda atau *input trigger* dan *output echo* terpasang dalam satu pin yang sama. (Prawiroredjo dan Nyssa, 2008)

## 2.6 LCD

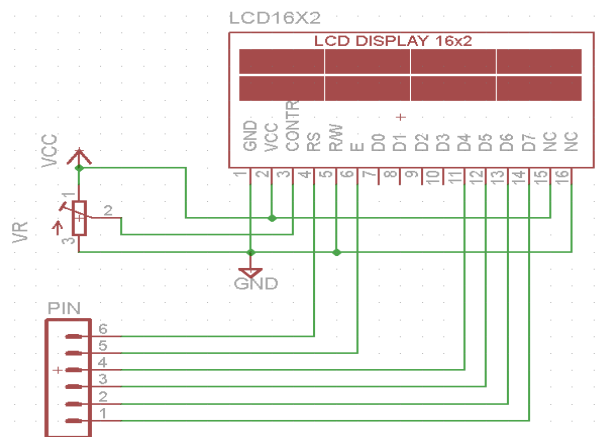
*LCD* adalah suatu *display LCD* adalah suatu display dari bahan cairan kristal yang pengoperasiannya menggunakan system dot matriks, *LCD* banyak digunakan sebagai *display* dari alat-alat elektronik seperti kalkulator, multimeter digital, jam digital dan sebagainya. Berikut ini adalah Gambar 2.8 yang memperlihatkan *blok diagram LCD*. (Supriyadi, 2014)



**Gambar 2.8 : Blok diagram LCD**

Sumber: [http://exploreembedded.com/wiki/LCD\\_16\\_x\\_2\\_Basics](http://exploreembedded.com/wiki/LCD_16_x_2_Basics)

LCD dapat digunakan dengan mudah dihubungkan dengan *microcontroller* AVR ATMega 8535. LCD yang digunakan dalam percobaan adalah LCD 2 x 16, lebar *display* 2 baris 16 kolom, yang mempunyai 16 pin konektor yang didefinisikan sebagai berikut terlihat pada Gambar 2.9 Konfigurasi *pin* pada LCD:

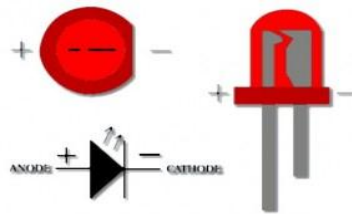


**Gambar 2.9 : Konfigurasi pin pada LCD**

Sumber: <http://www.bagusprehan.com/2013/12/konfigurasi-pin-lcd-16x2.html>

## 2.7 LED

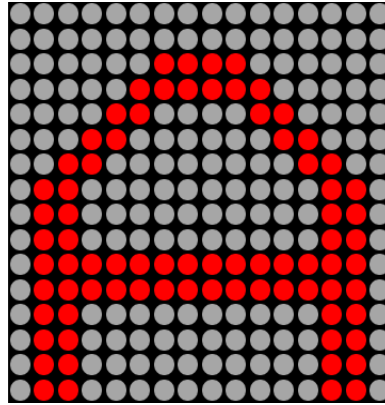
*LED* merupakan komponen yang dapat mengeluarkan emisi cahaya. *LED* merupakan produk temuan lain setelah dioda. Strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N juga melepaskan energi berupa energi panas dan energi cahaya. *LED* dibuat agar lebih efisien jika mengeluarkan cahaya. Untuk mendapatkan emisi cahaya pada semikonduktor, doping yang berbeda menghasilkan warna cahaya yang berbeda pula. Berikut ini adalah Gambar 2.10 yang memperlihatkan bentuk fisik *LED* dan simbol.(Budiharto dan Firmansyah, 2005)



**Gambar 2.10 : Bentuk fisik *LED* dan simbol**

Sumber: <http://elektronika-dasar.web.id/led-light-emitting-dioda/>

Pada saat ini warna-warna cahaya *LED* yang banyak adalah warna merah, kuning, dan hijau. *LED* berwarna biru sangat langka. Pada dasarnya semua warna bisa dihasilkan, namun akan menjadi sangat mahal dan tidak efisien. Dalam memilih *LED* selain warna, perlu diperhatikan tegangan kerja, arus maksimum dan disipasi dayanya. Rumah (*chasing*) *LED* dan bentuknya juga bermacam-macam, ada yang persegi empat, bulat, dan lonjong. Berikut ini adalah Gambar 2.11 yang memperlihatkan *LEDarray*.(Budiharto dan Firmansyah, 2005)



**Gambar 2.11 : LEDarray**

Sumber: <https://forum.arduino.cc/index.php?topic=293201.0>

## 2.8 Alarm/Buzzer

Alarm digunakan sebagai sistem peringatan tanda bahaya berupa bunyi atau suara. Sistem alarm yang digunakan adalah sistem *buzzer*. *Buzzer* atau bel listrik adalah suatu alat untuk memberi sinyal suara secara khas. Secara umum bel listrik sering digunakan untuk suatu rangkaian sensor dengan pengendali dan digunakan sebagai penanda yang berupa suara. Berikut ini adalah Gambar 2.12 bentuk fisik dari *buzzer*. (Tri Rahajoeningroem, Wahyudin, 2013)



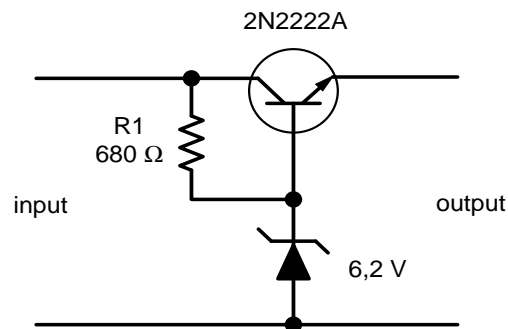
**Gambar 2.12 : Buzzer**

Sumber: Tri Rahajoeningroem, Wahyudin, 2013

## 2.9 Catu Daya

Rangkaian catu daya sangat dibutuhkan untuk memberi daya pada rangkaian kita. Rangkaian catu daya stabil namun sederhana dapat dibuat dari satu buah transistor sebagai penguat arus dan satu buah *dioda zener* sebagai penstabil tegangan.(Budiharto dan Firmansyah, 2005)

Berikut adalah Gambar 2.13 yaitu contoh rangkaian regulator tegangan:



**Gambar 2.13 : Regulator tegangan**  
 Sumber: Budiharto dan Firmansyah, 2005

Pada diagram di atas, tegangan stabil yang dihasilkan *dioda zener* sebesar 5,6 Volt (kita bisa memilih/membeli berbagai tipe tegangan *zener* yang kita inginkan), karena terjadi *drop* sekitar 0,6 V pada transistor antara *kolektor* dan *emitor*, maka tegangan *output* yang dihasilkan sebesar 5 V ( $5,6 \text{ V} - 0,6 \text{ V}$ ).(Budiharto dan Firmansyah, 2005)

## 2.10 Cairan *Intravena* (Infus)

Terapi cairan intravena merupakan pemberian cairan untuk penggantian cairan, pemberian obat, dan penyediaan nutrisi jika tidak ada pemberian dengan cara lain.(Smeltzer & Bare, 2001)

Umumnya cairan intravena diberikan untuk mencapai satu atau lebih tujuan berikut ini:

- a. Untuk menyediakan air, elektrolit, dan nutrisi untuk memenuhi kebutuhan sehari-hari.
- b. Untuk menggantikan air dan memperbaiki kekurangan elektrolit.
- c. Untuk menyediakan suatu medium untuk pemberian obat secara intravena.

Larutan elektrolit dianggap isotonik jika kandungan elektrolit totalnya (anion ditambah kation) kira-kira 310 mEq/L. Larutan dianggap hipotonik jika kandungan elektrolit totalnya kurang dari 250 mEq/L dan hipertonik jika kandungan elektrolit totalnya melebihi 375 mEq/L. Perawat juga harus mempertimbangkan osmolalitas suatu larutan, bahwa osmolalitas plasma adalah kira-kira 300 mOsm/L.

- a. Cairan isotonis: cairan yang diklasifikasikan isotonik mempunyai *osmolalitas* total yang mendekati cairan ekstraseluler dan tidak menyebabkan sel darah merah mengkerut atau membengkak. Contohnya saline normal (0,9% *natriumklorida*), larutan *ringer lactate*.
- b. Cairan hipotonik: tujuannya adalah untuk menggantikan cairan seluler, karena larutan ini bersifat hipotonis dibandingkan dengan plasma. Tujuan lainnya adalah untuk menyediakan air bebas untuk ekskresi sampah tubuh. Pada saat tertentu, larutan natrium hipotonik digunakan untuk mengatasi hipernatremia dan kondisi *hiperosmolar* yang lain. Contohnya saling berkekuatan menengah (*natrium klorida* 0,45%).
- c. Cairan hipertonik: dekstrosa 5% dalam air diberikan untuk membantu memenuhi kebutuhan kalori. Larutan salin juga tersedia dalam

konsentrasiosmolar yang lebih tinggi dari pada CES. Larutan-larutan ini menarik air dari kompartemen intraseluler ke ekstraseluler dan menyebabkan sel-sel mengkerut. Jika diberikan dengan cepat dan dalam jumlah besar, dapat menyebabkan kelebihan volume ekstraseluler dan mencetuskan kelebihan cairan sirkulasi dan dehidrasi. Berikut ini adalah Gambar 2.14 yang memperlihatkan botol infus.



**Gambar 2.14 : Botol infus**

*Sumber: <https://ask.fm/puspandini/answers/118905448178>*

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Tempat dan Waktu Penelitian**

##### **3.1.1 Tempat Penelitian**

Pembuatan dan pengujian alat monitoring ketinggian cairan infus menggunakan *Arduino Uno* ini dilakukan di:

1. Nama Tempat : Laboratorium Dasar Digital Universitas Medan Area.
2. Alamat : Jalan Kolam No. 1 Medan Estate.

##### **3.1.2 Waktu Penelitian**

Pembuatan dan pengujian sistem ini membutuhkan waktu dengan rincian sebagai berikut:

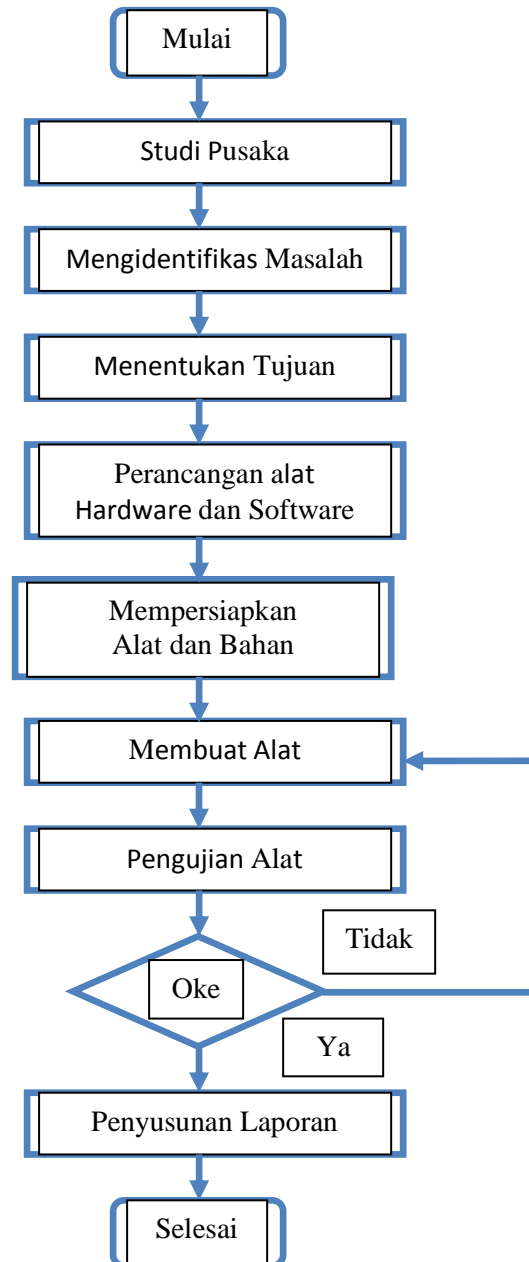
1. Penyediaan bahan dan alat : 1 minggu
2. Perancangan dan pembuatan seluruh sistem : 1 bulan
3. Pengujian sistem : 1 minggu
4. Penyusunan laporan Skripsi : 1 minggu

#### **3.2 Metoda Penelitian**

Pelaksanaan penelitian ini dilakukan dalam beberapa tahap untuk mempermudah dan memperjelas arah penelitian. Berikut ini adalah Gambar 3.1,



yaitu *flowchart* kerangka berfikir dalam penelitian, dimana berdasarkan *flowchart* inilah sebagai tahapan-tahapan yang dilakukan oleh peneliti dalam melakukan proses penelitian rancang bangun alat monitoring ketinggian cairan infus menggunakan *Arduino Uno*:



**Gambar 3.1 : Flowchart kerangka berfikir**

### 3.2.1 Alat dan Bahan

Alat-alat yang digunakan pada pembuatan sistem ini adalah: 1 set *tools mechanic*, gerinda, bor listrik, solder listrik, mistar, alat tulis. Alat-alat yang digunakan pada uji kinerja alat rancangan antara lain: multimeter, dan testpen.

Bahan elektrik maupun mekanik yang digunakan dalam pembuatan sistem ini secara umum adalah seperti pada Tabel 3.1 berikut:

**Tabel 3.1 : Penetapan komponen (bahan)**

No.	Komponen	No.	Komponen
1	Arduino Uno	10	Kabel pelangi
2	Sensor <i>Ultrasonik HY-SRF05</i>	11	Selang infus
3	Tabung Infus + Cairan	13	Steker listrik
4	Acrelic	14	Spicer besi
5	Balok kayu	15	Sekrup
6	Kawat penggantung infus	16	LCD 2x16
7	Pilox warna biru	17	Resistor 2K2
8	AC-DC Adaptor	18	Spicer plastik
9	Led dan Buzzer	19	Kabel Downloader

### 3.2.2 Rancangan Struktural

#### a. Dudukan Seluruh Sistem

Dudukan berfungsi sebagai tempat seluruh sistem elektrikal dan objek penelitian (botol infus + cairan) atau dalam hal ini adalah sebagai simulator skala kecil. Dudukan yang digunakan adalah terbuat dari kayu berbentuk balok persegi panjang dengan dimensi masing-masing sebagai berikut:

#### a.1 Balok kayu (sebagai tiang yang berdiri tegak):

Lebar = 4 cm

Tebal = 3,5 cm

Panjang = 87 cm

a.2 Balok kayu (sebagai penyangga tiang yang berdiri tegak):

Lebar = 4,5 cm

Tebal = 2,5 cm

Panjang = 20 cm

Jumlah = 2 buah (sisi kiri dan kanan)

a.3 Balok kayu (sebagai lengan balok kayu penyangga sisi kiri dan kanan):

Lebar = 5 cm

Tebal = 3,5 cm

Panjang = 29 cm

Jumlah = 1 buah

a.4 Balok kayu (sebagai lengan penggantung botol infus):

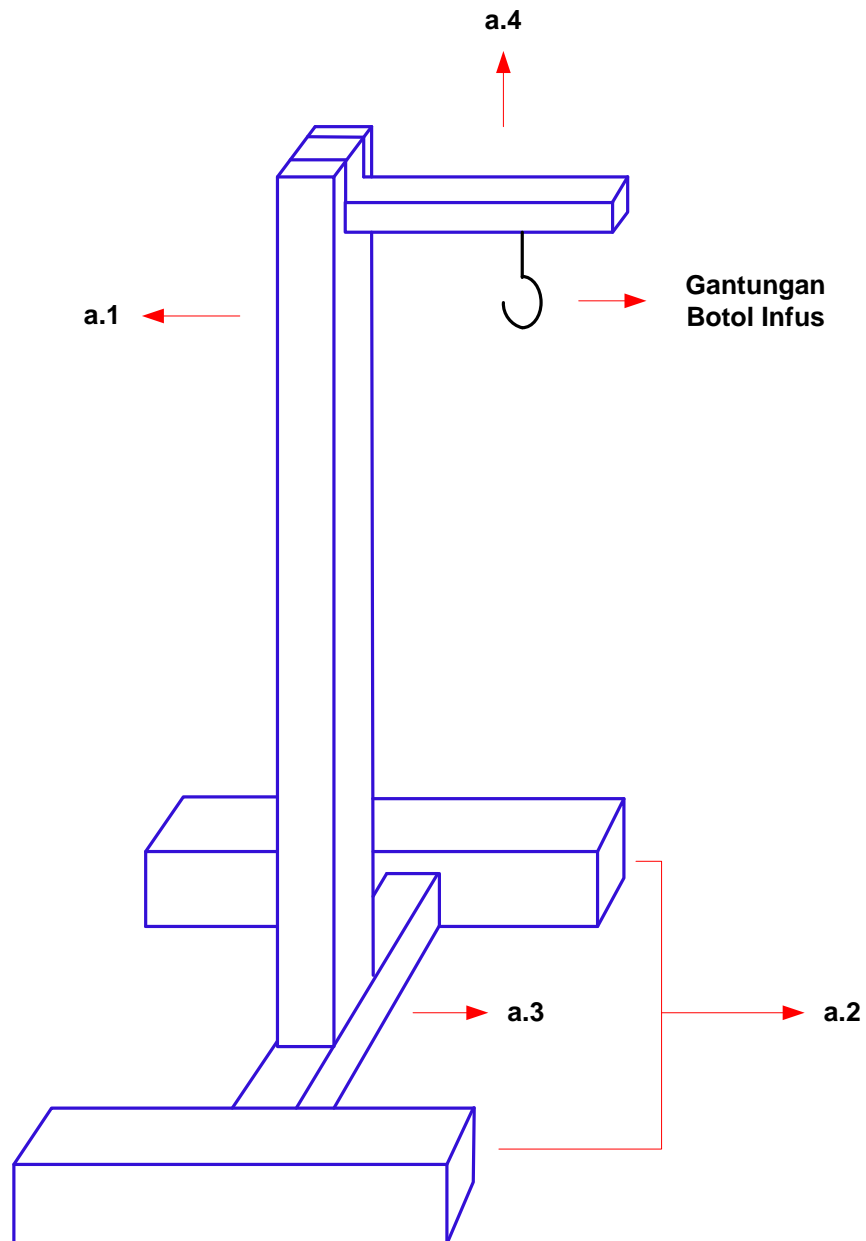
Lebar = 2,5 cm

Tebal = 2 cm

Panjang = 22,5 cm

Jumlah = 1 buah

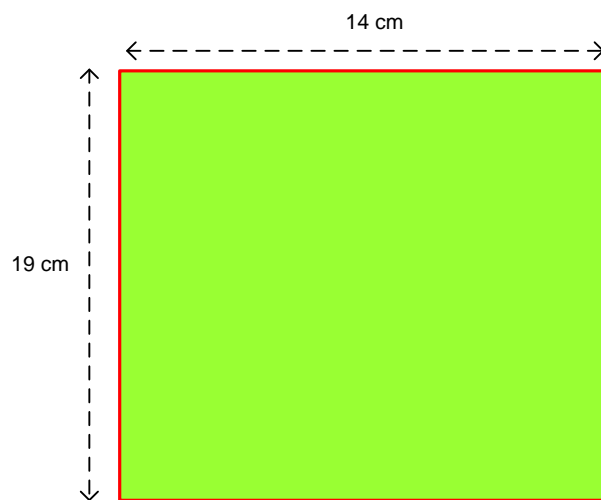
Selanjutnya ketiga balok kayu dengan dimensi seperti di atas didesain dan dikombinasikan satu sama lain sehingga membentuk seperti Gambar 3.2 berikut ini:



**Gambar 3.2 : Sketsa dudukan seluruh sistem**

b. Dudukan Sistem Elektrikal

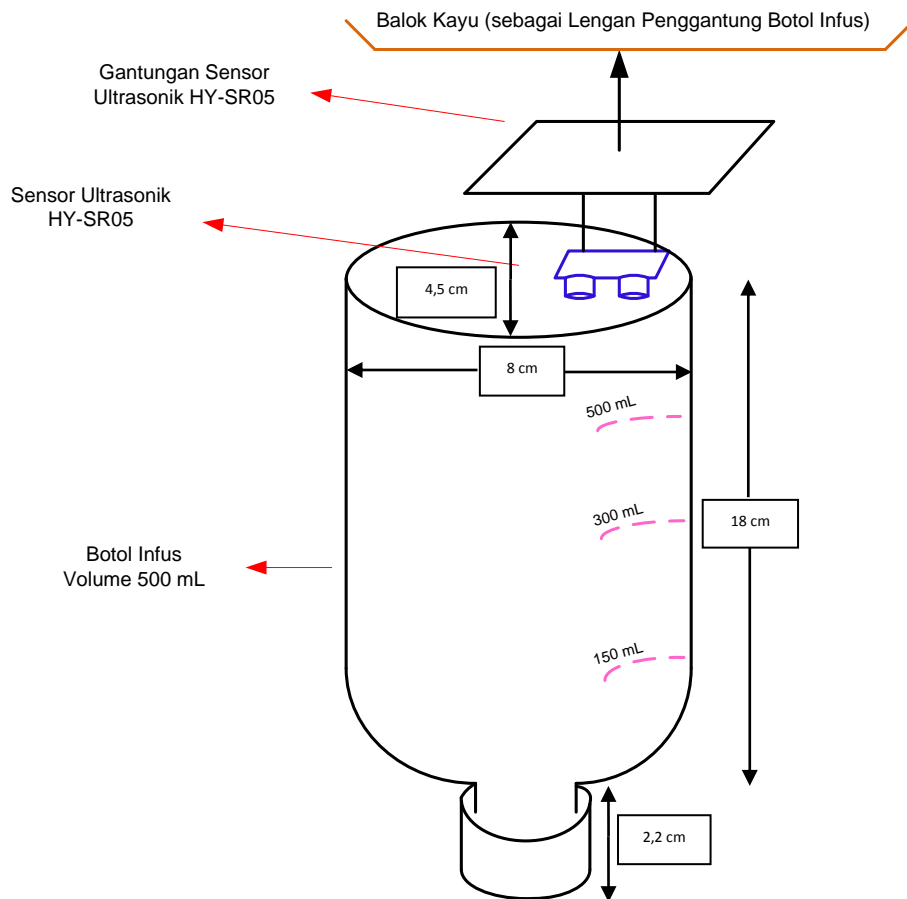
Dudukan ini berfungsi sebagai peletakan setiap sistem-sistem elektikal yang dirancang yang terdiri dari sistem minimum *Arduino Uno*, sistem monitoring (*LCD 2x16*), sistem indikator *Led*. Dudukan ini terbuat dari bahan *acrellic* dengan ketebalan 0,3 cm, dan selanjutnya bahan *acrellic* tersebut dirancang dan dibuat dengan model persegi panjang dimana panjangnya 19 cm, lebar 14 cm. Dimensi yang dibuat adalah bertujuan agar alat mudah dibawa kemana saja (*portable*) dan mudah diletakkan pada tiang dudukan yang berdiri tegak. Gambar 3.3 berikut ini adalah gambar yang menampilkan rancangan bagian dudukan sistem elektrikal:



**Gambar 3.3 : Desain dan dimensi dudukan sistem elektrikal**

c. Rancangan Tata Letak Sensor Ultrasonik

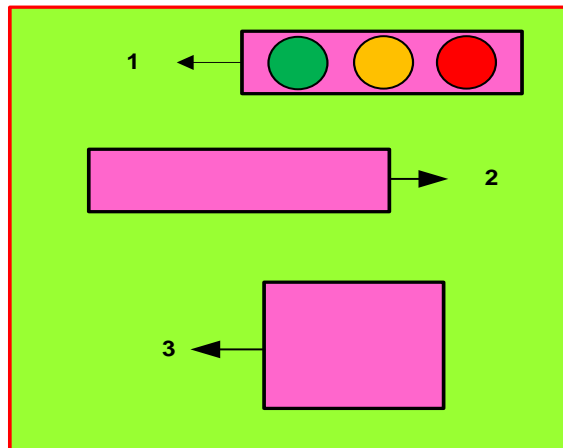
Berikut ini adalah Gambar 3.4 yaitu sketsa penempatan sensor ultrasonik HY-SR05 terhadap objek yang diukur pada penelitian ini yaitu ketinggian cairan infus dalam botolnya:



**Gambar 3.4:**  
Sketsa penempatan sensor ultrasonik dengan botol infus

d. Rancangan Tata Letak Sistem

Berikut ini adalah Gambar 3.5 yang menampilkan tata letak sistem-sistem yang akan dibuat pada penelitian ini:



**Gambar 3.5 : Desain tata letak sistem**

Keterangan Gambar 3.5:

1. Indikator *Led* (hijau, kuning, dan merah).
2. *LCD* 2x 16.
3. Sistem minimum *Arduino Uno*.

### 3.2.3 Rancangan Sistem Elektrikal

Rancangan sistem elektrikal yang dimaksud adalah meliputi:

1. Sistem AC-DC adaptor.
2. Sistem minimum *Arduino Uno*.
3. Sistem *indikator led*.
4. Sistem penampil data yaitu *LCD* 2x16.
5. Sistem *Buzzer*
6. Sistem secara keseluruhan.

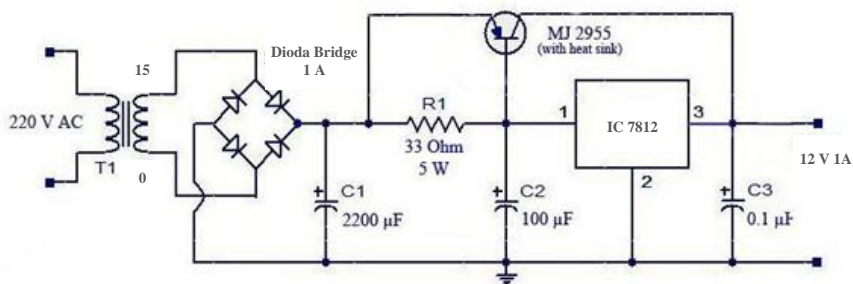
## A. Sistem AC-DC Adaptor

AC-DC adaptor yang dirancang adalah cukup menggunakan AC-DC adaptor yang sudah jadi dan lebih simpel yaitu dengan membelinya di toko-toko penjual komponen elektronika, namun hal yang harus diperhatikan adalah spesifikasinya harus sesuai dengan kebutuhan sistem yang dirancang secara keseluruhan. Berikut adalah Gambar 3.6 yang memperlihatkan bentuk fisik AC-DC adaptor yang digunakan beserta spesifikasinya yang tertera di badan adaptor:



Gambar 3.6 : AC-DC adaptor

Pada Gambar 3.6 di atas yang memperlihatkan bentuk fisik AC-DC adaptor dapat juga dilihat skema rangkaian didalamnya seperti Gambar 3.7 di bawah ini:



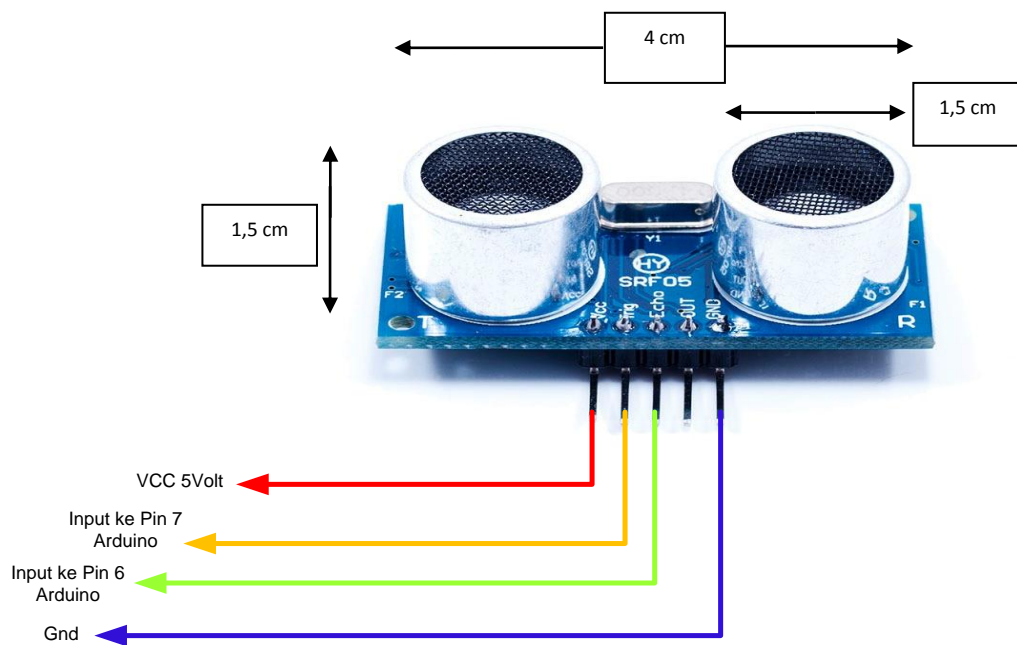
Gambar 3.7 : Skema rangkaian AC-DC adaptor



## B. Sistem Sensor Ultrasonik HY-SR05

Sistem ini juga tidak perlu dirancang ataupun dibuat lagi karena sensor ini sudah ada yang siap pakai namun yang perlu dipahami dan diperhatikan adalah bagaimana cara kerja sensor ini dan bagaimana cara penginstalasian pin-pin sensor terhadap sistem pengendali *Arduino Uno* agar dapat berfungsi sesuai karakteristiknya sebagai pendeteksi ataupun pengukur jarak benda.

Berikut ini akan dijelaskan bagaimana pola penginstalasian pin sensor ultrasonik yang disajikan dalam bentuk gambar. Berikut Gambar 3.8 yang memperlihatkan pola penginstalasiannya:



**Gambar 3.8 : Pola penginstalasian sensor ultrasonik HY-SRF05**

Keterangan pin Gambar 3.8:

*VCC = 5V Power Supply*

Pin sumber tegangan positif sensor atau pin 5V untuk koneksi ke tegangan 5V dc.

*Trig = Trigger/Penyulut.*

Pin ini yang digunakan untuk membangkitkan sinyal ultrasonik, dengan kata lain *Trigger Input* dipakai untuk memicu pembangkitan gelombang ultrasonic. Berupa sinyal '*HIGH*' selama minimal 100 us.

*Echo = Receive/Indikator.*

Pin ini yang digunakan untuk mendeteksi sinyal pantulan ultrasonik, dengan kata lain *Echo Output* untuk memantau kondisi logika, apakah gelombang ultrasonik sudah diterima kembali atau belum.

*GND = Ground/0V Power Supply*

Pin sumber tegangan negatif sensor.

*OUT = Mode (No Connection).*

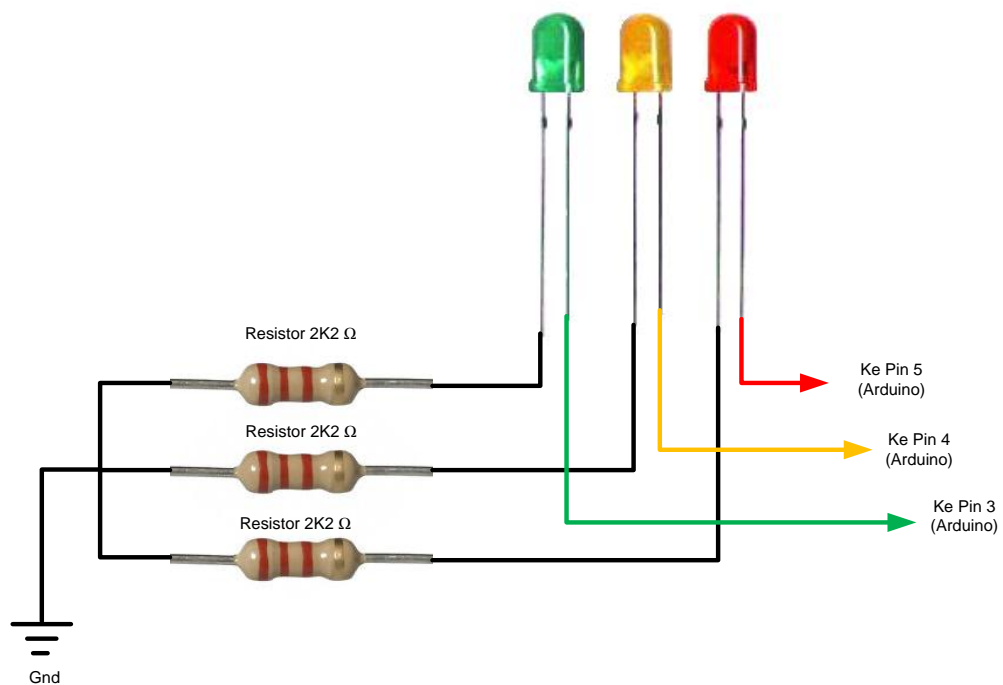
### **C. Sistem Indikator Led**

Sistem indikator *led* pada penelitian ini adalah digunakan sebagai pemberitahu atau kode dalam bentuk lampu berwarna berbeda-beda yang berfungsi untuk memperkuat hasil pemantauan dengan ketinggian yang diinginkan dalam penelitian ini, antara lain:

1. *Led* warna hijau sebagai pertanda jumlah mili liter ketersediaan cairan infus adalah 334,64 mL.

2. *Led* warna kuning sebagai pertanda jumlah mili liter ketersediaan cairan infus adalah 223,85 mL.
3. *Led* warna merah sebagai pertanda jumlah mili liter ketersediaan cairan infus adalah 2,89 mL.

Adapun rangkaian yang dibuat untuk sebuah sistem indikator *led* dapat dilihat seperti Gambar 3.9 berikut ini:

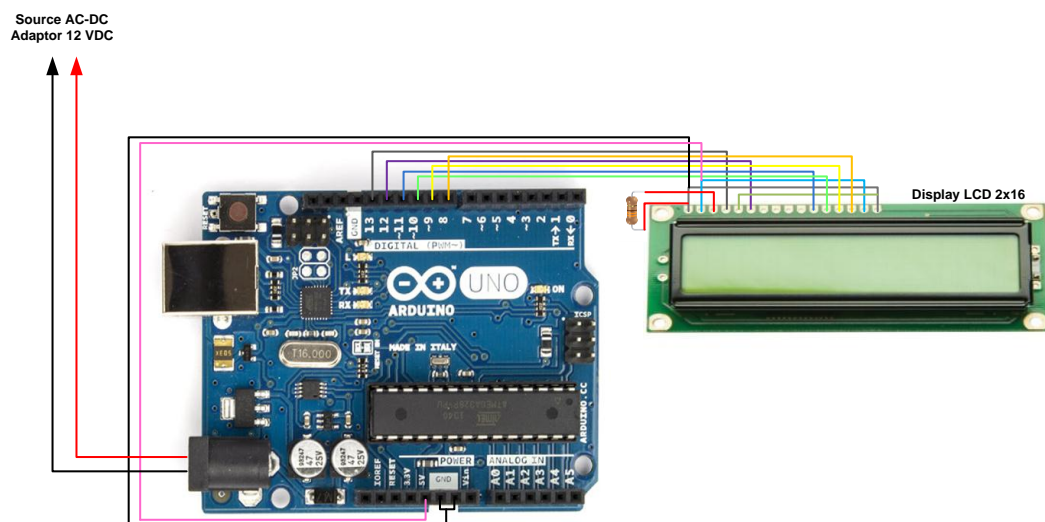


**Gambar 3.9 : Rangkaian sistem indikator *led***

#### D. Sistem Penampil Data (*LCD 2x16*)

Fungsi dari sistem ini adalah sebagai penampil data berupa informasi ketersediaan cairan infus pada botol dalam bentuk tulisan dan bukanlah dalam bentuk sinyal tegangan atau arus namun telah dirubah olehnya data tegangan menjadi suatu tulisan yang tertampil dilayarnya. Sedangkan tulisan yang tampil tergantung dari tulisan yang dibuat pada program yang dibuat.

Dalam penelitian ini perlu diketahui bagaimana pola penginstalasian *LCD 2x16* terhadap pin *Arduino Uno* agar dapat bekerja sebagai fungsi *LCD* sebagai penampil data. Berikut adalah Gambar 3.10 yaitu pola penginstalasian *LCD2x16* terhadap *Arduino Uno*:

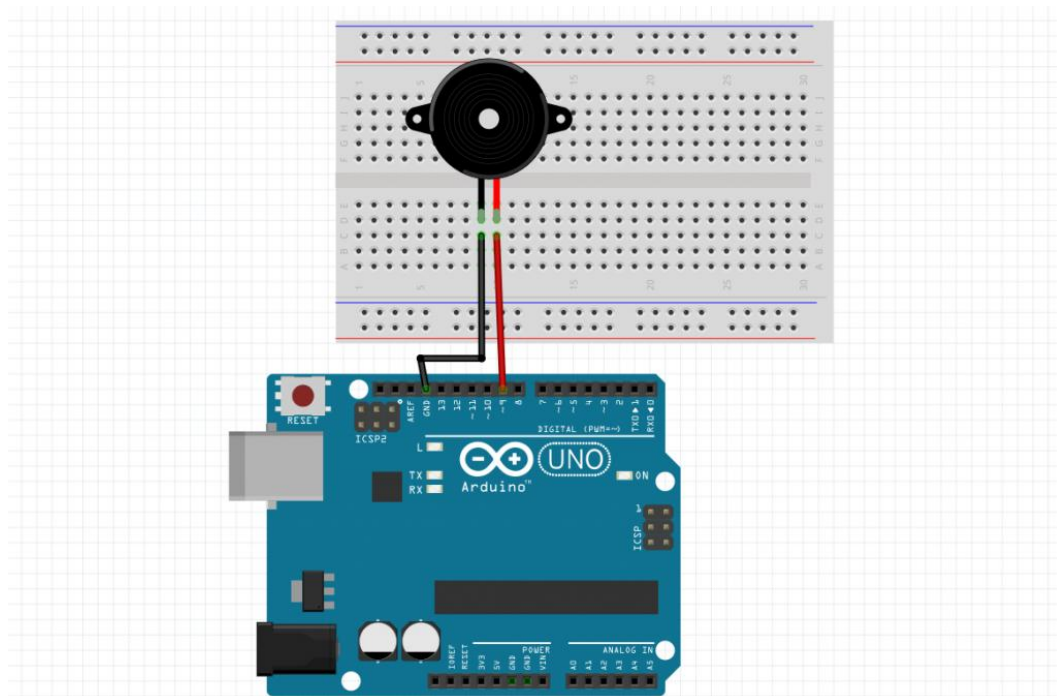


Gambar 3.10 : Pola instalasi *LCD 2x16* pada *Arduino Uno*

### E. Sistem *Buzzer*

Rangkaian *buzzer* pada penelitian ini berfungsi sebagai pemberitahu atau kode dengan mengeluarkan bunyi suara sebagai pertanda sensor mendeteksi *led* saat berwarna merah.

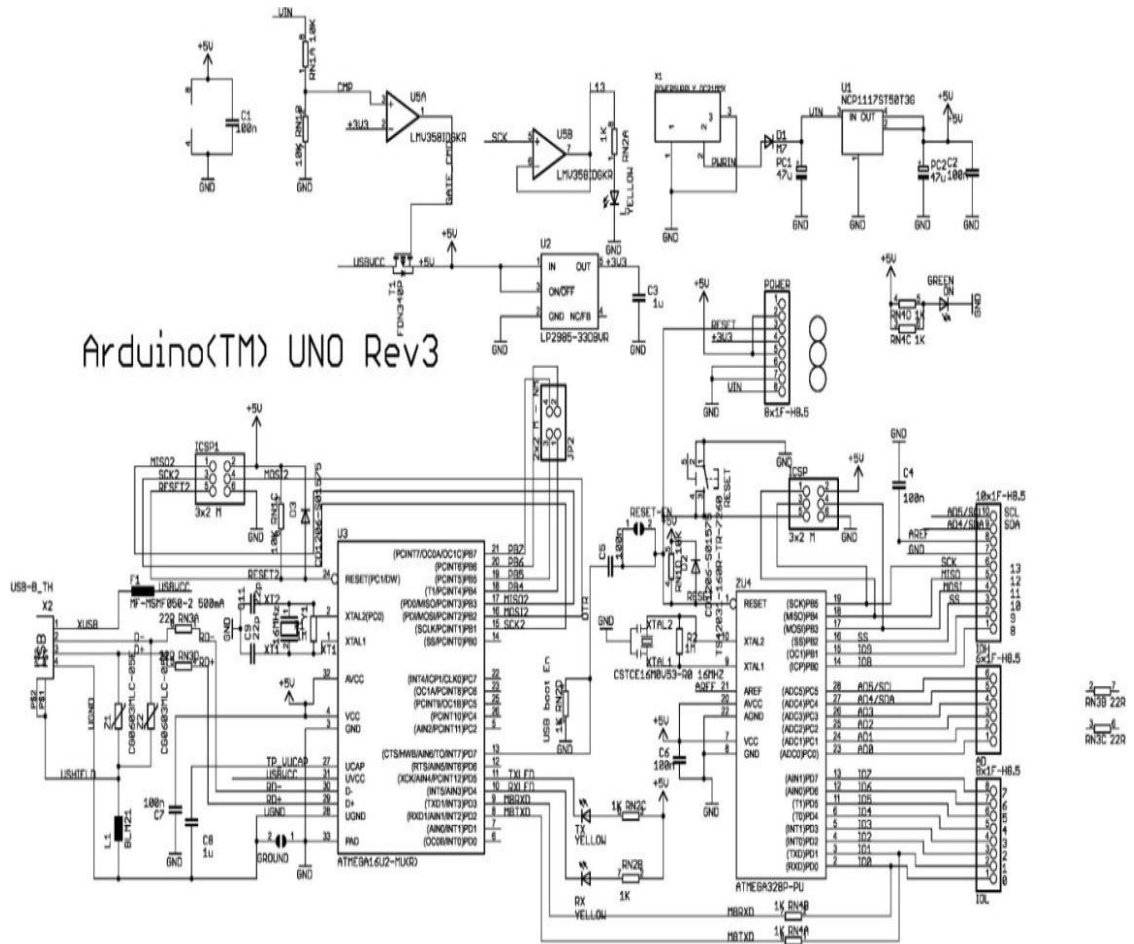
Pada Gambar 3.11 dibawah ini kaki negatif pada *buzzer* dihubungkan ke *ground* dan kaki positif *buzzer* dihubungkan ke *microcontroller*. Maka untuk menghidupkan *buzzer*, *port* yang terhubung ke *microcontroller* cukup mengeluarkan logika 1 (*high*) dan *buzzer* akan mati ketika *port* yang terhubung ke *microcontroller* mengeluarkan logika 0 atau (*low*).



Gambar 3.11: Pola instalasi *Buzzer* pada *Arduino Uno*

**F. Sistem Minimum Arduino Uno**

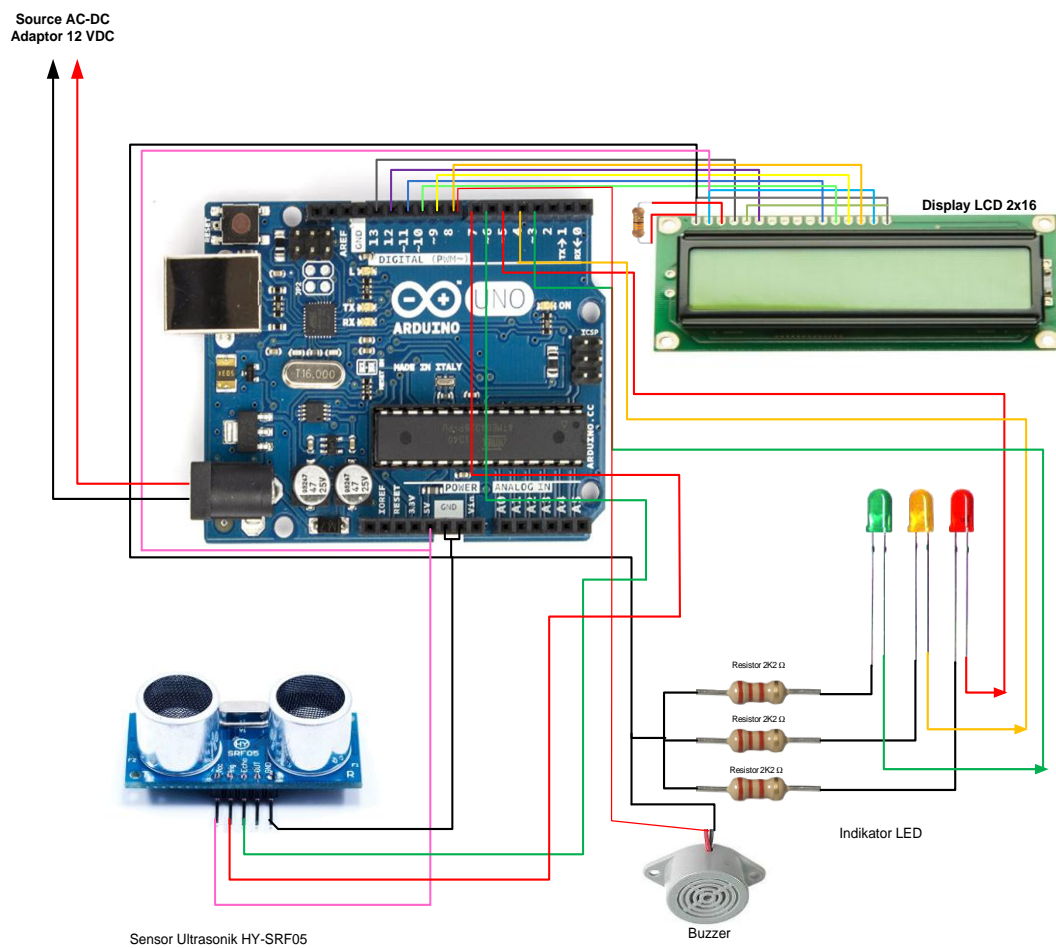
Sistem minimum *arduino* Gambar 3.12 di bawah ini adalah gambar yang menunjukkan skema rangkaian dari sistem minimum *Arduino Uno* beserta *microcontroller* ATmega 328:



**Gambar 3.12: Sistem minimum Arduino Uno**

## G. Sistem Secara Keseluruhan

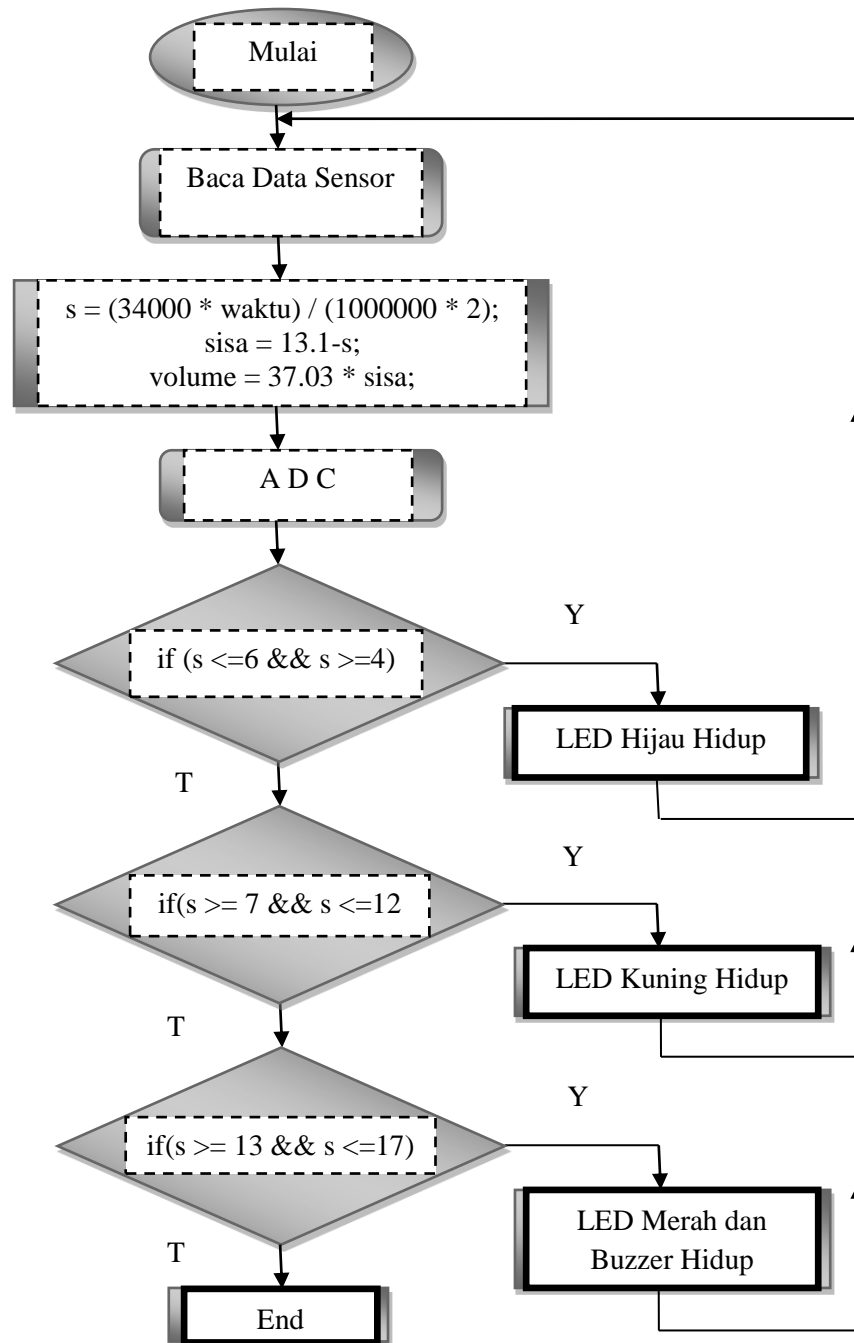
Dalam perancangan dan pembuatan sistem secara keseluruhan berarti seluruh komponen pembentuk sistem pemantauan ketinggian cairan infus akan dilakukan penggabungan seluruhnya baik dari segi mekanik maupun instalasi listriknya. Berikut Gambar 3.13 yang menampilkan skema rangkaian seluruh sistem.



Gambar 3.13 : Skema rangkaian seluruh sistem

### 3.3 Flowchart Program Monitoring Infus Menggunakan Arduino Uno

Berikut adalah Gambar 3.14 yang memperlihatkan alur kerja program monitoring infus menggunakan *Arduino Uno*:

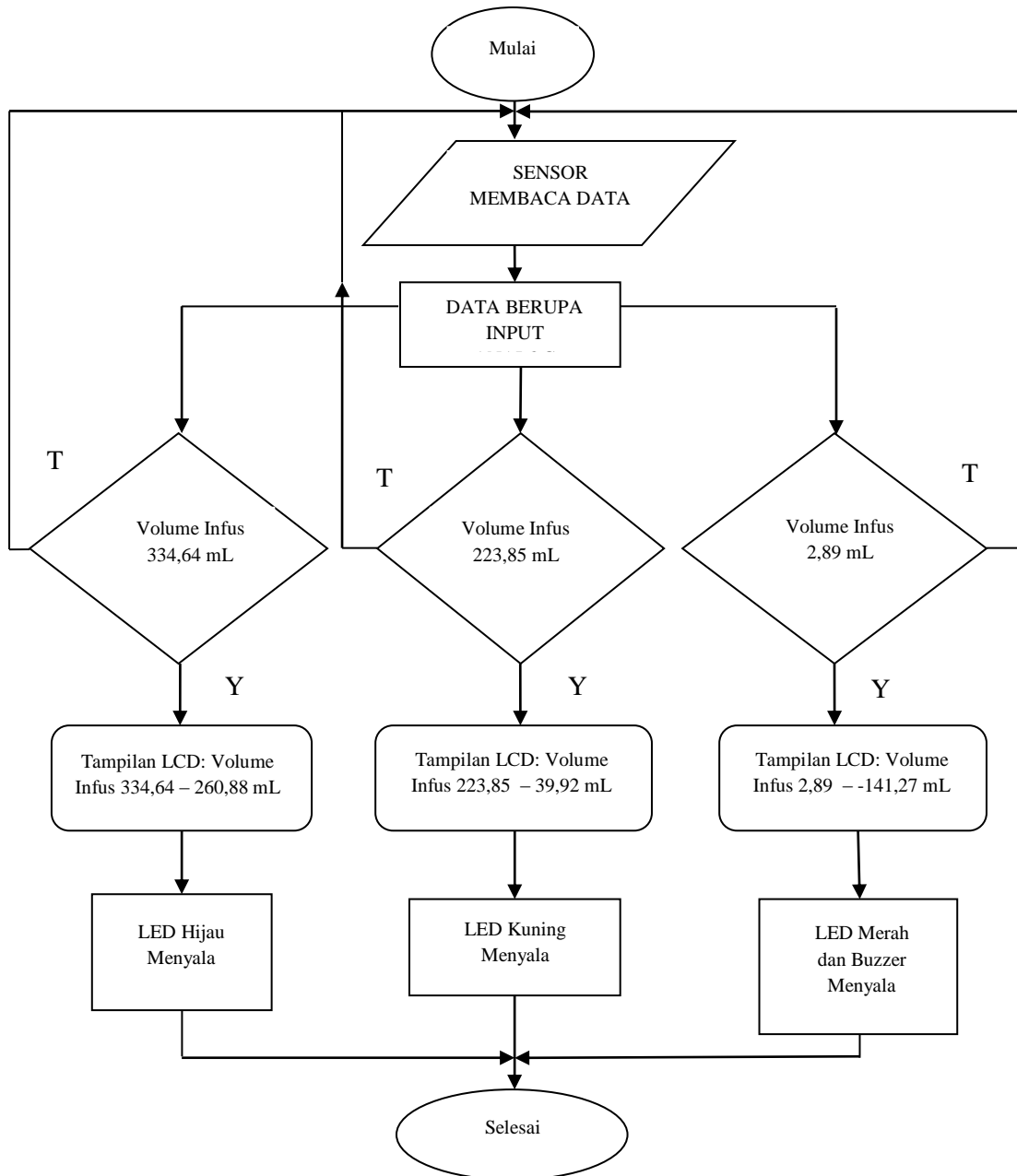


Gambar 3.14 : Flowchart program monitoring infus



### 3.4 Flowchart Sistem Kerja Alat

Berikut adalah Gambar 3.15 yang memperlihatkan alur kerja sistem monitoring ketinggian cairan infus menggunakan *Arduino Uno*:



Gambar 3.15 : Flowchart sistem kerja alat

## DAFTAR PUSTAKA

- Budiharto, Widodo dan Firmansyah, Sigit. 2005. *Elektronika Digital Dan Mikroprosesor*. Yogyakarta : Penerbit ANDI.
- Djuandi, Feri. 2011. *Pengenalan arduino*, <http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>. Diakses tanggal 21 Agustus 2017.
- Dokudok's Team. 2017. *Komplikasi Pemasangan Infus*. <http://dokudok.com/ketrampilan-klinis/komplikasi-pemasangan-infus/>. Diakses tanggal 30 Agustus 2017.
- Handaya, Yuda. 2010. *INFUS CAIRAN INTRAVENA (Macam-Macam Cairan Infus)*. <https://www.scribd.com/document/110031730/INFUS-CAIRAN-INTRAVENA>. Diakses tanggal 30 Agustus 2017.
- Istiyanto, Eko Jazi. 2014. *Pengantar Elektronika dan Instrumentasi*. Yogyakarta: Penerbit ANDI.
- Kadir, Abdul dan Heriyanto. 2005. *Algoritma Pemrograman Menggunakan C++*. Yogyakarta: Andi Offset.
- Kadir, Abdul. 2016. *Scratch for Arduino (S4A)-Panduan Mempelajari Elektronika dan Pemograman*. Yogyakarta. Penerbit ANDI.
- Prawiroredjo, Kiki dan Nyssa, Asteria. 2008. *Detektor jarak dengan sensor Ultrasonik berbasis Mikrokontroler*. Dosen jurusan Teknik Elektro-FTI Universitas Trisakti, JETri Vol. 7, Nmr 2.
- Smeltzer & Bare.(2001).*Keperawatan medical bedah*. Jakarta: EGC.
- Supriyadi, Nur. 2014. *Laporan Praktikum Mikroprosesor Modul II Seven Segment, Keypad dan Lcd*. FMIPA UNPAD.
- Tri Rahajoeningroem, Wahyudi. 2013. *SISTEM KEAMANAN RUMAH DENGAN MONITORING MENGGUNAKAN JARINGAN TELEPON SELULAR*. Teknik Komputer, Universitas Komputer, Universitas Komputer Indonesia.
- Wait, 2004. *Apa itu Emboli Paru-Paru: Gejala, Penyebab, Diagnosis, dan Cara Mengobati*. <https://www.docdoc.com/id/info/condition/emboli-paru>. Diakses tanggal 30 Agustus 2017.

### **Pemograman *Microcontroller* ATmega 328 pada *Arduino***

```
// pin LCD

#include<LiquidCrystal.h> // library

LiquidCrystal lcd (13,12,11,10,9,8); // set pin LCD

const float TRIG_PIN = 7;

const float ECHO_PIN = 6;

const int ledhijau = 5;

const int ledkuning = 4;

const int ledmerah = 3;

const int buzzer = 8;

void setup()

{

  lcd.begin(16, 2);

  // inisialisasi komunikasi serial:

  Serial.begin(9600);

  pinMode(TRIG_PIN,OUTPUT);

  pinMode(ECHO_PIN,INPUT);

  pinMode(ledmerah, OUTPUT);

  pinMode(ledkuning, OUTPUT);

  pinMode(ledhijau, OUTPUT);

  pinMode(buzzer, OUTPUT);

  lcd.begin(16, 2);
```

```

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Nama: CANON LEE");

    lcd.setCursor(0,1);

    lcd.print("NPM: 158120037");

    delay(400);

}

void loop()

{

float waktu, s, sisa;

float volume;

digitalWrite(TRIG_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIG_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG_PIN, LOW);

waktu = pulseIn(ECHO_PIN,HIGH);

// konversi waktu menjadi jarak

s = (34000 * waktu) / (1000000 * 2);

sisa = 16.97-s;//batas minimum cairan dari sensor - permukaan cairan dengan sensor

volume = 37.03 * sisa;//jumlah air dalam 1 cm x sisa

if (s <=6 && s >=4)

{

    lcd.setCursor(0,0);

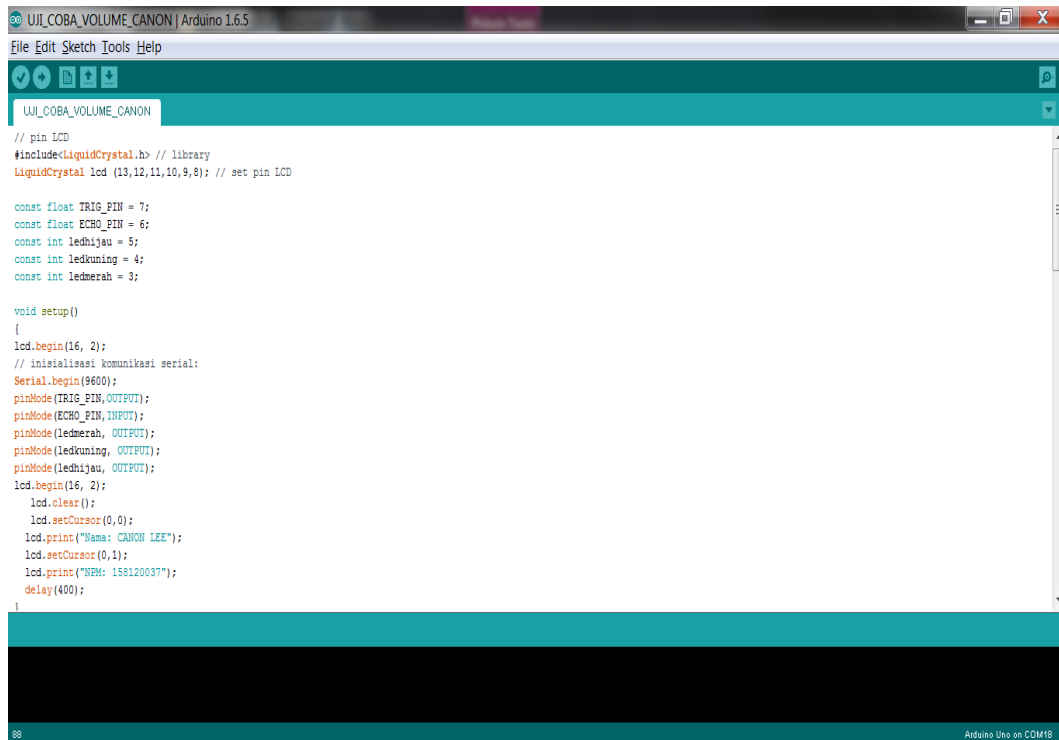
    lcd.print("h Cairan= ");

```

```
lcd.print(s);  
lcd.print("cm");  
  
lcd.setCursor(0,1);  
lcd.print("V Infus= ");  
lcd.print(volume);  
lcd.print("mL");  
digitalWrite(ledhijau, HIGH);  
digitalWrite(ledkuning,LOW);  
digitalWrite(ledmerah, LOW);  
digitalWrite(buzer, HIGH);  
delay(200);  
}  
else if(s >= 7 && s <=12){  
    lcd.setCursor(0,0);  
    lcd.print("h Cairan= ");  
    lcd.print(s);  
    lcd.print("cm");  
  
    lcd.setCursor(0,1);  
    lcd.print("V Infus= ");  
    lcd.print(volume);  
    lcd.print("mL");  
    digitalWrite(ledhijau, LOW);  
    digitalWrite(ledkuning, HIGH);  
    digitalWrite(ledmerah, LOW);
```

```
digitalWrite(buzer, HIGH);  
  
delay(200);  
  
}  
  
else if(s >= 13 && s <=17){  
  
    lcd.setCursor(0,0);  
  
    lcd.print("h Cairan= ");  
  
    lcd.print(s);  
  
    lcd.print("cm");  
  
  
    lcd.setCursor(0,1);  
  
    lcd.print("V Infus= ");  
  
    lcd.print(volume);  
  
    lcd.print("mL");  
  
    digitalWrite(ledhijau, LOW);  
  
    digitalWrite(ledkuning, LOW);  
  
    digitalWrite(ledmerah, HIGH);  
  
    digitalWrite(buzer, LOW);  
  
}  
  
delay(200);  
  
}
```

Gambar berikut ini menampilkan model layar jendela aplikasi untuk menuliskan program bahasa “C” di atas.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "UII\_COBA\_VOLUME\_CANON | Arduino 1.6.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for file operations and a compiler icon. The main text area shows the following code:

```
UII_COBA_VOLUME_CANON
// pin LCD
#include<LiquidCrystal.h> // library
LiquidCrystal lcd (13,12,11,10,9,8); // set pin LCD

const float TRIG_PIN = 7;
const float ECHO_PIN = 6;
const int ledhijau = 5;
const int ledkuning = 4;
const int ledmerah = 3;

void setup()
{
  lcd.begin(16, 2);
  // inisialisasi komunikasi serial:
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(ledmerah, OUTPUT);
  pinMode(ledkuning, OUTPUT);
  pinMode(ledhijau, OUTPUT);
  lcd.begin(16, 2);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Nama: CANON LEE");
  lcd.setCursor(0,1);
  lcd.print("NPM: 158120037");
  delay(400);
}
```

The status bar at the bottom indicates "88" on the left and "Arduino Uno on COM18" on the right.

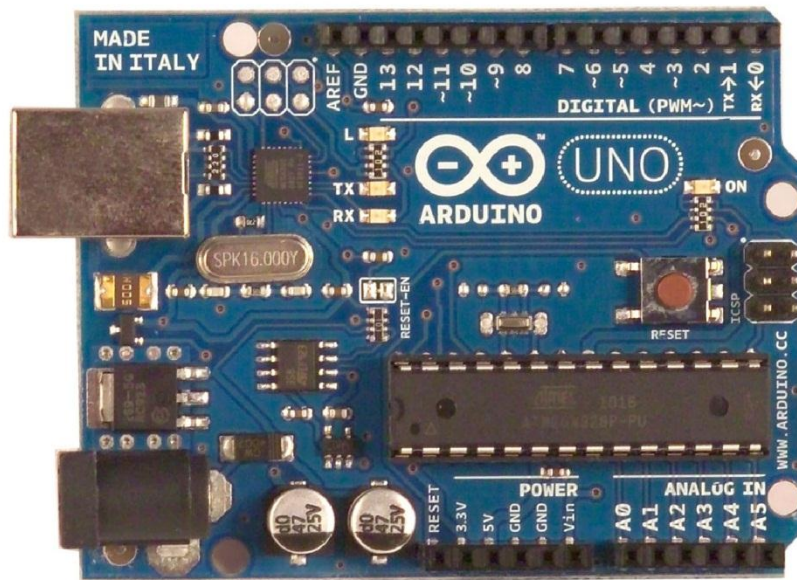
**Gambar : Jendela aplikasi penulisan program**

**Tabel Hasil Pengujian Volume dan Jarak.**

Jarak X	Volume L
4,06	477,95
5,01	443,32
6,88	374,06
7,01	369,04
8,14	326,86
9,01	294,13
10,05	256,36
11,27	211,03
12,02	165,07
13,02	146,19
14,11	105,91
15,03	71,91
16,93	1,41



# Arduino UNO



## Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

## Index

Technical Specifications

Page 2

How to use Arduino  
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies  
half sqm of green via Impatto Zero®

Page 7



*radiospares*

**RADIONICS**



# Technical Specification

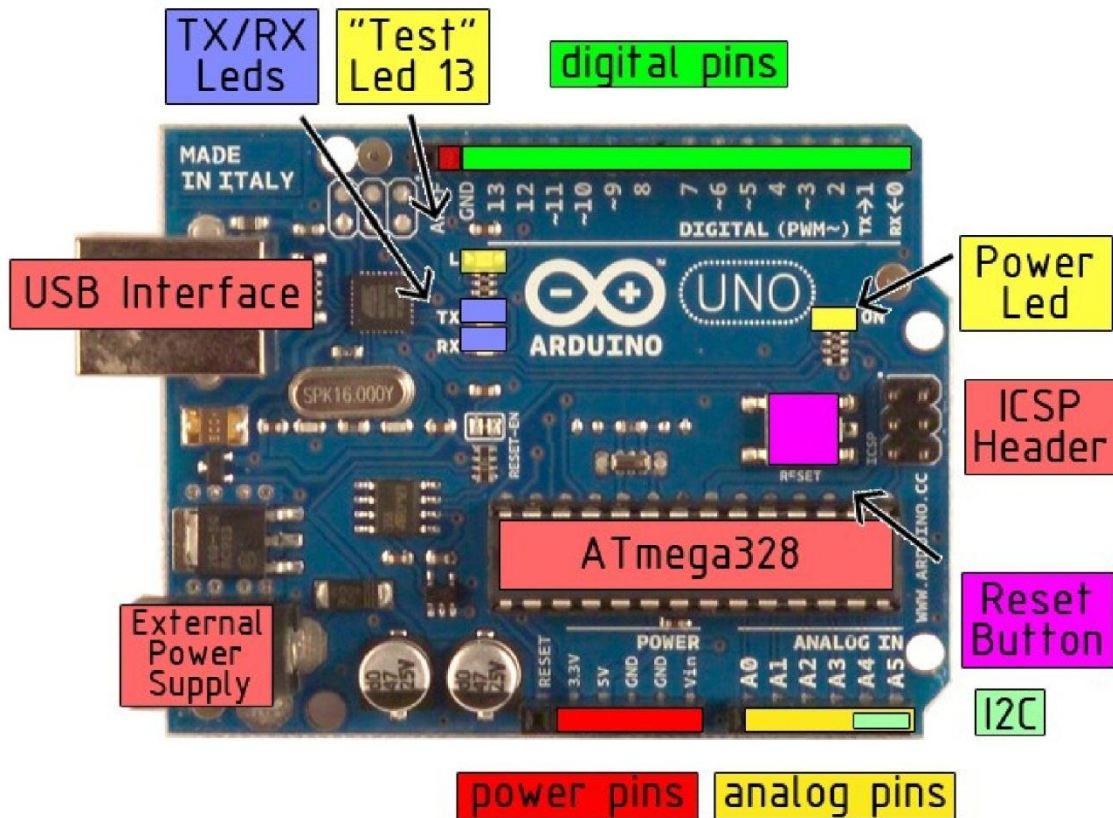


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

## the board



*radiospares*

*RADIONICS*



## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



**RADIOSPARES**

**RADIONICS**



The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I<sup>2</sup>C: 4 (SDA) and 5 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an \*.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

## Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



**radiospares**

**RADIONICS**



## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

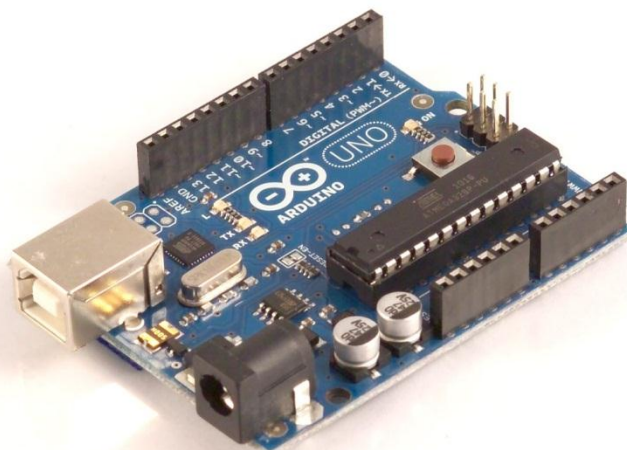
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



**radiospares**

**RADIONICS**



# How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

## Linux Install

## Windows Install

## Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

## Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>  
Arduino-0017>Examples>  
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Blink | Arduino 0017
File Edit Sketch Tools Help
[Icons]
Blink $
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```



Done compiling.

Press Compile button  
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

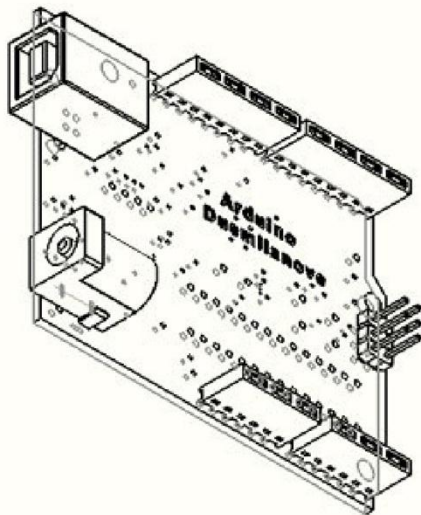
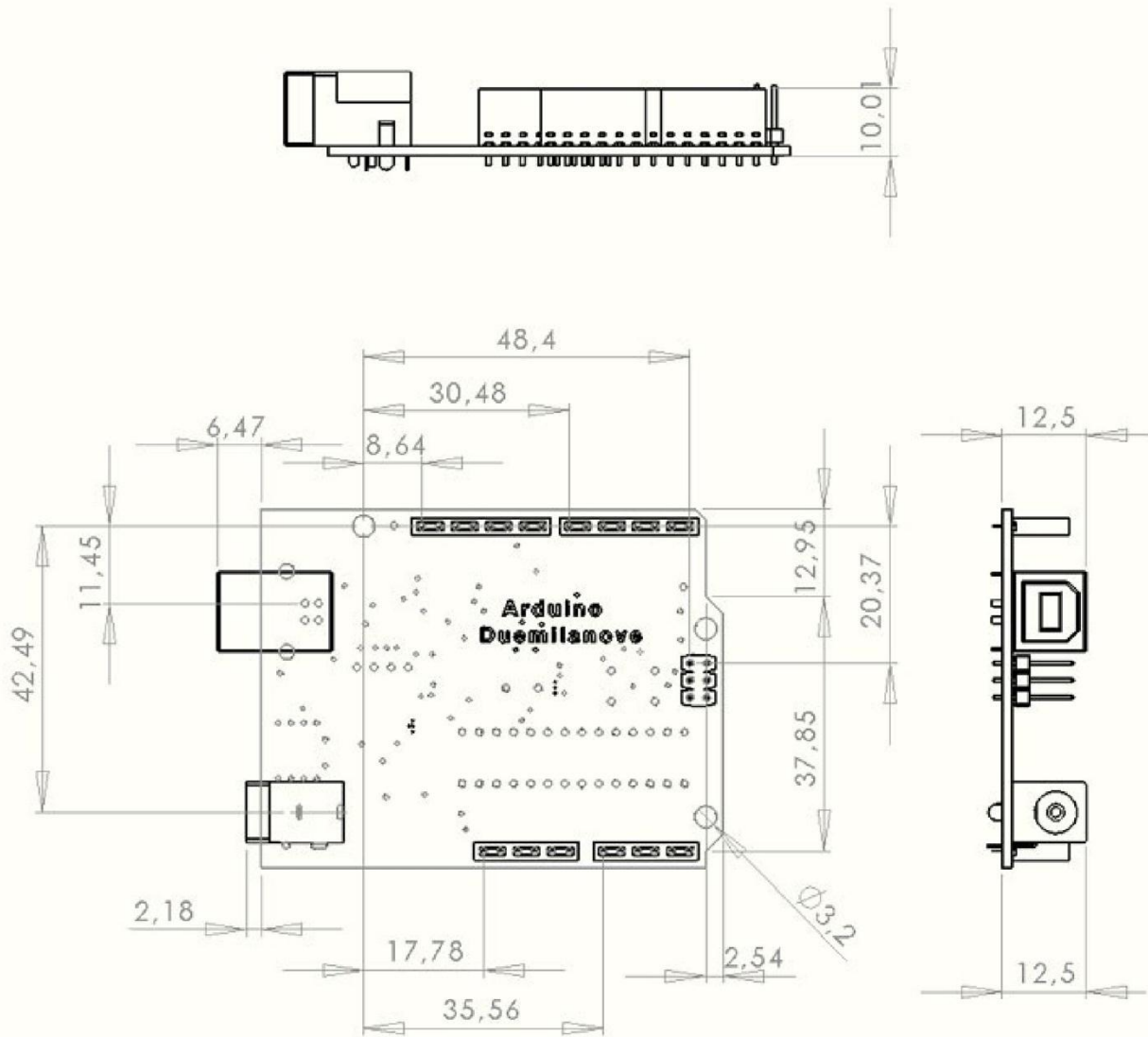


**radiospares**

**RADIONICS**



## Dimensioned Drawing



**radiospares**

**RADIONICS**



# Terms & Conditions



## 1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

## 2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

## 3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

## 4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



## Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



*radiospares*

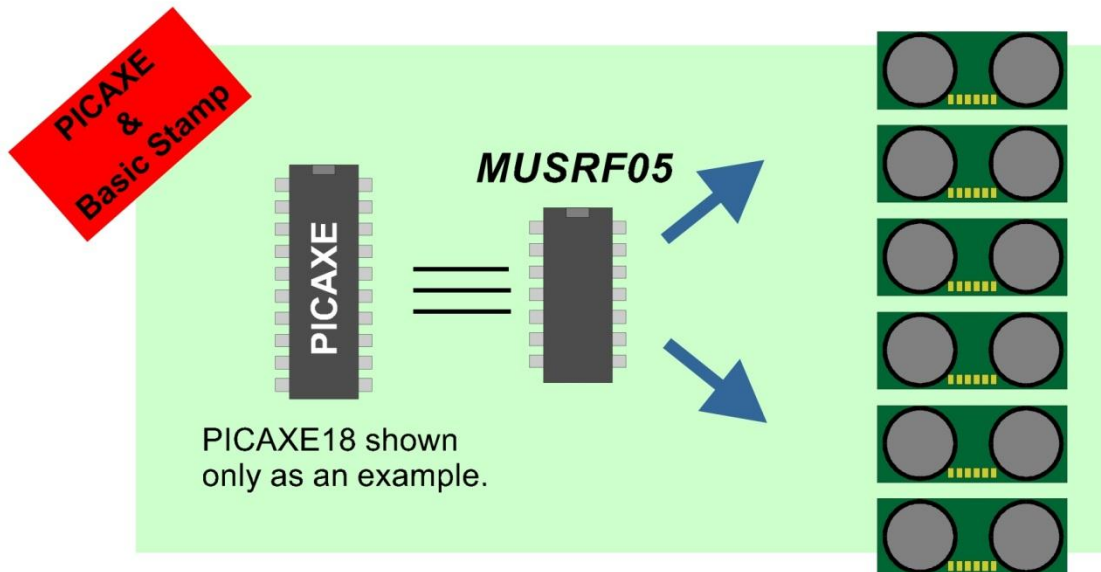
**RADIONICS**





## MUSRF05 Interface I.C.

This allows up to 6 of the superb Devantech SRF05 Ultrasonic range-finding modules to be controlled using a simple serial 2 wire connection from PICAXE or BASIC STAMP.



### Features.

- ✓ Simple serial control from PICAXE or BASIC STAMP.
- ✓ Simple serial data reading using Serin Command.
- ✓ Single or multiple sampling.
- ✓ Individual Control for each Ultrasonic module.
- ✓ Can return data in inches , centimetres or microseconds.
- ✓ Can return Minimum and Maximum values when Multi-Sampling.
- ✓ Uses PIC16F688 controller with calibrated 1% precision timer.
- ✓ Range up to 5 metres.
- ✓ Much lower overall cost than multiple I2C modules (SRF08/10).
- ✓ Saves significant code space in your PICAXE or Stamp.
- ✓ Saves pins if using multiple sensors.

### Overview.

The MUSRF05 controller allows the PICAXE or Stamp programmer to obtain range data from up to 6 Devantech SRF05 ultrasonic ranger-finder modules.

The command output from PICAXE is of the general syntax:-

```
Serout <output pin> , T2400 , ( <Command No.> , <No. of Samples> , <Units> )
```

Reading the returned data is of the general syntax:-

```
Serin <input pin> , T2400 , <Range> , <Max> , <Min>
```

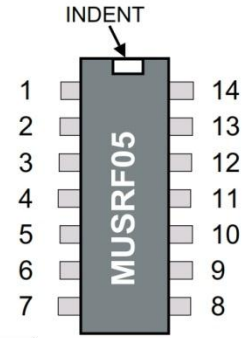
See overleaf for more information on data sizes and 'real world' formatting.

***Our website includes a small PICAXE Basic example showing a simple example.***

### Pin Numbers.

If you look closely at the I.C. case you will see an 'indent' at one end.

Pin numbers are referenced to that indent.



Pin Number	Description	Function	Notes
1	+ve	+5V Power Supply	1
2	Select Format	ASCII / Terminal	2
3	US Power	O/P Power Switch	3
4	Reset	Reset MUSRF05	4
5	SIN	Serial in (+2K2 res.)	5
6	SOUT	Serial Out (+2K2 res.)	5
7	Indicate	LED Indicator	6
8	US6	I/O SRF05 Module 6	7
9	US5	I/O SRF05 Module 5	7
10	US4	I/O SRF05 Module 4	7
11	US3	I/O SRF05 Module 3	7
12	US2	I/O SRF05 Module 2	7
13	US1	I/O SRF05 Module 1	7
14	Gnd	0v Power Ground	1

### NOTES. (Please READ them, they are for your information.)

- The power supply must be a well regulated 5V shared with the Devantech SRF05 Ultrasonic Module(s). This can also be shared with the PICAXE or Stamp 5V power supply. We recommend decoupling capacitors (200nF to 10uF) are placed across the +V/Gnd supply pins.
- Connect this to +V for normal use. If you want serial output to a PC Terminal type application then connect this pin to Ground (0V). [ Terminal Format:  $\mu$ seconds <space> Av cm <space> Min cm <space> Max cm. ] In normal use you can send ASCII values to your PC to be read from serial buffer by your own application.
- This is an **inverted** output intended to switch a P-Chan MOSFET or PNP transistor.  
**CAUTION:** This is **NOT** a direct output power supply. **Direct connection to a load may cause damage!!**
- Resets the MUSRF05 programme. Can be connected via a push-button switch to Gnd (0V). Can also be remotely reset by PICAXE / Stamp using the transistor circuit as shown in example.
- Serial IN is the connection from PICAXE / Stamp output pin. **N.B. Connect using 2K2 resistor.** Serial OUT is the return to PICAXE / Stamp input pin. **N.B. Connect using 2K2 resistor.**
- LED indicator. Connect via 2K2 for Green LEDs or a 4K7 for Red LEDs.
- These are the I/O lines to the Devantech SRF05 Ultrasonic modules. (NOT any other types). These can be connected directly to our SRFMate pcb. Or, if you connect MUSRF05 directly to the SRF05 module, you must make sure that the SRF05 module 'Mode' pin is Grounded.



SRFMate makes connection easier and also allows 'daisy-chaining' of 5V/Gnd power supply.

## MUSR05 Command Details and Data Size.

<Value> Units	Returned Data Sizes		
	<Range>	<Max>	<Min>
<1> Inch	Byte	Byte	Byte
<2> CM	Word	Word	Word
<3> $\mu$ Seconds	Word	Word	Word

### PICAXE Examples:

#### ' Simplest Command in INCHES.

```
Serout 7 , T2400 , (1 , 1 , 1)
Serin 1 , T2400 , b0
```

```
' Tell U/S Module 1 to get a single sample.
' Get result in inches and store in b0
```

#### ' Multi-Sample in Inches

```
Serout 7 , T2400 , (1 , 5 , 1)
Serin 1 , T2400 , b0 , b1 , b2
```

```
' Tell U/S Module 1 to get 5 samples. Return in Inches.
' Store Range in b0 , Max val in b1 and Min Val in b2
```

#### ' Simplest Command in Centimetres.

```
Serout 7 , T2400 , (1 , 1 , 2)
Serin 1 , T2400 , w0
```

```
' Tell U/S Module 1 to get a single sample.
' Get result in inches and store in b0
```

#### ' Multi-Sample in Centimetres

```
Serout 7 , T2400 , (1 , 5 , 2)
Serin 1 , T2400 , w0 , w1 , w2
```

```
' Tell U/S Module 1 to get 5 samples. Return in Inches.
' Store Range in w0 , Max Val in w1 and Min Val in w2
```

#### ' Simplest Command in Microseconds.

```
Serout 7 , T2400 , (1 , 1 , 3)
Serin 1 , T2400 , w0
```

```
' Tell U/S Module 1 to get a single sample.
' Get result in inches and store in b0
```

#### ' Multi-Sample in Microseconds

```
Serout 7 , T2400 , (1 , 5 , 3)
Serin 1 , T2400 , w0 , w1 , w2
```

```
' Tell U/S Module 1 to get 5 samples. Return in Inches.
' Store Range in w0 , Max Val in w1 and Min Val in w2
```

### Note:

If you choose Multi-Samples the max/min samples are optional, so if you only require an averaged value then omit the second and third parameters. Example:-

#### ' Multi-Sample in Inches

```
Serout 7 , T2400 , (1 , 5 , 1)
Serin 1 , T2400 , b0
```

```
' Tell U/S Module 1 to get 5 samples. Return in Inches.
' Store Range in b0 . Ignore max and min bytes.
```

### IMPORTANT PICAXE NOTE.

PICAXE is quite slow. In the above examples (shown using w0 , w1 and w2) you may have to replace:

w0 with b0,b1      w1 with b2,b3      w2 with b4,b5  
[ e.g. Serin 1 , T2400 , b0 , b1 , b2 , b3 , b4 , b5 ]

### Why have Multiple samples?

Taking an average is the most precise method, by knowing the 'spread' of results you can determine the **quality** of the measurement. But remember, the more samples you request then the longer the measurement takes.

### Results.

The returned values are obviously integer values and therefore have a precision of 1 unit. So, for example, a real internal measurement of 1.5 inches will be returned as 1 inch and 1.5cm will be returned as 1cm. For greater precision you may choose to get the results in microseconds and do your own maths. Precision: +/- 1 unit.

## Command Details.

### Ultrasonic Module Control.

**Syntax:** Serout <output pin> , T2400 , ( <Command Number> , <No. of Samples> , <Units> )

#### Command Number:

1 to 6

#### Action

Trigger Ultrasonic Module 1 to 6

10

Switch ON power to U/S Modules

20

Switch OFF power to U/S Modules

#### No. of Samples

1 to 20

Return averaged range value and Max and Min values

#### Units

1

Inches

2

Centimetres

3

MicroSeconds

### Getting Started.

If you make a circuit which uses the Power Control functions (as in our example overleaf) your first programme step before getting range data is to Switch ON the power. Note: When you download a new programme to your PICAXE it will send a bit of nonsense to the output pins.

So to initialise we would recommend you have code a bit like this:-

(Pause 1000 - optional at start if you experience problems)

Serout <o/pin> , T2400 , (255):pause 1000 ' Clears MUSRF05 buffer

Serout <o/pin> , T2400 , (10): pause 1000 ' Switches U/S power ON

.... now you can do the code for your range measurements.

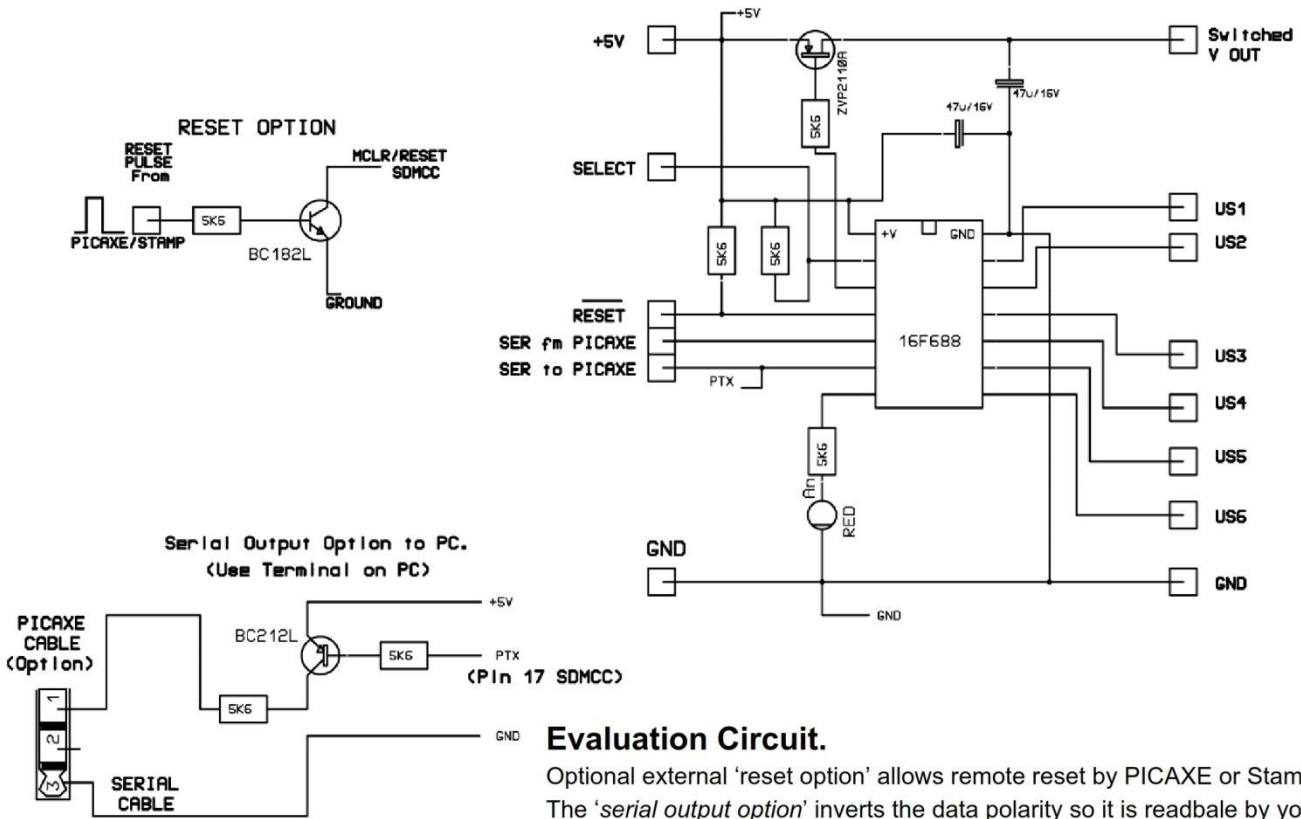
### General.

It is well worth a look at Devantech's website at <http://www.robot-electronics.co.uk/> to read up about the modules and their performance.

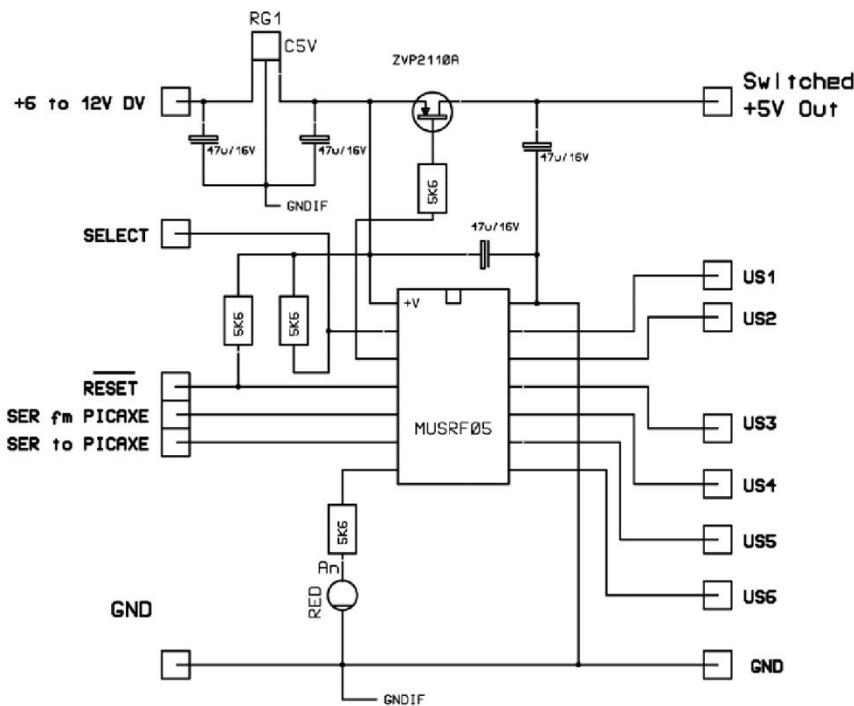
It is also a good idea to have a look at the beam patterns of ultra-sonic modules. As you will see the 'beam' pattern is quite a spread and is definitely NOT a pencil beam. Understanding this will hopefully explain why you sometimes get odd results with ultrasonic range-finding.

Ultrasonic can also bounce around a bit and the longer the range then the more likely you are to get anomalous results - hence our option for multi-sampling. Multi-sampling won't cure odd echo effects but the Min-Max 'spread' can be used to get an idea of the level of uncertainty.

### Evaluation pcb for SRF05 Ultrasonic Module



### Interface pcb for SRF05 Ultrasonic Module



Both Serial connections (to & from Basic Stamp / PICAXE) must use 2K2 resistors to limit current.

## Further Information.

### PICAXE or Basic Stamp Electrical Connections.

1. You must have a commoned ground (0V) between the MUSRF05 ic and your PICAXE / Basic Stamp.
2. Both serial connections (between PICAXE / Basic Stamp and MUSRF05 ic) must use 2K2 resistors.

### Ultrasonic Transducer Ranges.

The ranges determined by the SRF05 module depend on the size and material of the target.

We have found that a 175mm square plywood target can be detected accurately up to 2.5m. Beyond that the figures are not too reliable.

However, a 1m square target was accurately ranged at up to 4 metres. Larger targets were ranged at over 5 metres  
The targets used were plywood and positioned along the main axis of the Ultrasonic module.

### The Ultrasonic 'Beam'.

The output from a 40kHz transducer is NOT a 'pencil' beam. This diagram shows a typical output spread.

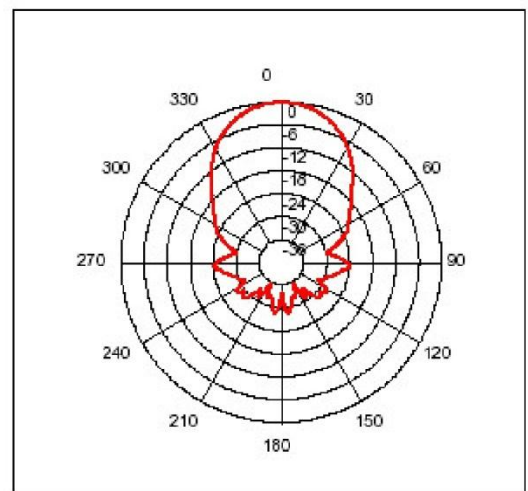
You should realise that the physical positioning of the transducer / module is very important.

After all, if you position it too close to the ground then you will get the reflection from the ground rather than the intended target.

This becomes more important as the target distance increases.

Full Spec: <http://www.robot-electronics.co.uk/datasheets/t400s16.pdf>

It is possible to make short tubes and lined with a soft material to adjust the pattern. You may choose to experiment, but remember, you will probably affect the maximum range. Do NOT use metal.



Beam Spread @ 40kHz (-6dB)

### MUSRF05 Electrical Specifications (Typical).

<b>Power</b>	5V DC (Variations from this may affect oscillator frequency and accuracy)
<b>I (Supply)</b>	1mA average (IC only. Each SRF05 takes about 4mA on average)
<b>Boot Up Time</b>	About 500mS after power-on / reset.
<b>Sample Delay</b>	Depends on range. You must include a 50mS Pause between trigger requests.
<b>Osc. Accuracy</b>	1% Factory Calibrated (Microchip spec)
<b>Range Accuracy</b>	+ / - 1 inch or +/- 10 microseconds.
<b>Serial Polarity IN</b>	PICAXE T2400 . All others 2400 baud Inverted.
<b>Serial Polarity OUT</b>	As above.
<b>Serial Line load</b>	Both serial connections must use a 2K2 resistor in-line to limit current.



P.O. Box 2509 Wareham Dorset BH20 6YH U.K.  
Tel: +44 (0)1929 405388 Fax: 448719941420  
E-Mail: [MUSRF05@fgcvme.co.uk](mailto:MUSRF05@fgcvme.co.uk)