

LEMBARAN PENGESAHAN

OAN_ARC ROBOT BERODA EXPERT BATTLE PEMADAM API

TUGAS AKHIR

Oleh:

Johanes Sitohang 3210701031

Ari Teja Kusuma 3210701032

Hendy Singkop P. Hasibuan 3210701036

**Diajukan dan disahkan sebagai laporan Tugas Akhir
di Program Studi Teknik Elektro Politeknik Batam**

Batam, 25 Juni 2010

Pembimbing,

Hendawan Soebhakti, ST.

NIK : 104031

OAN_ARC ROBOT BERODA EXPERT BATTLE PEMADAM API

TUGAS AKHIR

Oleh:

Johanes Sitohang 3210701031

Ari Teja Kusuma 3210701032

Hendy Singkop P. Hasibuan 3210701036

Disusun untuk memenuhi syarat kelulusan Program Diploma III

Program Studi Teknik Elektro

Politeknik Batam



PROGRAM STUDI TEKNIK ELEKTRO

POLITEKNIK BATAM

BATAM

2010

KATA PENGANTAR

Rasa syukur yang tidak terhingga penulis haturkan kepada Tuhan Yang Maha Esa, atas limpahan karunia dan ilmu sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan baik yang tidak terlepas dari bantuan dosen dan rekan-rekan mahasiswa Politeknik Batam.

Penulisan Laporan Tugas Akhir ini dimaksudkan untuk melengkapi persyaratan kelulusan tingkat Diploma III Program Studi Teknik Elektro Politeknik Batam. Untuk memenuhi persyaratan tersebut maka penulis mencoba untuk mengaplikasikan sebuah sistem dalam bentuk robot yang dikontrol secara otomatis. Fungsi dari robot yang dibuat untuk memadamkan api pada sebuah ruangan yang berbentuk labirin.

Penulisan Laporan Tugas Akhir ini dapat disusun dan diselesaikan dengan baik tidak terlepas dari bantuan dan dukungan dari semua pihak yang ikut dalam membimbing penulis untuk menyelesaikan Laporan Tugas Akhir ini. Pada kesempatan ini, penulis mengucapkan terkma kasih kepada:

1. Tuhan Yang Maha Esa, atas anugerah yang telah diberikan kepada penulis.
2. Kedua orang tua dan keluarga atas jasa Doa, bimbingan dan nasehat.
3. Bapak Dr. Priyono Eko Sanyoto, selaku Direktur Politeknik Batam.
4. Bapak Iman Fahruzi, ST. selaku Ka. Prodi Teknik Elektro Politeknik Batam.

5. Bapak Hendawan Soebakti, ST. selaku dosen pembimbing KRCI Expert Battle dan juga dosen pembimbing Tugas Akhir.
6. Bapak Ahmad Riyad Firdaus, MT. selaku dosen pengampu Tugas Akhir.
7. Seluruh Dosen-dosen Teknik Elektro Politeknik Batam.
8. Seluruh Tim Robot Politeknik Batam yang membantu dalam riset.
9. Seluruh Alumni Tim Robot Politeknik Batam yang ikut membimbing dalam proses pembuatan dan Tugas Akhir.
10. Seluruh teman-teman yang telah membantu atas terselesaikannya buku laporan ini.

Semoga amal dan ibadah diterima di sisi Tuhan yang Maha Esa, atas bantuan moril maupun spiritual demi terselesaikannya laporan ini. Penulis sadar masih banyak kekurangan yang terdapat pada Laporan Tugas Akhir ini, oleh karena itu saran dan kritik yang membangun penulis harapkan dari semua kalangan.

Batam,

Penulis

ABSTRAK

Robot cerdas beroda Expert Battle Oan_Arc, merupakan robot yang dirancang untuk memadamkan api/lilin serta mengambil boneka yang terdapat di dalam sebuah ruangan yang berbentuk labirin. Robot ini dirancang dari tiga bagian utama, yakni bagian mekanik, elektronik dan program. Ketiga bagian itu menjadi satu kesatuan untuk menjalankan fungsinya yaitu memadamkan api/lilin dan mengambil sebuah benda.

Design dari robot ini berbentuk seperti mobil kontraktor, dengan sepasang penjepit dibagian depan robot. Penggerak robot ini menggunakan motor DC dan servo. Dua roda masing-masing mempunyai satu motor DC. Juga dipasang dua servo yang berfungsi untuk sendi lengan penjepit boneka dan poros perputaran dudukan kipas pemadam. Sedangkan untuk memadamkan api digunakan kipas yang terpasang di sebuah motor DC 12 V. Robot yang dirancang ini menggunakan berbagai sensor yang berfungsi untuk melakukan pengeksekusian terhadap tiap-tiap ruangan. Sensor-sensor yang digunakan yaitu SRF untuk mendeteksi jarak dinding/objek, *Magnetic Compass* untuk mendeteksi arah posisi, *Linetrack Analog* untuk mendeteksi perbedaan warna lantai.

Sebagai otak dari robot ini untuk mengatur kinerja semua sensor adalah Mikrokontroler ATmega 8535. Rangkaian ini digunakan sebagai otak dari robot yang penulis buat.

ABSTRACT

A wheeler robot Expert Battle Oan_Arc, is a robot designed to extinguish the fire/candle and take the doll contained in a space that is shaped maze. This robot is designed from the three main parts, namely the mechanical, electronics and programming. All three parts into a single entity to perform its functions to extinguish the fire/candle and retrieve an object.

The design of the robot is shaped like cars contractor, with a pair of pincers at the front of the robot. This robot move by DC motors and servo. Two wheels of each has a DC motor. Also installed two servo that serves to clamp the arm joints and shaft rotation doll fan extinguisher holder. While the fan used to extinguish fires that installed in a motor DC 12 V. These robots are designed using various sensors that function to do the execution of each room. Sensors used to detect the distance of the SRF wall/object, Magnetic Compass for detecting the direction of the position, Line Track Analog to detect differences in color of the floor.

As the brain of this robot is to regulate the performance of all sensors is Microcontroller ATmega 8535. This circuit is used as the brain of the robot that authors create.

DAFTAR ISI

KATA PENGANTAR	i
ABSTRAK.....	iii
DAFTAR ISI	v
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
Bab I Pendahuluan.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan dan Manfaat.....	2
1.3 Rumusan Masalah.....	2
1.4 Batasan Masalah	2
1.5 Metode Penulisan	3
1.6 Sistematika Penulisan.....	4
Bab II Ikhtisar Sistem	5
2.1 Deskripsi Umum	5
2.2 Karakteristik.....	7
2.3 Lingkungan Operasi Pengembangan	8
Bab III Landasan Teori.....	9
3.1 Mikrokontroler.....	9
3.2 Sensor ultrasonik	12
3.3 Sensor Arah	13
3.4 Sensor Warna.....	14

3.5 Motor Servo.....	15
3.6 Driver Motor Kipas.....	16
3.7 Driver Motor DC Roda	17
3.8 Baterai.....	18
Bab IV Perancangan Sistem.....	20
4.1 Perancangan Elektronik	20
4.1.1 Power Supply.....	20
4.1.2 Minimum Sistem Mikrokontroler ATmega 8535.....	20
4.1.3 Sensor Jarak.....	23
4.1.4 Sensor Arah.....	23
4.1.5 Sensor Warna.....	24
4.1.6 Motor Servo.....	24
4.1.7 Driver Motor Kipas.....	26
4.1.8 Driver Motor Penggerak Roda.....	27
4.2 Perancangan Program.....	28
4.2.1 flowchart Pemograman.....	29
Bab V PENGUKURAN, PENGUJIAN DAN ANALISA SISTEM.....	31
5.1 Pengukuran.....	31
5.1.1 Pengukuran Tegangan Masukan dan Keluaran Power Supply...31	
5.1.2 Pengukuran Sensor Jarak.....	32
5.1.3 Pengukuran Sensor Arah.....	35
5.1.4 Pengukuran Sensor Warna.....	36
5.1.5 Pengukuran Data Motor Servo.....	36

5.1.6 Pengukuran Keluaran Driver Motor Kipas.....	38
5.1.7 Pengukuran Keluaran Driver Motor DC 180 rpm	39
5.2 Pengujian	40
5.3 Analisa Sistem	41
5.3.1 Power Supply.....	41
5.3.2 Sensor Jarak	42
5.3.3 Sensor Warna	44
5.3.4 Motor Servo	44
5.3.5 Driver Motor Kipas	45
5.3.6 Driver Motor DC.....	45
Bab VI Kesimpulan dan Saran.....	46
6.1 Kesimpulan.....	46
6.7 Saran.....	47
Daftar Pustaka.....	48
Halaman Lampiran	49

DAFTAR TABEL

Tabel. 3.1	Konfigurasi Pin Port.....	11
Tabel 3.2	Kondisi motor.....	18
Tabel 4.1	Pengukuran diver kipas.....	26
Tabel 5.1	Hasil Pengukuran Power Supply dengan Regulator.....	31
Tabel 5.2	Pengukuran Keluaran Sensor Jarak SRF.....	33
Tabel 5.3	Hasil Pengujian Sensor Arah CMPS03.....	35
Tabel 5.4	Hasil Pengukuran Sensor Garis / Warna.....	36
Tabel 5.5	Hasil Pengukuran Keluaran Motor Servo.....	37
Tabel 5.6	Hasil Pengukuran Driver Motor Kipas.....	38
Tabel 5.7	Tabel pengukuran driver motor.....	39
Tabel 5.8	Pengujian robot dalam satuan waktu.....	40
Tabel 5.9	Hasil Pengukuran sensor jarak SRF04.....	43

DAFTAR GAMBAR

Gambar 2.1	Diagram blok proses input dan output.....	5
Gambar 3.1	Konfigurasi pin ATmega 8535	10
Gambar 3.2	Sensor ultrasonik	12
Gambar 3.3	CMPS03 <i>Magnetic Compass</i>	13
Gambar 3.4	<i>Communication Protocol</i>	14
Gambar 3.5	Pantulan cahaya LED yang Ditangkap oleh Photodiode	14
Gambar 3.6	Bentuk dari Motor Servo.....	16
Gambar 3.7	Transistor TIP122.....	17
Gambar 3.8	Bentuk IC L298.....	18
Gambar 3.9	Baterai 2700 mAh (krisbow).....	19
Gambar 4.1	Power Supply dengan Regulator.....	20
Gambar 4.2	Rangkaian Minimum sistem mikrokontroler.....	22
Gambar 4.3	Rangkaian Sensor jarak.....	23
Gambar 4.4	Rangkaian Sensor arah.....	23
Gambar 4.5	Rangkaian Sensor Warna.....	24
Gambar 4.6	Rangkaian Motor Servo.....	24
Gambar 4.7	Lebar pulsa yang digunakan untuk mengontrol servo.....	25
Gambar 4.8	Pengendalian sudut.....	25
Gambar 4.9	Driver Kipas terhubung ke Mikrokontroler.....	26
Gambar 4.10	Skematik dasar driver motor DC.....	27
Gambar 4.11	Lapangan pertandingan	28

Gambar 4.12 Flow Chart.....	29
Gambar 5.1 Titik ukur power supply.....	31
Gambar 5.2 Pengukuran Sinyal Frekuensi dari Sensor Jarak.....	32
Gambar 5.3 Rangkaian skematik <i>Magnetic Compass</i> CMPS03.....	35
Gambar 5.4 Pengukuran atau Pengambilan Arah Data <i>Compass</i>	36
Gambar 5.5 Pengukuran Keluaran dari sensor warna.....	36
Gambar 5.6 Pengukuran Motor Servo.....	36
Gambar 5.7 Rangkaian <i>driver</i> Motor Kipas.....	38
Gambar 5.8 Rangkaian <i>driver</i> motor.....	39
Gambar 5.9 Lapangan Pertandingan.....	40
Gambar 5.10 Grafik pengukuran sensor jarak SRF04.....	43

BAB I

PENDAHULUAN

1.1 Latar belakang

Indonesia adalah salah satu Negara yang setiap tahun selalu mengadakan kontes robot. Kontes robot itu sendiri bertujuan agar generasi pemuda Indonesia khususnya kalangan mahasiswa dapat mengaplikasikan mata kuliah yang mereka dapat dari dosen pengajar. Kontes robot ini memiliki beberapa kategori, diantaranya Kontes Robot Indonesia (KRI), Kontes Robot Cerdas Indonesia (KRCI), dan yang saat ini Indonesia juga memiliki kategori baru yaitu Kontes Robot Seni Indonesia (KRSI).

Pada KRCI 2009 penulis mengikuti salah satu kategori lomba yaitu expert battle. Pada kategori ini sebuah robot bertugas melakukan pemadaman api dan mengambil sebuah boneka dalam ruangan yang sudah ditentukan, selain itu juga robot harus mampu melewati rintangan berupa polisi tidur dan kembali ke *home*.

Pada penelitian ini akan dibahas proses pembuatan robot yang dapat berfungsi sesuai dengan aturan pertandingan KRCI EXPERT BATTLE tahun 2009.

1.2 Tujuan dan Manfaat

Tujuan Pembuatan dan Perancangan Tugas Akhir:

1. Membuat sebuah robot yang berfungsi sesuai dengan aturan pertandingan KRCI EXPERT BATTLE 2009

Manfaat Pembuatan dan Perancangan Tugas Akhir :

1. Dapat merancang sebuah robot yang dapat diikutsertakan dalam KRCI 2009

1.3 Rumusan Masalah

1. Bagaimana merancang desain robot yang memenuhi aturan pertandingan KRCI EXPERT BATTLE 2009.
2. Bagaimana pemrograman agar robot dapat berfungsi pada saat dilapangan secara baik.

1.4 Batasan Masalah

Untuk memberikan penekanan khusus agar isi dari tugas akhir ini tidak terlalu melebar, maka penulis membatasi permasalahan yang dibahas. Adapun batasan masalahnya adalah sebagai berikut:

1. Robot yang dibuat hanya dapat digunakan dalam pertandingan KRCI 2009
2. Pemrograman robot menggunakan bahasa C dengan Compiler CodeVision AVR.

1.5 Metode Penulisan

Dalam pembuatan laporan ini, penulis mendapatkan data dan informasi dari dosen pembimbing tugas akhir, dosen-dosen teknik elektro, *internet*, serta buku-buku atau media lainnya yang menyangkut tentang materi yang dikerjakan.

Sedangkan untuk mendapatkan data dari alat yang telah penulis kerjakan, penulis menggunakan metode pengujian dan pengamatan, dengan cara melakukan pengambilan data melalui alat ukur, pengetesan dengan program dan lain-lain. Dan hasil akhir dari pengujian, penulis melakukan perbandingan antara teori dan analisa praktek.

1.6 Sistematika Penulisan

Sistematika penulisan merupakan bagian dari laporan yang mempunyai tujuan untuk mempermudah pembaca terhadap pemahaman pembaca terhadap isi yang terkandung didalamnya. Hal ini dibuat untuk menghindari kesalahan penafsiran.

Penulisan laporan ini dikelompokkan menjadi beberapa bagian antara lain:

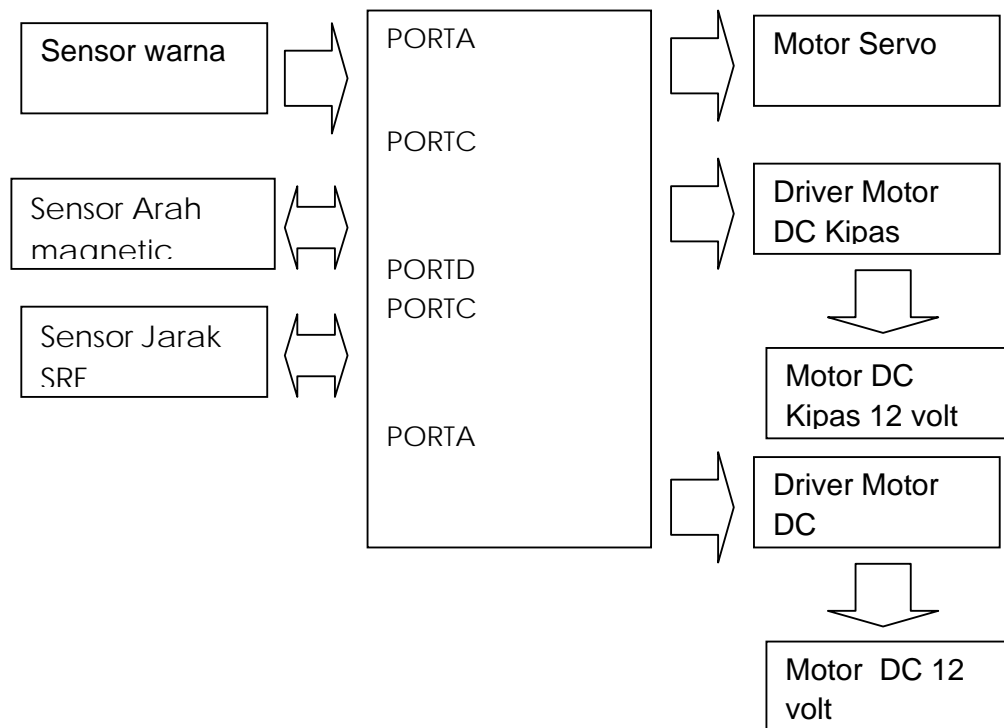
- BAB I Pendahuluan berisikan latar belakang, tujuan dan manfaat tugas akhir, rumusan masalah, batasan masalah, dan sistematika penulisan.
- BAB II Ikhtisar Sistem berisikan deskripsi umum, karakteristik dan lingkungan operasi dan pengembangan.
- BAB III Landasan Teori menerangkan teori dasar yang digunakan.
- BAB IV Perancangan Sistem menggambarkan desain perangkat lunak maupun keras (*hardware*) secara lengkap.
- BAB V Pengujian dan Analisa Sistem
Menjelaskan tentang cara pengujian dan menganalisa sistem.
- BAB VI Kesimpulan dan Saran merupakan ringkasan dari hasil analisa yang di buat.

BAB II

IKHTISAR SISTEM

2.1 Deskripsi Umum

Robot Cerdas Expert Battle Oan_Arc merupakan robot yang dirancang untuk melakukan tugas mencari dan memadamkan lilin/api dan mengambil boneka . Robot ini merupakan kombinasi dari tiga bagian yaitu mekanik, elektronik dan optimasi program.



Gambar 2.1 Diagram Blok Proses *Input* dan *Output*

Robot ini menggunakan berbagai *input* dan *output* berupa sensor dan motor yang dikendalikan dengan menggunakan IC Mikrokontroler ATmega 8535. Semua sensor diaktifkan dengan program yang telah dirancang untuk mengikuti alur yang diinginkan sesuai dengan strategi yang dibutuhkan. Setelah program didownload robot akan bergerak mencari lilin/api secara otomatis pada labirin-labirin dan juga boneka.

Adapun fungsi dari sensor-sensor tersebut:

- Sensor warna (*Line Track*) analog, sebagai alat mendeteksi warna lantai lapangan atau garis lingkaran letak dari api/lilin.
- Sensor arah (*Magnetic Compass*), sebagai alat untuk mensinkronisasi jalannya robot agar bergerak lurus dan berputar sesuai derajat yang telah ditentukan.
- Sensor jarak (SRF), sebagai alat mendeteksi jarak tiap ruangan sehingga robot tidak menabrak sekat-sekat yang ada pada tiap ruangan.
- Motor Servo, sebagai alat untuk menjepit boneka dan penggerak leher tiang kipas pemadam api.
- Driver motor DC kipas, sebagai rangkaian sistematis untuk mengontrol kecepatan pemadaman lilin/api oleh kipas.
- Motor DC kipas 12 volt, sebagai alat untuk memadamkan lilin/api.

- Driver motor DC, sebagai rangkaian sistematis untuk mengontrol motor penggerak roda robot agar dapat maju, mundur, dan berputar.
- Motor DC 12 volt, sebagai alat penggerak roda robot.

2.2 Karakteristik

Perancangan pada Robot Cerdas Expert Battle Oan_arc memiliki beberapa karakteristik diantaranya adalah sebagai berikut:

- Robot ini menggunakan satu buah IC AT mega 8535.
- Robot ini dilengkapi dengan 3 buah motor servo sebagai penjepit dan penggerak untuk mengambil dan geleng memadamkan api.
- Robot ini dilengkapi 2 buah motor DC 185rpm sebagai penggerak roda dan motor DC 12 volt sebagai kipas untuk memadamkan api.
- Robot ini dilengkapi sensor jarak SRF dengan jarak deteksi maksimum 3 meter.
- Robot ini dilengkapi 1 buah sensor warna untuk membedakan antara warna gelap dan warna terang dari lantai.
- Robot ini dilengkapi sensor arah *Magnetic Compass* 1 buah dengan radius arah 360°.
- Dimensi robot adalah panjang 28 cm, lebar 23 cm, dan tinggi 18 cm.

- Robot ini menggunakan plat berukuran 1.5 mm untuk badan dan lengan penjepit robot.
- Sumber tegangan untuk robot ini menggunakan baterai krissbow 13 buah dengan masing-masing tegangan baterai 1.3 volt dan arus 2.77 amper.

2.3 Lingkungan Operasi Pengembangan

Lingkungan operasi robot pengambilan boneka dan pemadaman api ini dapat diaplikasikan pada Kontes Robot Cerdas Indonesia yang merupakan kontes tahunan yang diselenggarakan oleh pemerintah Indonesia.

Kedepannya dapat diaplikasikan ke dalam bentuk realita sehingga untuk pengambilan suatu benda atau pemadaman api di dalam sebuah ruangan, yang tidak dapat terjangkau oleh manusia dapat dilakukan oleh robot.

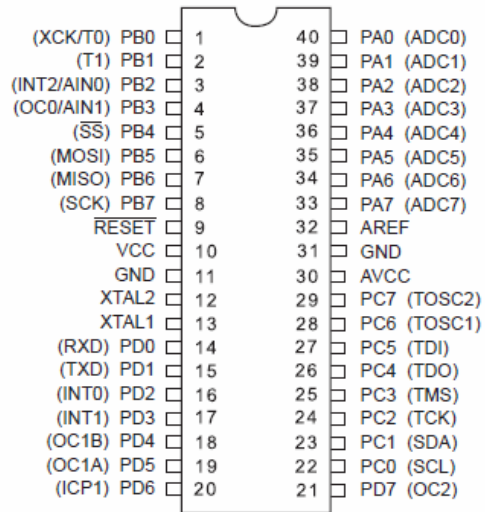
BAB III

LANDASAN TEORI

3.1 Mikrokontroler

Mikrokontroler merupakan sebuah chip yang mempunyai input dan output serta dikendalikan dengan program yang bisa ditulis dan dihapus dengan cara khusus. Dalam robot ini digunakan Mikrokontroler Atmega 8535 yang merupakan keluarga dari AVR Mikrokontroler dengan fasilitas sebagai berikut :

- Frekuensi clock maksimum 16 MHz.
- Jalur I/O 32 buah, yang terbagi dalam PortA, PortB, PortC dan PortD.
- Analog to Digital Converter 10 bit sebanyak 3 buah.
- Timer/Counter sebanyak 3 buah.
- CPU 8 bit yang terdiri dari 32 register.
- Watchdog Timer dengan osilator internal.
- SRAM sebesar 512 byte.
- Memori Flash sebesar 8 Kbyte dengan kemampuan read while write.
- Interrupt internal maupun eksternal.
- Port komunikasi SPI
- EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
- Analog Comparator.
- Komunikasi serial standar USART dengan kecepatan maksimal 2,5 Mbps.



Gambar 3.1 Konfigurasi Pin ATmega 8535

Sumber www.docstoc.com/docs/36699306/leraning-microcontroller-avr-atmega85351632

Penjelasan masing-masing pin ATmega 8535 adalah sebagai berikut:

- VCC merupakan pin masukan catu daya.
- GND merupakan pin ground.
- Port A (PA0 - PA7) merupakan pin I/O dua arah dan pin masukan ADC.
- Port B (PB0 – PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer0 / Counter0 pada pin PortB 1, komparator analog dan SPL.
- Port C (PC0 – PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Watchdog Timer, komparator analog dan Timer Oscilator.
- Port D (PD0 – PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal dan komunikasi serial.
- RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
- XTAL1 dan XTAL2 merupakan pin masukan clock eksternal.

- AVCC merupakan pin masukan tegangan untuk ADC (Analog to Digital Converter).
- AREF merupakan pin masukan tegangan referensi ADC (Analog to Digital Converter).

Atmega 8535 mempunyai empat buah port yaitu PortA, PortB, PortC, dan PortD. Tiap port mempunyai tiga buah register bit, yaitu DDR_{xn}, PORT_{xn}, dan PIN_{xn}. Huruf 'x' mewakili nama huruf dari port sedangkan huruf 'n' mewakili nomor bit. Bit DDR_{xn} terdapat pada I/O address DDR_x, bit PORT_{xn} terdapat pada I/O address PORT_x, dan bit PIN_{xn} terdapat pada I/O address PIN_x. Bit DDR_{xn} dalam register DDR_x (*Data Direction Register*) menentukan arah pin. Bila DDR_{xn} diset 1, maka PIN_x berfungsi sebagai pin *output*. Bila DDR_{xn} diset 0 maka PIN_x berfungsi sebagai pin *input*. Bila PORT_{xn} diset 1 pada saat pin terkonfigurasi sebagai pin *input*. Bila PORT_{xn} diset 1 pada saat pin terkonfigurasi sebagai pin *output* maka pin port akan berlogika 1. Dan bila PORT_{xn} diset 0 pada saat pin terkonfigurasi sebagai pin *output* maka pin port akan berlogika 0.

Tabel. 3.1 Konfigurasi Pin Port

DDRxn	PORTxn	PUD (SFOR)	Pin I/O	Pull Up
0	0	x	input	no
0	1	0	input	yes
0	1	1	input	no
1	0	x	output	no
1	1	x	output	no

Tabel diatas menunjukkan konfigurasi pin pada port-port mikrokontroler. Bit 2 – PUD = *Pull-up Disable*, bila bit diset bernilai 1 maka *pull-up* pada port I/O akan dimatikan walaupun register DDxn dan PORTxn dikonfigurasi untuk menyalakan *pull-up* (DDRxn=0, PORTxn=1).

3.2 Sensor Jarak

Sensor jarak yang digunakan adalah tipe SRF04 yang merupakan sensor jarak yang memanfaatkan lama pantulan gelombang suara ultrasonik. Keluaran dari sensor ini berupa sinyal analog. Sensor ini terdiri dari 3 pin yang dapat dilihat pada gambar berikut:



Gambar 3.2 Sensor SRF 04

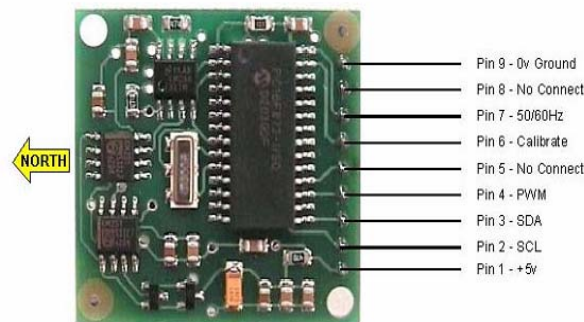
Sumber gambar: www.tokoelektronika.com/tutorial/praktikumrobot.html

Adapun cara kerja dari sensor ini dapat mendeteksi objek dengan cara memancarkan gelombang suara ultrasonik dan kemudian mendeteksi lama pantulannya. SRF akan memancarkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler (pulsa high selama 10uS). Gelombang ultrasonik ini akan dipancarkan selama 200uS sampai mengenai objek dan dipantulkan kembali. Proses menunggu pantulan akan menghasilkan sebuah pulsa dan pulsa ini akan berhenti ketika terdeteksi oleh SRF. Lebar pulsa inilah yang mempersentasikan jarak dari objek tersebut.

3.3 Sensor Arah

Sensor arah merupakan sensor navigasi yang digunakan untuk menunjukkan data dari tiap perbedaan arah mata angin. Sensor yang penulis gunakan ialah sensor CMPS03 *Magnetic Compass* buatan Devantech ltd.

Sensor ini disupply dengan tegangan 5V, mempunyai 9 pin yang dapat dilihat pada gambar berikut:



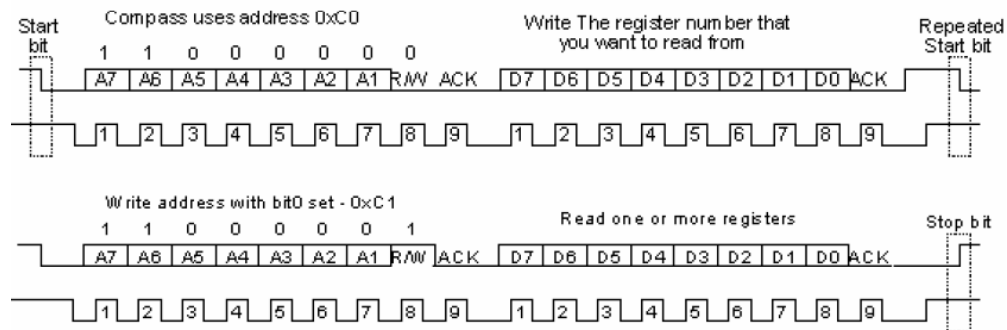
Gambar 3.3 CMPS03 Magnetic Compass

Sumber gambar <http://forum.didox.web.id/viewtopic.php?f=39&t=152>

Pembacaan data sensor kami menggunakan I2C yaitu SDA dan SCL. Kedua pin ini harus diberi *pullup* karena keluaran dari I2C kecil. Jadi data perlu dikuatkan dengan masukan arus yang lebih besar.

Pada robot yang dikerjakan penulis menggunakan jalur I2C, dikarenakan pada Codevision AVR C fasilitas I2C telah tersedia dan memudahkan penulis dalam melakukan pembuatan program.

I2C merupakan sebuah communication protocol dimana proses kerjanya dimulai dengan mengirimkan start bit, address modul digital compass dengan read/write low (0xC0), kemudian nomor register yang akan dibaca. Selanjutnya diikuti dengan start bit lagi, address modul digital compass dengan read/write high (0xC1). Selanjutnya anda bisa membaca satu atau dua register (8 bit atau 16 bit). Untuk register 16 bit, yang pertama kali dibaca adalah high byte. Berikut gambar proses I2C communication protocol.

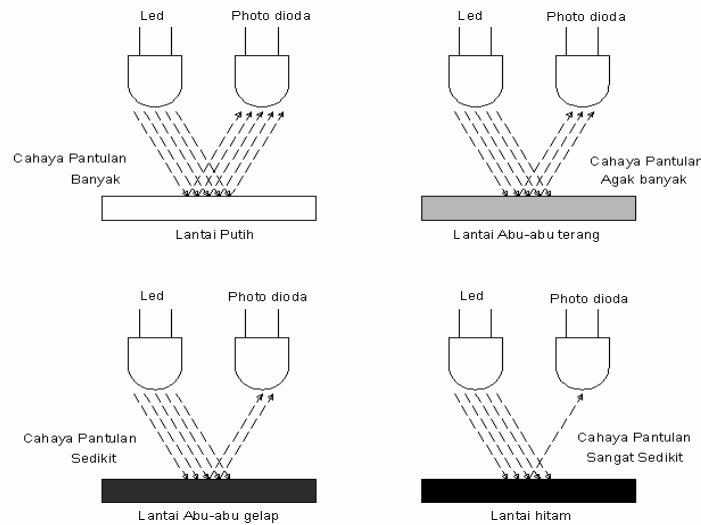


Gambar 3.4 Communication Protocol

Sumber gambar <http://forum.didox.web.id/viewtopic.php?f=39&t=152>

3.4 Sensor Warna

Sensor ini terdiri dari transmitter dan receiver. Led digunakan sebagai transmitter dan photodiode digunakan sebagai receiver. Led yang digunakan berwarna putih dan photodiode yang digunakan berukuran kecil yaitu 3mm.



Gambar 3.5 Pantulan Cahaya Led yang Ditangkap oleh Photodiode

Sensor ini bekerja membaca pantulan cahaya dari led ke lantai kemudian diterima oleh photodiode. Photodiode akan berubah tahanannya apabila menerima cahaya yang berbeda. Apabila jumlah cahaya yang diterima oleh photodiode besar maka tahanan photodiode akan kecil dan apabila cahaya yang diterima sedikit maka tahanannya akan besar. Perubahan nilai tahanan ini yang dimanfaatkan untuk merubah nilai tegangan yang akan dimasukkan ke ADC mikrokontroler. Data tegangan analog yang masuk ke ADC diubah ke data 8 bit. Data 8 bit yang telah diubah digunakan sebagai acuan untuk pembacaan warna lantai.

3.5 Motor Servo

Motor servo terdiri dari sebuah motor DC, serangkaian gear, sebuah potensiometer, sebuah shaft dan rangkaian kontrol elektronik. Biasanya motor servo berbentuk kotak segi empat dengan sebuah output shaft motor dan konektor dengan 3 kabel yaitu power, ground dan kontrol.

Gear motor servo ada yang terbuat dari plastik, karbon, metal dan titanium. *Gear* motor yang digunakan pada proyek ini adalah *gear* karbon. Di dalam motor servo terdapat potensiometer yang digunakan sebagai sensor posisi. Potensiometer tersebut dihubungkan dengan output *shaft* untuk mengetahui posisi aktual *shaft*. Ketika motor DC berputar, maka output *shaft* juga berputar dan sekaligus memutar potensiometer. Rangkaian kontrol kemudian dapat membaca kondisi. Rangkaian kontrol kemudian dapat membaca kondisi potensiometer tersebut untuk mengetahui posisi aktual *shaft*. Jika posisinya sesuai dengan yang diinginkan, maka motor DC akan berhenti. Sudut operasi motor servo (*Operating Angle*) bervariasi tergantung jenis motor servo.

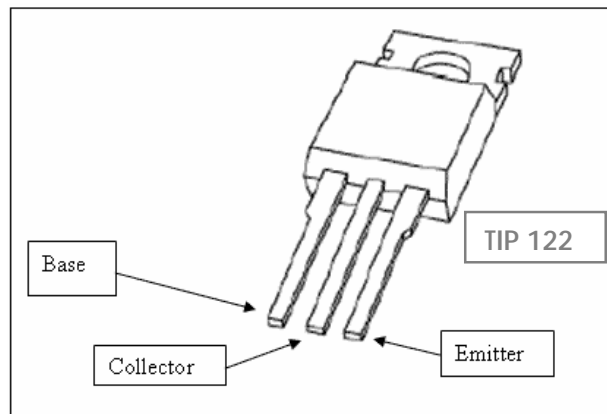
Cara kerja dari servo adalah dengan memberikan pulsa (PWM) ke input sinyal. Besarnya pulsa yang dimasukkan adalah 20mS. Dengan nilai periode high berkisar dari 1100uS – 1900uS. Selain dari lamanya periode *high* aktif selebihnya adalah kondisi periode low pulsa. Sehingga perioda yang harus dikeluarkan setiap memberikan data adalah dengan memberikan perioda sebesar 20mS.



Gambar 3.6 Bentuk dari Motor Servo
Sumber gambar www.robotshop.ca/hitec-hs-6965hb-servo-motor.html

3.6 Driver Motor Kipas

Driver motor kipas merupakan alat yang digunakan untuk mengaktifkan atau menggerakkan kipas. Driver ini bekerja dengan menggunakan rangkaian *switching* transistor TIP122. Transistor akan bekerja saturasi apabila diberi kondisi *high* pada basis karena transistor ini merupakan tipe NPN. Apabila kaki basis mendapat arus yang cukup untuk mengaktifkan kaki basis. Maka arus akan mengalir dari kaki kolektor ke kaki emitor transistor. Agar transistor aktif, dibutuhkan tegangan bias maju antara kaki basis dengan kaki emitor dan tegangan bias mundur antara kaki basis dengan kaki kolektor. Apabila kondisi ini terpenuhi maka transistor mendapat bias aktif.

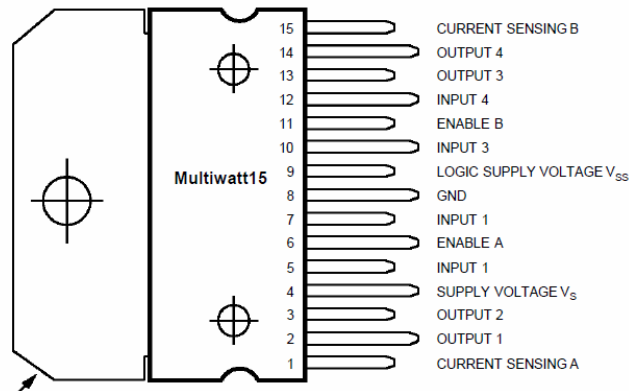


Gambar 3.7 Transistor TIP122

Sumber gambar www.nemeiksas.com/corporateimage/electronicproject.htm

3.7 Driver Motor DC penggerak roda

Driver motor DC ini merupakan rangkain yang digunakan untuk mengaktifkan atau menggerakkan motor DC. Rangkaian driver ini bekerja dengan menggunakan PWM dari IC L298. Alasan penulis menggunakan ic ini karena IC ini dapat berkerja dengan baik sampai dengan arus maksimal sekitar 4A dengan tegangan maksimal berkisar 40VDC. Selain itu juga IC ini mempunyai dua pin enable, 4 pin input dan 4 pin output. Karakteristik tersebut sesuai dengan kebutuhan untuk mengendalikan 2 buah motor dc.



Gambar 3.8 bentuk IC L298

Sumber gambar www.solarbotics.com/assets/datasheets/sgs_l298_motor_driver_ic.pdf

Setiap kaki enablenya memiliki 2 pin input dan 2 pin output, enable berfungsi untuk mengaktifkan apakah outputnya akan berlogika 1 atau 0. Sedangkan kaki input berfungsi sebagai pengontrol output, dimana output akan memiliki keluaran secara *variable*.

Table 3.2 kondisi motor.

Input	Output	Enable	Kondisi Motor
0	0	0	berhenti
1	0	1	berputar kekiri
0	1	1	berputar kekanan
1	1	1	mengerem (berhenti)

3.8 Baterai

Baterai merupakan sumber tegangan yang digunakan untuk mengaktifkan kinerja dari robot. Baterai yang digunakan adalah baterai *charge* keluaran perusahaan krisbow. Seri dari baterai yang digunakan adalah 2700mAh.

Baterai adalah perangkat yang mampu menghasilkan tegangan DC, yaitu dengan cara mengubah energi kimia yang terkandung didalamnya menjadi energi

listrik melalui reaksi elektro kima, Redoks (Reduksi – Oksidasi). Batere terdiri dari beberapa sel listrik, sel listrik tersebut menjadi penyimpan energi listrik dalam bentuk energy kimia. Sel batere tersebut elektroda – elektroda. Elektroda negatif disebut katoda, yang berfungsi sebagai pemberi elektron. Elektroda positif disebut anoda yang berfungsi sebagai penerima elektron. Antara anoda dan katoda akan mengalir arus yaitu dari kutub positif (anoda) ke kutub negatif (katoda). Sedangkan electron akan mengalir dari katoda menuju anoda.



Gambar 3.9 Baterai 2700 mAh (krisbow)

BAB IV

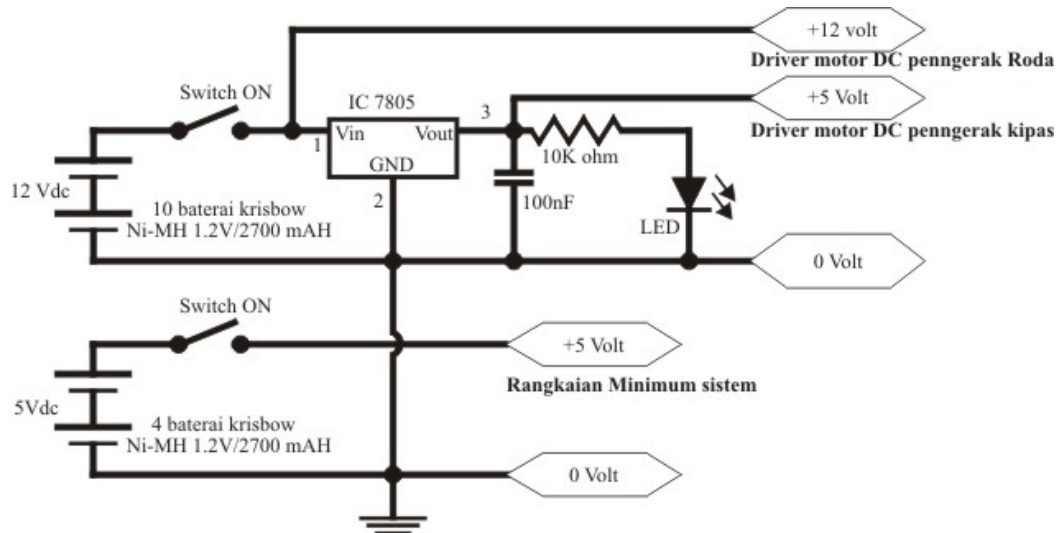
PERANCANGAN SISTEM

4.1 Perancangan Elektronika

4.1.1 Power Supply

Power supply pada robot ini menggunakan 14 baterai rechargeable NI-MH tipe AA 1.2 V / 2700 mAH. 10 baterai digunakan untuk mensupply tegangan pada motor dan 4 baterai untuk mensupply tegangan pada minimum sistem dan juga motor servo.

Berikut ini adalah rancangan rangkaian power supply :



Gambar 4.1 Power Supply dengan Regulator

4.1.2 Minimum Sistem Mikrokontroler ATmega 8535

Mikrokontroler yang digunakan dalam proyek ini menggunakan jenis mikrokontroler keluaran AVR yaitu ATmega 8535. Adapun I/O yang digunakan pada mikrokontroler ini adalah sebagai berikut:

A) Input Mikrokontroler ATmega 8535:

- Sensor jarak SRF 1 : Pin A.1
- Sensor jarak SRF 2 : Pin A.2
- Sensor jarak SRF 3 : Pin A.3
- Sensor warna : Pin A.0
- Tombol Start : Pin D.7
- Jalur data (SDA) : Pin D.5
- Jalur data (SCL) : Pin D.6
- Tombol reset : Pin reset ATmega

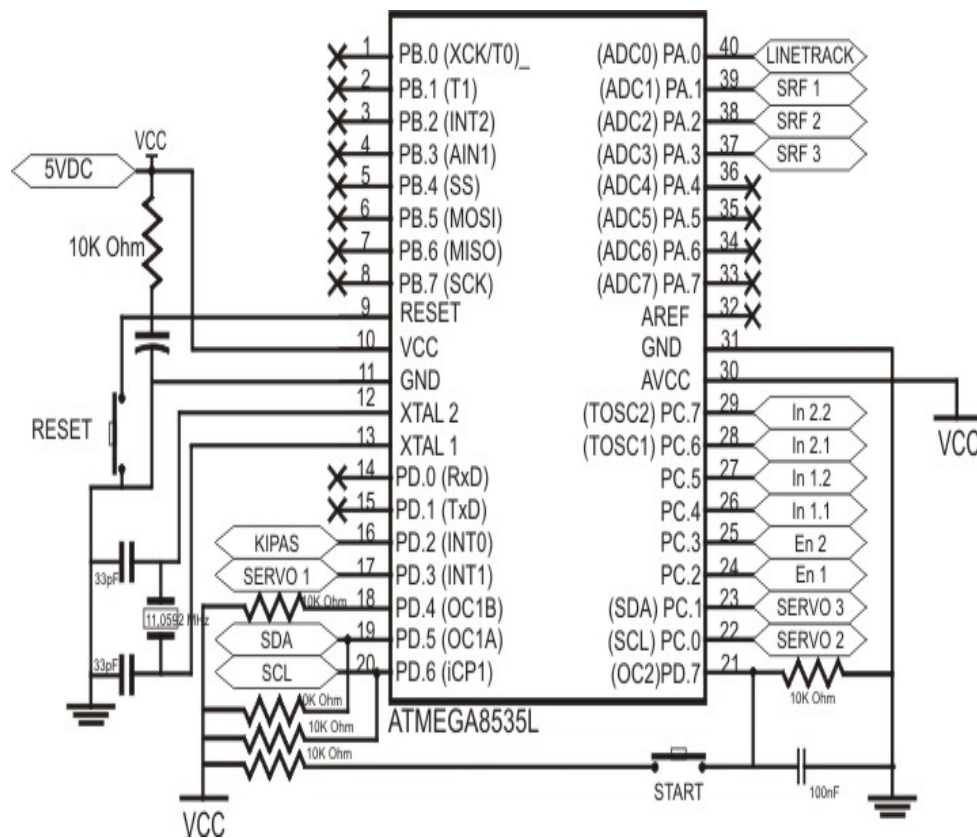
B) Output Mikrokontroler ATmega 8535:

- Motor DC kipas : Port D.2
- Servo 1 : Port D.3
- Servo 2 : Port C.0
- Servo 3 : Port C.1
- Enable 1 (driver motor L298) : Port C.2
- Enable 2 (driver motor L298) : Port C.3
- In 1.1 (driver motor L298) : Port C.4
- In 1.2 (driver motor L298) : Port C.5

- In 2.1 (driver motor L298) : Port C.6
- In 2.2 (driver motor L298) : Port C.7

C) Pin download :

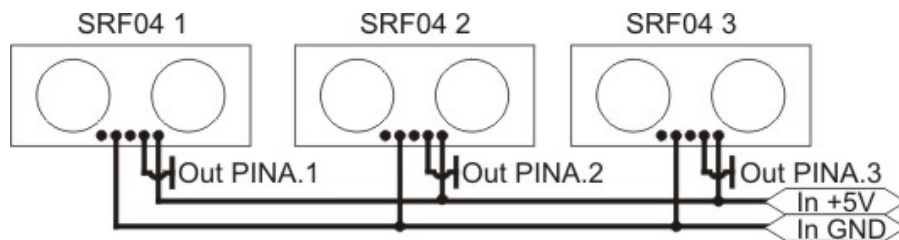
- MISO (SPI Bus Master Input/Slave Output) : Port B.5
- MOSI (SPI Bus Master Output/Slave Input) : Port B.6
- SCK (SPI Bus Serial Clock) : Port B.7



Gambar 4.2 Rangkaian minimum sistem mikrokontroler

4.1.3 Sensor Jarak

Sensor jarak SRF mempunyai keluaran digital, dimana kita dapat mengatur jarak yang kita inginkan dengan mengatur program yang kita buat.

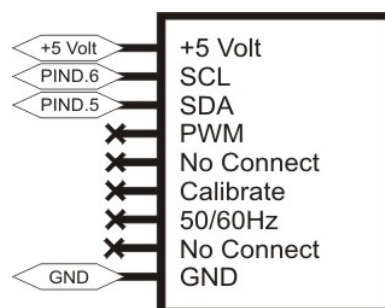


Gambar 4.3 Rangkaian Sensor jarak

Letak pemasangan sensor SRF ini berada pada posisi depan robot, sisi serong kiri robot dan sisi serong kanan pada robot.

4.1.4 Sensor Arah

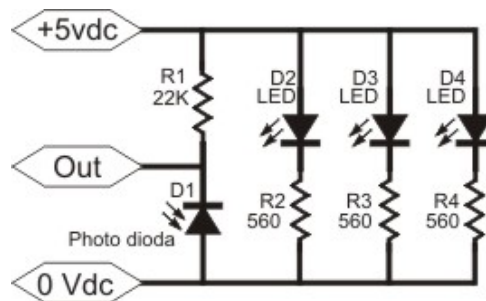
Sensor arah yang digunakan adalah *Digital Magnetic Compas CMPS03*. Data arah kompas dikeluarkan melalui pin dalam bentuk komunikasi I2C (*Inter Integrated Circuit*) untuk clock dimasukan melalui SCL. Masing-masing pin dihubungkan mikrokontroler menggunakan resistor *Pull Up* 10k ohm.



Gambar 4.4 Rangkain sensor arah

4.1.5 Sensor Warna

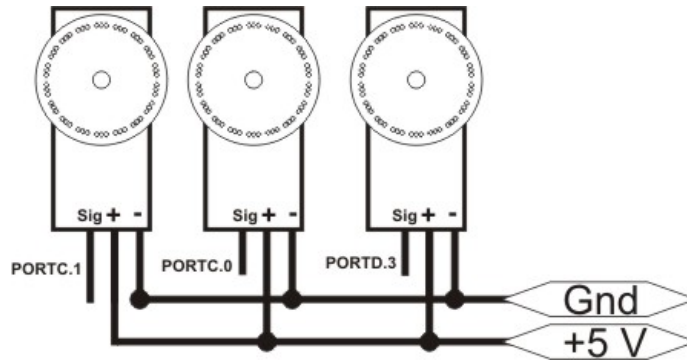
Sensor warna ini menggunakan 3 buah LED (*light emitting diode*) sebagai sumber cahaya. Output rangkain sensor warna berupa tegangan analog yang akan dibaca oleh input ADC mikrokontroller sehingga dapat dibedakan warna lapangan, yaitu warna putih, hijau dan warna gelap lainnya.



Gambar 4.5 Rangkaian Sensor Warna

4.1.6 Motor Servo

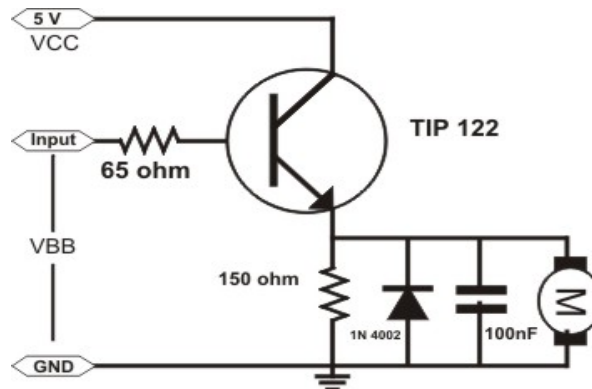
Servo yang digunakan adalah jenis Servo Hi-Tech. Total sudut putaran pada servo ini adalah 90°.



Gambar 4.6 Rangkaian Motor Servo

4.1.7 Driver Motor Kipas

Driver motor kipas ini menggunakan transistor TIP122 yang berfungsi sebagai saklar untuk memutus dan menyambung arus ke motor kipas. Driver ini berkerja pada daerah saturasi cut off.



Gambar 4.7 Driver Kipas terhubung ke Mikrokontroler

Perhitungan untuk menentukan berapa tegangan dan arus yang dibutuhkan sehingga motor dapat berkerja maksimal.

$$V_{BB} = (V_B + V_{BE} + V_E)$$

$$I_E > I_M$$

$$I_E = I_M + I_{RE}$$

$$V_{BB} = (I_B \times R_B) + V_{BE} + (I_E \times R_E)$$

$$I_B = 12\text{mA}$$

$$I_M = 65\text{mA}$$

$$I_E = I_M + I_{RE} = 90\text{mA}$$

$$I_{RE} = I_E - I_M = 90 - 65 =$$

25mA

Agar motor dapat berkerja dibutuhkan tegangan sebesar 3.5 V jadi nilai R_B dan R_E adalah :

$$V_{BB} = (V_B + V_{BE} + (I_E \times R_E))$$

$$V_B = (5V - 0.7V - 3.5V) = 0.8V$$

$$R_B = V_B / I_B$$

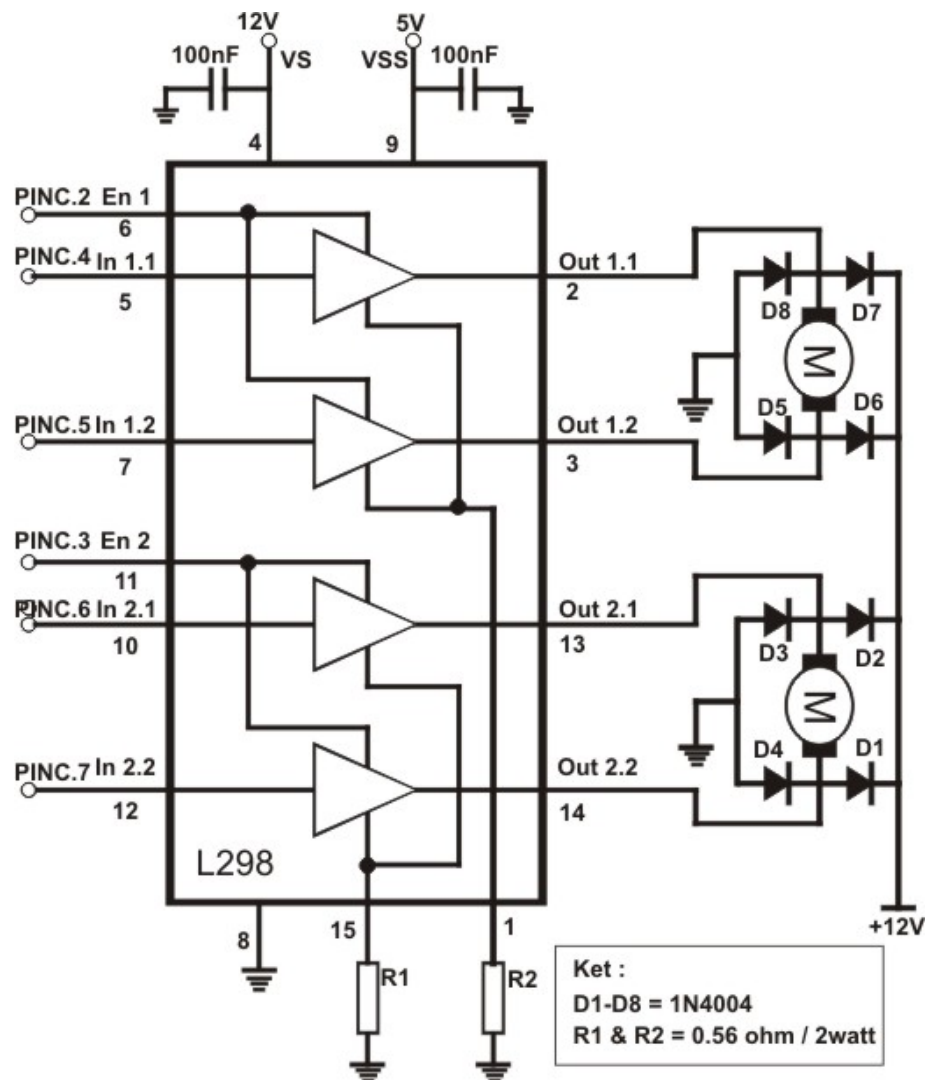
$$R_B = 0.8V / 0.012A = 66.6 \text{ ohm}$$

$$R_E = V_E / I_{RE}$$

$$R_E = 3.5V / 0.025 = 140 \text{ ohm}$$

4.1.8 Driver Motor Penggerak roda

Pada rangkaian driver motor ini menggunakan IC L298. Dengan pemasangan 4 buah dioda agar IC L298 tidak rusak akibat arus balik yang di timbulkan oleh motor.

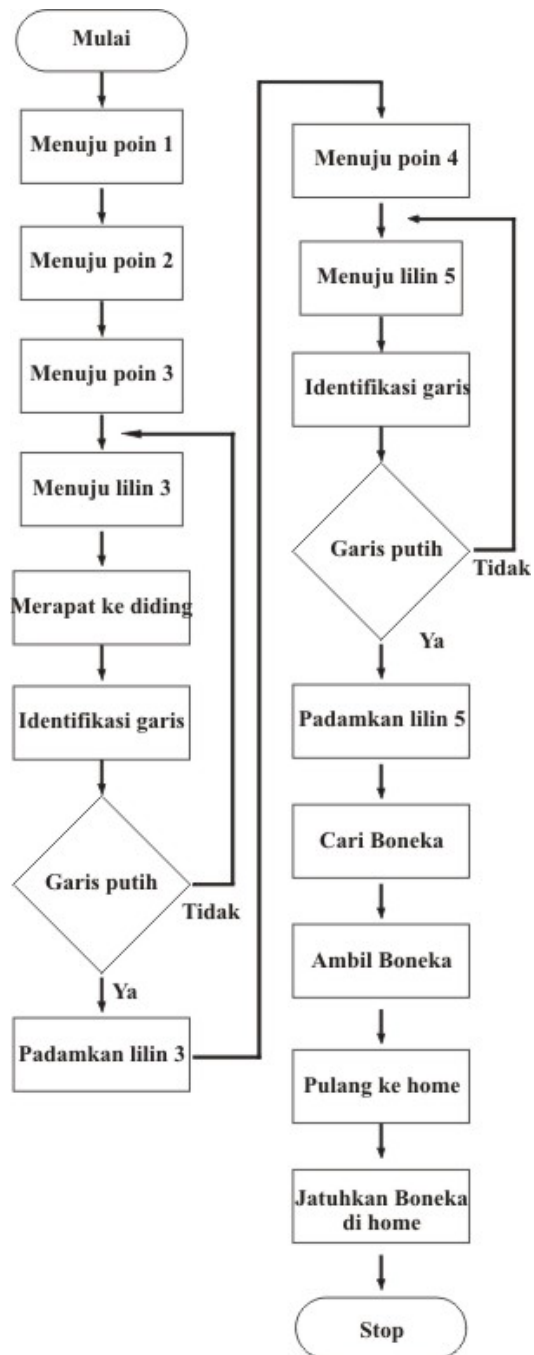


Gambar 4.8 skematik dasar driver motor DC

4.2 Perancangan Program

Optimasi program pada robot ini dibuat menggunakan software codevision C compiler. Dalam melaksanakan tugas robot harus bergerak dari home mencari pintu keluar mencari ruangan lain, memeriksa lilin dan mematikan lilin apabila

4.2.1 Flowchart Pemograman



Gambar 4.10 Flow Chart

Gambar diatas adalah flowchart robot dari awal hingga akhir. Proses yang dilakukan, pertama adalah keluar dari ruangan home kemudian menuju poin-poin tersebut dan mencari garis putih. Setelah itu memadamkan api. Setelah melakukan pemadaman api tersebut robot menuju posisi boneka dan kemudian mengambilnya. Setelah mendapat boneka yang sesuai, robot membawa boneka tersebut ke home awal dan meletaknya di home.

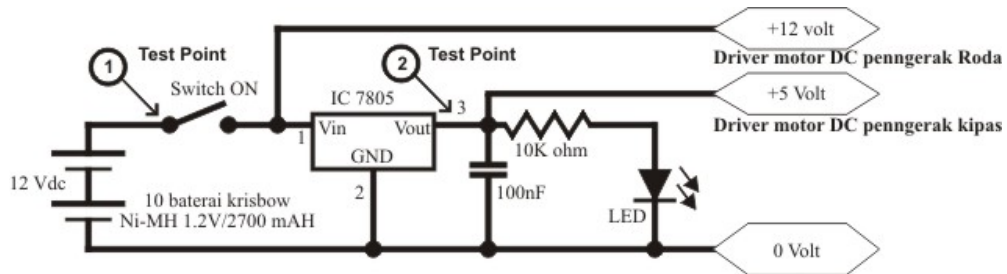
BAB V

PENGUKURAN, PENGUJIAN DAN ANALISA SISTEM

5.1 Pengukuran

Tujuan pengukuran ini dilakukan untuk mengetahui kinerja dari suatu sistem yang telah dirancang apakah sesuai dengan spesifikasi atau tidak. Alat yang digunakan dalam pengukuran ini adalah dengan menggunakan multimeter, oscilloscope, mistar, LCD dan stopwatch.

5.1.1 Pengukuran Tegangan Masukan dan Keluaran Power Supply



Gambar 5.1 Titik ukur power supply

Tabel 5.1 Hasil Pengukuran Power Supply dengan Regulator

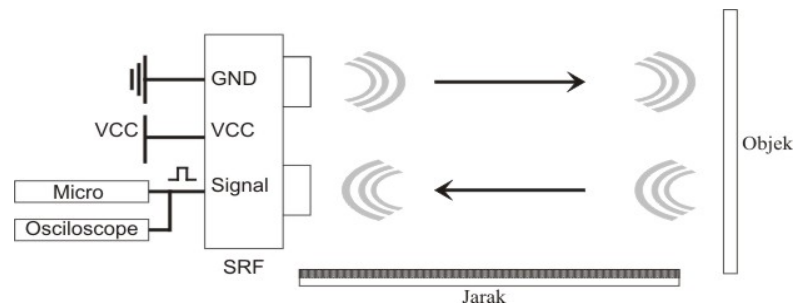
Test point	Switch on terbuka	Switch on tertutup
1	13.2 V	13.2 V
2	0 V	4.7 V

Untuk nilai persentasi *error* dari keluaran power supply adalah sebagai berikut:

$$\begin{aligned}\% \text{ error} &= (| \text{ nilai pengukuran} - \text{ nilai spesifikasi baterai} | / \text{ nilai spesifikasi} \\ &\text{ baterai}) * 100\% \\ &= (| 4.7 - 5 | / 5) * 100\% \\ &= (| -0.3 | / 5) * 100\% \\ &= 0.06 * 100\% \\ &= 6\%\end{aligned}$$

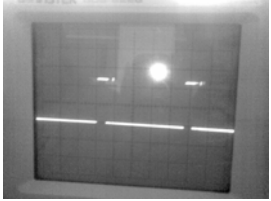
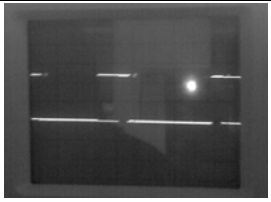
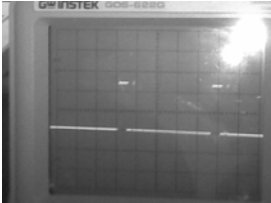

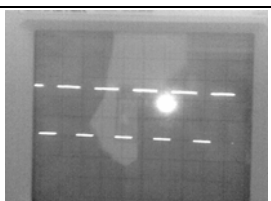
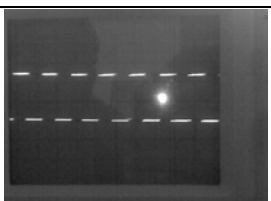
Tegangan yang dihasilkan sudah cukup untuk mengaktifkan kipas. Karena dengan tegangan 3.5 saja motor DC tamiya dapat berputar dengan baik.

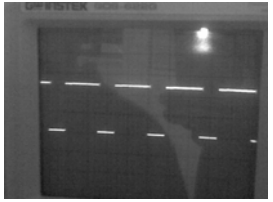
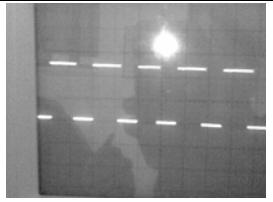


5.1.2 Pengukuran Sensor Jarak



Gambar 5.2 Pengukuran sinyal frekuensi dari Sensor Jarak

Tabel 5.2 Pengukuran Keluaran Sensor Jarak SRF

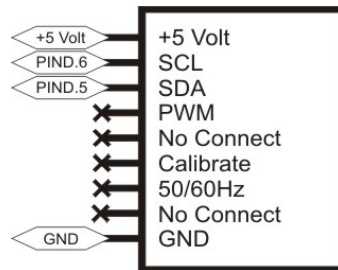
Jarak	Gambar Sinyal	Periode Sinyal High(ms)	Tegangan (Vpp)
10 cm		0,16	3,8
20 cm		0,35	3,8
30 cm		0,7	3,8
40 cm		1,4	3,8
50 cm		1,9	3,8
60 cm		2.4	3,8

70 cm		2,8	3,8
80 cm		3.05	3,8
90 cm		3,3	3,8
100 cm		3,65	3,8

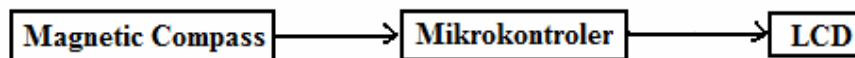
Dari hasil pengukuran diatas didapatkan kesimpulan bahwa nilai Periode signal dengan nilai Jarak adalah berbanding lurus dimana semakin besar nilai periode signal yang diterima oleh SRF maka semakin besar pula hasil pembacaan jaraknya .

5.1.3 Pengukuran Sensor Arah

Pengukuran data keluaran dari CMPS03 dilakukan dengan mengambil data dari SDA dan SCL yang hasil keluarannya tersebut di tampilkan pada LCD.



Gambar 5.3 Rangkaian Skematik *Magnetic Compass CMPS03*

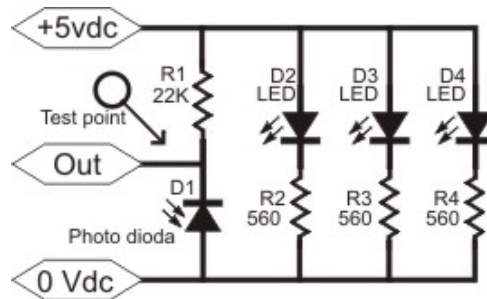


Gambar 5.4 Pengukuran atau Pengambilan Arah Data Compass

Tabel 5.3 Hasil Pengujian Sensor Arah CMPS03

Arah	Arah Kompas	Data Tampilan LCD
Utara	0°	0
Timur Laut	45°	32
Timur	90°	64
Tenggara	135°	96
Selatan	180°	128
Barat Daya	225°	160
Barat	270°	192
Barat Laut	315°	224

5.1.4 Pengukuran Sensor Warna

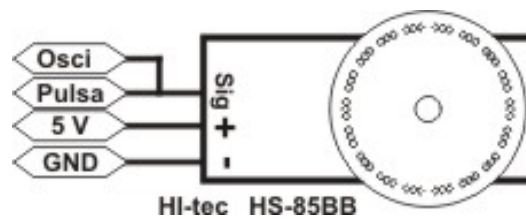


Gambar 5.5 Pengukuran Keluaran dari sensor warna

Tabel 5.4 Hasil Pengukuran Sensor Garis / Warna

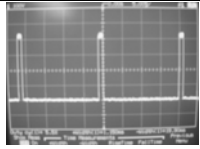
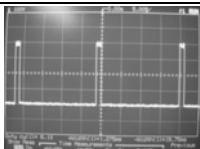
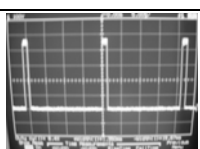
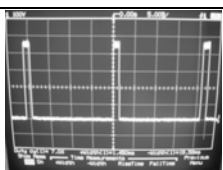
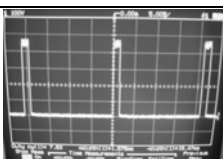
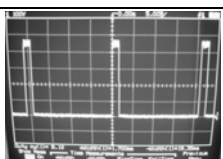
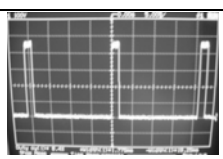
Warna	Tegangan Output (VDC) Test Point
Hitam	3,9 V
Hijau	2,5 V
Biru	2,73 V
Putih	0,36 V

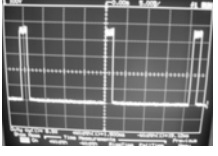
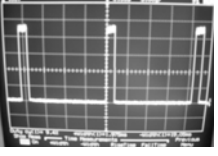
5.1.5 Pengukuran Data Motor Servo



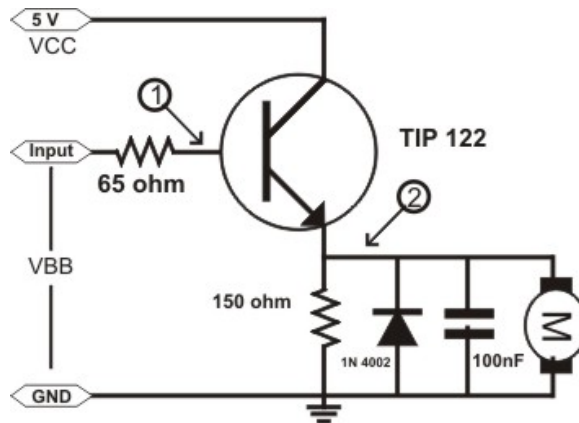
Gambar 5.6 Pengukuran Motor Servo

Tabel 5.5 Hasil Pengukuran Keluaran Motor Servo

Nilai Delay pada Program	Posisi Putaran Motor Servo (Derajat)	Gambar Sinyal	Periode Pulsa High	Periode Pulsa Low
1100 uS	0°		1100 uS	18.90 mS
1200 uS	10°		1200 uS	18.80 mS
1300 uS	20°		1300 uS	18.70 mS
1400 uS	30°		1400 uS	18.62 mS
1500 uS	40°		1500 uS	18.50 mS
1600 uS	50°		1600 uS	18.40 mS
1700 uS	60°		1700 uS	18.35 mS

1800 μ S	70°		1800 μ S	18.25 mS
1900 μ S	80°		1900 μ S	18.15 mS

5.1.6 Pengukuran Keluaran Driver Motor Kipas

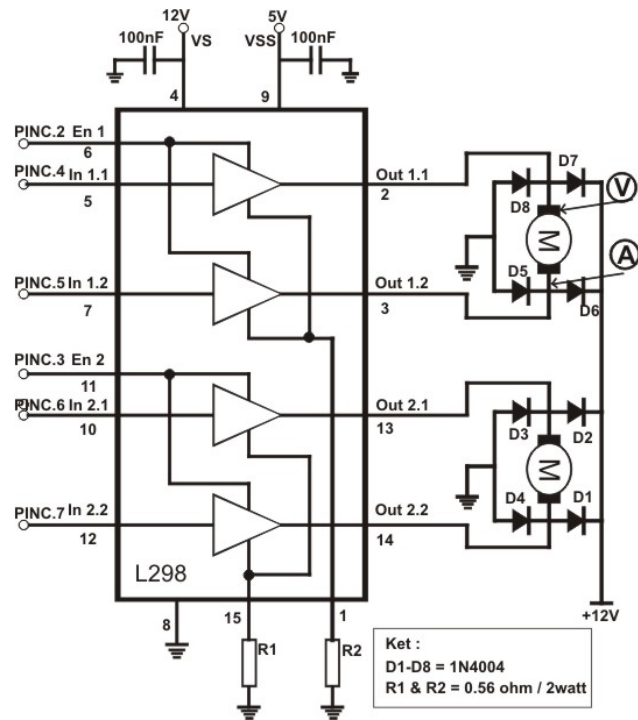


Gambar 5.7 Rangkaian Driver Motor Kipas

Tabel 5.6 Hasil Pengukuran Driver Motor Kipas

Test Point	Tegangan Keluaran (VDC)	
	Kondisi Input Low	Kondisi Input High
1	3,2mV	4.2V
2	-1.8 mV	3,7V

5.1.7 Pengukuran Keluaran Driver Motor DC 180 rpm



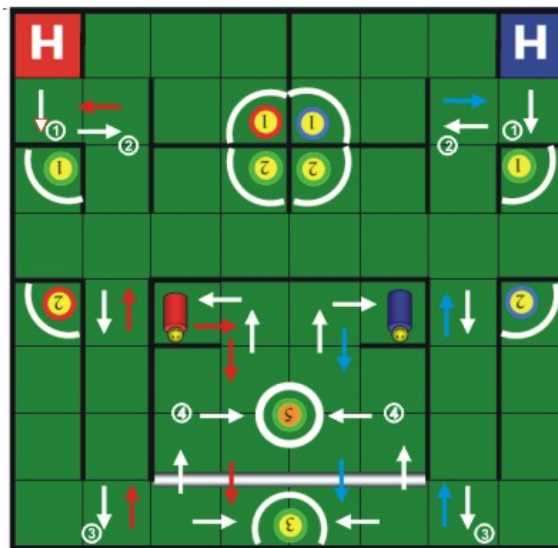
Gambar 5.8 Rangkaian driver motor

Tabel 5.7 tabel pengukuran driver motor

Data Pengujian Motor DC 180 rpm			
Tegangan (Volt)	PWM	Tegangan Pada Motor (V)	Arus Pada Motor (A)
12	0%	0.00	0.01
12	20%	5.42	0.050
12	40%	7.5	0.053
12	60%	8.42	0.054
12	80%	9.23	0.055
12	100%	11.05	0.056
13.2	0%	0.01	0.015
13.2	20%	7.2	0.057
13.2	40%	9.5	0.065
13.2	60%	11	0.069
13.2	80%	11.8	0.070
13.2	100%	12.00	0.072

5.2 Pengujian

Pengujian adalah proses yang paling terpenting karena dari pengujian ini akan terlihat apakah kinerja robot berjalan dengan baik dan sesuai yang diharapkan. Dalam pengujian robot. Beberapa sample akan diambil yang kemudian dibandingkan apakah persentase robot sudah mencapai hasil yang baik.



Gambar 5.9 Lapangan Pertandingan

Tabel 5.8 pengujian robot dalam satuan waktu

Data pengujian robot			
Posisi	Percobaan 1	Percobaan II	Percobaan III
Point 2	2.37 detik	2.54 detik	2.77 detik
Point 3	8.14 detik	8.78 detik	8.70 detik
Padam Lilin 3	22.39 detik	22.54 detik	22.47 detik
Padam Lilin 5	33.1 detik	33.47 detik	33.36 detik
Ambil boneka	43.56 detik	44.03 detik	43.68 detik
Letakan boneka	63.99 detik	64.49 detik	64.43 detik

Dari hasil pengujian diatas dapat dihitung berapa persentase error dalam tiga kali pengujian. Nilai tercepat adalah 63.99 detik dan yang terlambat adalah 64.49 detik. Jadi nilai errornya adalah :

$$\text{Error} = ((\text{waktu terlama} - \text{waktu tercepat})/\text{waktu tercepat}) \times 100\%$$

$$\text{Error} = ((64.49 - 63.99)/63.99) \times 100\% = 0.782\%.$$

Nilai errornya adalah 0.782 %

Dengan nilai error seperti itu robot dapat dikatakan mempunyai error yang kecil. Nilai error tersebut bisa saja diakibatkan oleh beberapa hal, antara lain:

- Kondisi lapangan dengan robot. Terkadang robot mengalami slip pada saat melakukan pengereman.
- Nilai tegangan power supply yang semakin lama semakin menurun.
- Pembacaan sensor – sensor yang kurang sempurna.

5.3 Analisa Sistem

5.3.1 Power Supply

Dengan rangkaian power supply yang telah dibahas pada bab sebelumnya didapat bahwa hasil keluaran power supply dengan IC regulator 7805 tidak maksimal yaitu sebesar 4.72 V. Hal ini tidak sesuai dengan datasheet IC itu sendiri karena dituliskan bahwa IC tersebut mengeluarkan tegangan berkisar 4.8V – 5.2V. Tetapi dengan tegangan sebesar 4.72 tidak berpengaruh banyak terhadap performa kipas, kipas mampu berkerja dengan tegangan 3.5V.

Dengan tegangan 4.72V rangkaian minimum sistem tetap aktif karena tegangan operasinya berkisar 4.5 – 5.2 akan tetapi tidak cukup baik apabila digunakan untuk mensupply sensor seperti sensor jarak SRF04 dimana tegangan operasinya berkisar 5 – 6V. Sehingga untuk rangkaian minimum sistemnya menggunakan 4 baterai krisbow AA jenis Ni-MH 1.2V/2700mAH dengan keluaran total mencapai 5.2V. Dengan nilai keluaran tersebut semua sensor yang digunakan dapat berkerja dengan baik.

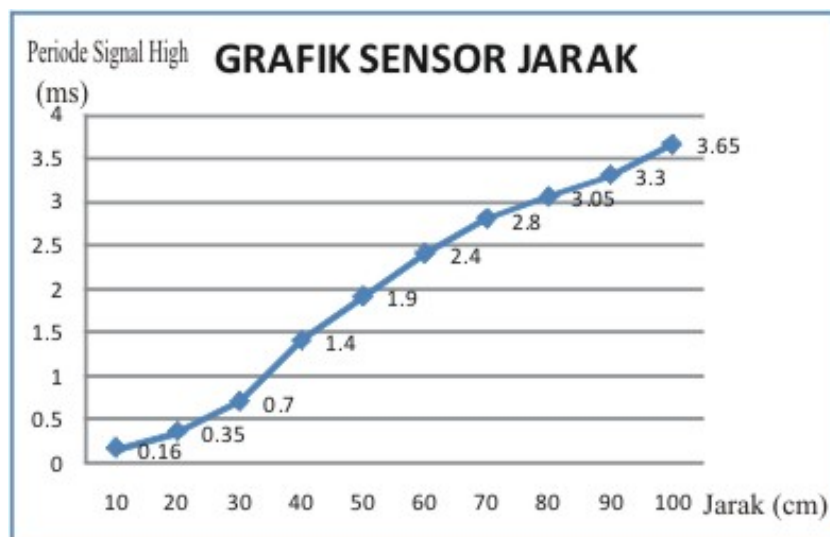
5.3.2 Sensor Jarak

Sensor jarak yang digunakan adalah jenis SRF04. SRF04 mempunyai tegangan operasi minimal 5V – 6V. Sedangkan ATmega8535 mempunyai tegangan operasi maksimal sebesar 5.2V. Akibatnya adalah saat tegangan dibawah 5v maka minimum sistem tetap bekerja sedangkan SRF04 tidak aktif, sehingga tanpa pembacaan jarak robot menjadi error dan kacau.

Selain itu juga SRF04 mempunyai karakteristik area pembacaan yang cukup luas yaitu sebesar 55° ke kanan dan 55° kekiri dari titik tengah. Sehingga ketika robot berjalan kadangkala dinding-dinding yang berada disamping robot pada lapangan terbaca seolah – olah berada didepan robot. Akibatnya robot menjalankan perintah yang seharusnya tidak dilakukan. Dapat disimpulkan bahwa untuk penggunaan sensor jarak dengan area yang sempit dianjurkan untuk penggunaan jenis sensor jarak lainnya yang karakteristik area pembacaannya lebih kecil.

Tabel 5.9 Hasil Pengukuran sensor jarak SRF04

periode signal high (ms)	jarak (cm)
0.16	10
0.35	20
0.7	30
1.4	40
1.9	50
2.4	60
2.8	70
3.05	80
3.3	90
3.65	100



Gambar 5.10 grafik pengukuran sensor jarak SRF04

Dari hasil pengukuran sensor jarak SRF04 terlihat grafik yang tidak linier.

5.3.3 Sensor Warna

Berdasarkan hasil pengukuran dan pengujian yang telah dilakukan bahwa nilai output tegangan sensor yang dihasilkan bisa berbeda-beda dengan warna yang sama. Beberapa faktor yang mempengaruhi kinerja dan kepekaan sensor.

- Intensitas cahaya (cahaya ruangan)
- Faktor warna
- Posisi penempatan sensor
- Jarak antara photo dioda dan LED
- Banyak LED yang dipasang

5.3.4 Motor Servo

Dalam penggunaan motor servo yang perlu diperhatikan adalah nilai torsi masing-masing servo dan juga bahan dasar gearnya. Karena ini berpengaruh terhadap kemampuan servo. Apakah sesuai dengan beban yang dibebankan atau tidak. Pada robot ini menggunakan dua jenis servo. Pertama adalah servo dengan gear berbahan plastik dengan torsi 3kg/cm dan yang kedua motor servo dengan gear berbahan dasar titanium dengan torsi 30kg/cm. Untuk yang berbahan titanium digunakan sebagai penjepit robot, hal ini ditujukan agar penjepit robot sewaktu membawa boneka tetap kuat dan boneka tidak jatuh. Sedangkan untuk motor servo dengan gear berbahan plastik digunakan untuk menggerakkan dudukan kipas agar dapat berputar sampai dengan 80°.

5.3.5 Driver Motor Kipas

Driver motor kipas yang digunakan adalah sistem switching dari sebuah transistor yaitu TIP 122 jenis NPN. Dimana arus yang mampu dilewatkan oleh transistor ini cukup besar. Karena transistor ini mampu melewatkan arus sampai dengan maksimal 5A. Untuk penggunaan rangkaian ini sebaiknya tidak menggunakan power supply dengan IC regulator untuk mensupply tegangan VCCnya. Karena rangkaian ini akan membebani IC dan mengakibatkan IC regulatornya cepat panas.

5.3.6 Driver Motor DC

Pada robot ini terdapat 4 buah roda dengan dua buah motor yang masing-masing motor menggerakkan 2 buah roda dengan bantuan *timing belt*. Untuk driver motor itu sendiri menggunakan IC L298. IC L298 mampu melewatkan arus sebesar 2A amper namun dari beberapa sumber IC ini juga mampu melewatkan arus lebih dari 3A hanya saja penggunaannya sedikit berbeda yaitu secara parallel. Dimana pin EN1 dan EN2 dengan satu sumber yang sama kemudian IN1.1 dan IN2.1 juga dipasang dengan satu sumber begitu juga dengan pin input dan output yang lain dipasang dengan cara yang sama. IC L298 cenderung cepat panas sehingga untuk mengatasinya perlu pemasangan *headsink* pada *body* IC.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil analisa data diatas dapat diambil beberapa kesimpulan diantaranya:

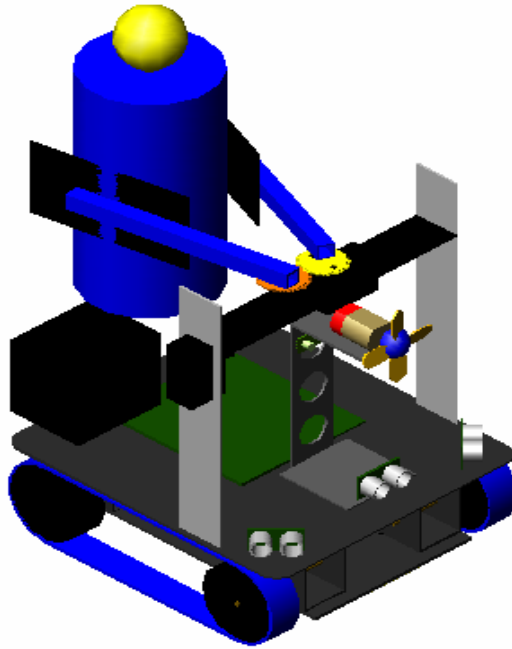
1. Robot dapat bekerja sesuai aturan dalam KRCI expert battle 2009 dengan waktu rata – rata 64.30 untuk memadamkan 2 lilin, mengambil boneka dan meletakkan boneka ke Home.
2. Untuk sensor jarak SRF04, menghasilkan perbandingan jarak objek dengan keluaran pulsa high yang tidak linier.

6.2 Saran

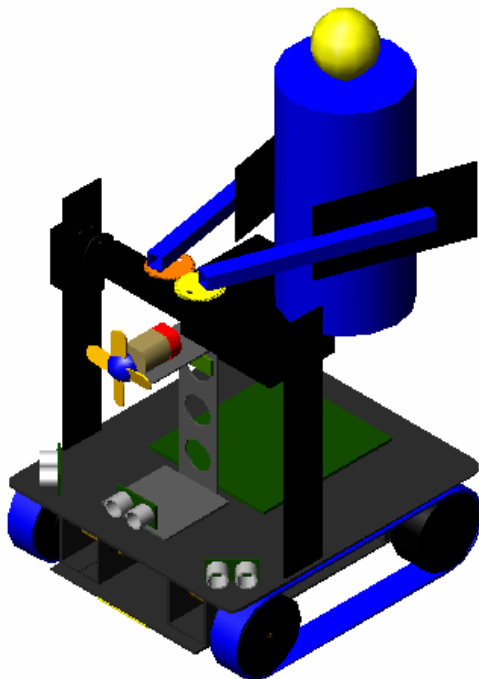
Untuk sistem yang lebih baik kedepannya perlu ditambahkan beberapa hal diantaranya :

1. Perlunya pengembangan sensor warna yang tidak terpengaruh cahaya ruangan.
2. Perlunya menggunakan sensor api untuk mendeteksi ada tidaknya api.
3. Perlu dikembangkan system kendali kecepatan motor dc dengan umpan balik (feed back) sehingga kecepatan motor dc menjadi stabil.

DIMENSI ROBOT



Gambar dimensi robot tampak samping kanan atas



Gambar

dimensi robot tampak samping kiri atas

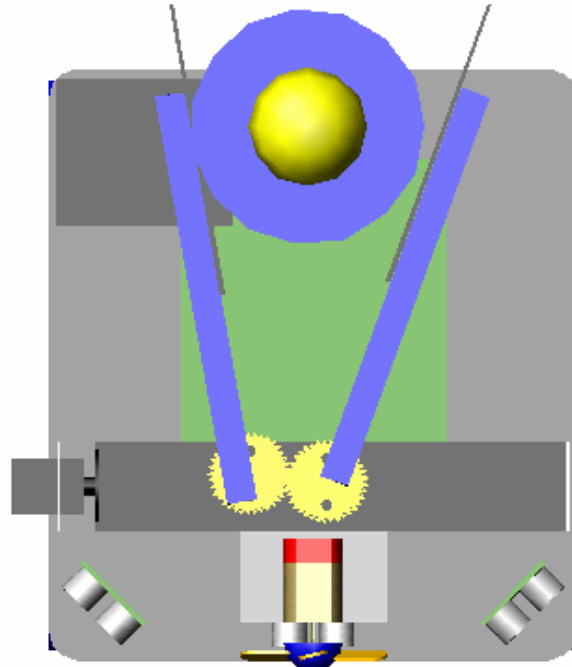
Ket :

Tinggi robot: 25cm

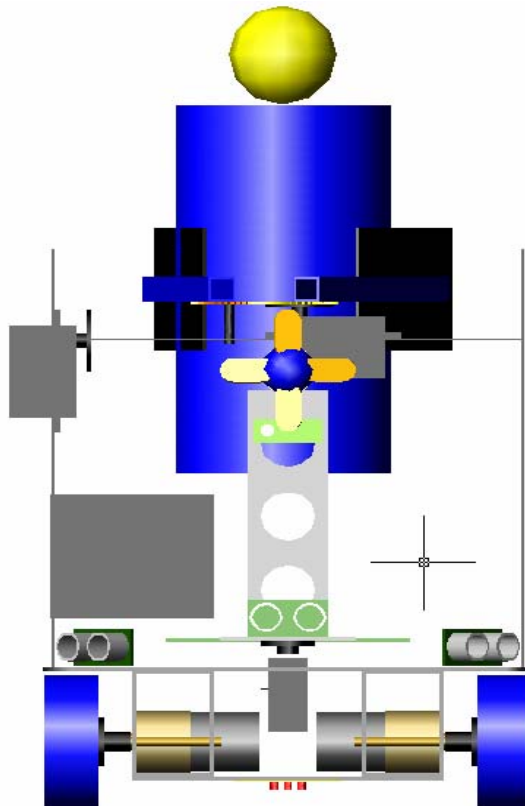
Tinggi robot + griiper = 37cm

Lebar robot = 25 cm

Panjang robot = 26 cm



Gambar dimensi robot tampak atas



Gambar dimensi robot tampak depan

List Program Menggunakan Code Vision dalam Bahasa C

A. Sensor Jarak SRF

```

#include <mega8535.h>
#include <stdio.h>
#include <delay.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

// Declare your global variables here
unsigned int Count1;
unsigned int Jarak1;
bit SRF1;
unsigned char kata1[16];
void SRFI()
{
    Count1=0;
    DDRA.1=1;
    PORTA.1=1;
    delay_us(10);
    PORTA.1=0;
    DDRA.1=0;
    PORTC.A=1;
    while(PINA.1==0){};
    while(PINA.1==1) Count1++;
    Jarak1=(Count1/5.288115756);
    if(Jarak1<250)SRF1=1; else SRF1=0;
}

void OutDataSRF()
{
    lcd_clear();
    sprintf(kata1,"%5d",Jarak1);
    lcd_gotoxy(0,0);lcd_puts(kata1);
    lcd_gotoxy(0,1);
    if(SRF1)
        lcd_putsf("Dekat");
    else
        lcd_putsf("Jauh ");
    delay_ms(10);
}

void main(void)
{
    .
    .
    .

```

```

while (1)
{
    // Place your code here
    SRFI();
    OutDataSRF();
};
}

```

B. Sensor Arah Magnetic Compass

```

#include <mega8535.h>
#include <stdio.h>
#include <delay.h>

// I2C Bus functions
#asm
    .equ __i2c_port=0x12 ;PORTD
    .equ __sda_bit=5
    .equ __scl_bit=6
#endasm
#include <i2c.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

// Declare your global variables here
unsigned char data;
unsigned int posisi;
unsigned char Msg1[16];
void compass_read()
{
    i2c_start();
    i2c_write(0xC0);
    i2c_write(0x01);
    i2c_start();
    i2c_write(0xC1);
    data=i2c_read(0);
    i2c_stop();
}
void TampilCompass()
{
    posisi=data;
    sprintf(Msg1,"Bearing=%d",data);
    lcd_clear();
    lcd_gotoxy(0,0);lcd_puts(Msg1);
}
void main(void)
{
    .
    .
    .
}

```

```

while (1)
{
    // Place your code here
    compass_read();
    TampilCompass();
};
}

```

C. Sensor Warna/Garis dan Kipas

```

#include <mega8535.h>
#include <stdio.h>
#define kipas PORTD.2

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>
#include <delay.h>
#define ADC_VREF_TYPE 0xE0

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}
// Declare your global variables here
unsigned char buf1[16];
unsigned char dt;
bit LTrack;
unsigned char linetrack()
{
    line=read_adc(0);
    if(line>0&&line<120)
        {buka=10; /*lcd_gotoxy(0,0); lcd_putsf("putih");*/}
    else
    if(line>120&&line<190)
        {buka=20; /*lcd_gotoxy(0,0); lcd_putsf("merah");*/}
    else
    if(line>190&&line<255)
        {buka=30; /*lcd_gotoxy(0,0); lcd_putsf("biru");*/}
    return(buka);
}

```

```

void main(void)
{
.
.
while (1)
{
    // Place your code here
    Linetrack();
If (linetrack<15)
    PORTD.2=1;
    Dlay_ms(1000);
    PORTD.2=0;
};
}

```

D. Motor Servo

```

#define Servo PORTC.0
#include <delay.h>
#include <mega8535.h>
//Declare your global variables here
void main(void)
{
.
.
while(1)
{
    // Place your code here
    for(i=0;i<25;i++)
    {
        Servo=1;delay_us(1500);Servo=0;delay_us(18500);
    }
    for(i=0;i<25;i++)
    {
        Servo=1;delay_us(2000);Servo=0;delay_us(18000);
    }
};
}

```

E. Motor Servo

```

#include <mega8535.h>
#include <stdio.h>
#include <delay.h>

#define enka    PORTC.2
#define enki    PORTC.3
#define dirki1  PORTC.4
#define dirki2  PORTC.5
#define dirka1  PORTC.6
#define dirka2  PORTC.7
#define start   PIND.7

```

```
// I2C Bus functions
#asm
    .equ __i2c_port=0x12 ;PORTD
    .equ __sda_bit=5
    .equ __scl_bit=6
#endasm
#include <i2c.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Place your code here
    xcount++;
    if(xcount<=lpwm)enki=1;
    else enki=0;
    if(xcount<=rpwm)enka=1;
    else enka=0;
    TCNT0=0xff;
}
// Declare your global variables here
void maju()
{
    //unsigned int i;
    lpwm=200;    rpwm=200;
    dirki1=0; dirki2=1;
    dirka1=0; dirka2=1;
}

void main(void)
{
    .
    .
    .
    while (1)
    {
        // Place your code here
        Maju ();
    }
};
```

Lampiran Listing Program Robot

/******

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

Project :
Version :
Date : 4/28/2009
Author :
Company :
Comments:

Chip type : ATmega8535L
Program type : Application
Clock frequency : 11.059200 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 128

*****/

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

```
#define enka PORTC.2  
#define enki PORTC.3  
#define dirki1 PORTC.4  
#define dirki2 PORTC.5  
#define dirka1 PORTC.6  
#define dirka2 PORTC.7  
#define start PIND.7  
#define servo1 PORTC.0  
#define servo PORTC.1  
#define servo2 PORTD.1  
#define kipas PORTD.2
```

```
unsigned char xcount, rpwm, lpwm;  
unsigned char line, word[16], word1[16];
```

```
unsigned int data_adc, data_adc1, Count1, Count2, Count3, Count4;  
unsigned int jrk_ka, jrk_dep, jrk_blk, jrk_ki;  
unsigned int Count4, i, Count1, Count2, Count3, i;
```

```

// I2C Bus functions
#asm
    .equ __i2c_port=0x12 ;PORTD
    .equ __sda_bit=5
    .equ __scl_bit=6
#endasm
#include <i2c.h>

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Place your code here
    xcount++;
    if(xcount<=lpwm)enki=1;
    else enki=0;
    if(xcount<=rpwm)enka=1;
    else enka=0;
    TCNT0=0xff;
}

#include <delay.h>

#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

// Declare your global variables here

void maju()
{
    lpwm=250;    rpwm=250;
    dirki1=0; dirki2=1;
    dirka1=0; dirka2=1; }

void maju_pelan()
{
    lpwm=90;    rpwm=60;
    dirki1=0; dirki2=1;
    dirka1=0; dirka2=1; }

```

```

void stop()
{ dirki1=1; dirki2=1;
  dirka1=1; dirka2=1;
  enka=1; enki=1;}

void mundur()
{ lpwm=200;   rpwm=200;
  dirki1=1; dirki2=0;
  dirka1=1; dirka2=0;}

void mundur_pelan()
{lpwm=80;   rpwm=80;
  dirki1=1; dirki2=0;
  dirka1=1; dirka2=0; }

void belok_kanan()
{ lpwm=200;   rpwm=200;
  dirki1=0; dirki2=1;
  dirka1=0; dirka2=0; }

void belok_kanan1()
{ lpwm=10;   rpwm=200;
  dirki1=0; dirki2=1;
  dirka1=0; dirka2=1; }

void belok_kanan_lambat()
{ lpwm=10;   rpwm=100;
  dirki1=0; dirki2=1;
  dirka1=0; dirka2=1; }

void belok_kiri()
{ lpwm=200;   rpwm=200;
  dirki1=0; dirki2=0;
  dirka1=0; dirka2=1;}

void belok_kiril()
{ lpwm=200;   rpwm=10;
  dirki1=0; dirki2=1;
  dirka1=0; dirka2=1; }
void belok_kiri_lambat()
{ lpwm=100;   rpwm=10;
  dirki1=0; dirki2=1;
  dirka1=0; dirka2=1; }

void pivot_kanan()
{ lpwm=255;   rpwm=255;
  dirki1=0; dirki2=1;
  dirka1=1; dirka2=0; }

void pivot_kiri()
{ lpwm=255;   rpwm=255;
  dirki1=1; dirki2=0;
  dirka1=0; dirka2=1; }

```

```

/*****
/***** Program Servo *****/
/*****

void angkat()
    { for(i=0;i<30;i++){servo1=1;
      delay_us(1100);servo1=0;delay_us(18900);} }

void balikan()
    { for(i=0;i<30;i++){servo1=1;
      delay_us(1700);servo1=0;delay_us(18300);} }

void jepit()
    { for(i=0;i<30;i++){servo2=1;
      delay_us(1400);servo2=0;delay_us(18600);} }

void buka_jepit()
    { for(i=0;i<30;i++){servo2=1;
      delay_us(1700);servo2=0;delay_us(18300);} }

void tahan()
    { for(i=0;i<10;i++){servo2=1;
      delay_us(1500);servo2=0;delay_us(18500);}
      for(i=0;i<10;i++){servo2=1;
      delay_us(1700);servo2=0;delay_us(18300);} }

void kanan1()
    {for(i=0;i<10;i++){servo=1; delay_us(1900);servo=0;delay_us(18100);}}
void kanan2()
    {for(i=0;i<10;i++){servo=1; delay_us(1800);servo=0;delay_us(18200);}}
void kanan3()
    {for(i=0;i<10;i++){servo=1; delay_us(1700);servo=0;delay_us(18300);}}
void kanan4()
    {for(i=0;i<10;i++){servo=1; delay_us(1600);servo=0;delay_us(18400);}}
void tengah()
    {for(i=0;i<10;i++){servo=1; delay_us(1500);servo=0;delay_us(18500);}}
void kiril()
    {for(i=0;i<10;i++){servo=1; delay_us(1400);servo=0;delay_us(18500);}}
void kiri2()
    {for(i=0;i<10;i++){servo=1; delay_us(1300);servo=0;delay_us(18700);}}
void kiri3()
    {for(i=0;i<10;i++){servo=1; delay_us(1200);servo=0;delay_us(18800);}}
void kiri4()
    {for(i=0;i<10;i++){servo=1; delay_us(1100);servo=0;delay_us(18900);}}
void ke_tengah1()
    {kanan1(); kanan2(); kanan3(); kanan4(); tengah(); }
void ke_kiri()
    {tengah();kiril();kiri2();kiri3();kiri4();}
void ke_kanan()
    { tengah(); kanan4(); kanan3(); kanan2(); kanan1(); }
void ke_tengah2()
    { kiri4(); kiri3(); kiri2(); kiril();tengah(); }
void geleng()
    { ke_kanan(); ke_tengah1(); ke_kiri(); ke_tengah2(); }

```

```

void geleng_kiri()
  { ke_kiri(); ke_tengah2(); ke_kiri(); ke_tengah2(); }

void servo_centre()
  { for(i=0;i<20;i++)
    { servo=1; delay_us(1550); servo=0; delay_us(18450); } }

/*****
/***** program srf *****/
/*****

void SRF_dep()
{ Count1=0; DDRA.0=1;
  PORTA.0=1; delay_us(10);
  PORTA.0=0; DDRA.0=0;
  PORTA.0=1; while(PINA.0==0){}
  while(PINA.0==1)
    { Count1++; }
  jrk_dep=(Count1/5.288115756)*2;
}

void SRF_kanan()
{ Count2=0; DDRA.1=1;
  PORTA.1=1; delay_us(10);
  PORTA.1=0; DDRA.1=0;
  PORTA.1=1; while(PINA.1==0){};
  while(PINA.1==1)
    { Count2++; }
  jrk_ka=(Count2/5.288115756)*2;
}

void SRF_kiri()
{ Count4=0; DDRA.3=1;
  PORTA.3=1; delay_us(10);
  PORTA.3=0; DDRA.3=0;
  PORTA.3=1; while(PINA.3==0){};
  while(PINA.3==1)
    { Count4++; }
  jrk_ki=(Count4/5.288115756)*2;
}

void ping_aktif()
{
  SRF_dep();
  SRF_kiri();
  SRF_kanan();
}

/*****
/***** Program Linetrack *****/
/*****

unsigned char linetrack_depan()
{
  line=read_adc(6);
  return(line);}

```

```

void linetrack_aktifkan()
{
    linetrack_depan();
}

/*****
/*****          Program Kompas          *****/
/*****/

unsigned char data;
void kompas_aktif()
{
    i2c_start();
    i2c_write(0xC0);
    i2c_write(0x01);
    i2c_start();
    i2c_write(0xC1);
    data=i2c_read(0);
    i2c_stop();
}

unsigned int kompas_baca()
{
    unsigned int a;
    i2c_start();
    i2c_write(0xC0);
    i2c_write(0x01);
    i2c_start();
    i2c_write(0xC1);
    a=i2c_read(0);
    i2c_stop();
    return a;
}

void pvt_90ka()
{
    int data_lokal, posisi_lokal, x_lokal;

    data_lokal=0;
    posisi_lokal=0;
    x_lokal=0;

    //compass_read();
    data_lokal=kompas_baca();
    if(data_lokal<128)
    {
        posisi_lokal=data_lokal+55;
        do
        {
            pivot_kanan();
            delay_ms(10);
            stop();
            data_lokal=kompas_baca();
        }
        while(data_lokal<posisi_lokal);
    }
}

```

```

else
{
    x_lokal=data_lokal-128;
    posisi_lokal=x_lokal+60;
    do
    {
        pivot_kanan();
        delay_ms(10);
        stop();
        data_lokal = kompas_baca();
        if(data_lokal<128) x_lokal=data_lokal+128;
        else x_lokal=data_lokal-128;
    }
    while(x_lokal<posisi_lokal);
}
}

```

```

void pvt_90ki()
{
    int data_lokal, posisi_lokal, x_lokal;

    data_lokal=0;
    posisi_lokal=0;
    x_lokal=0;

    //compass_read();
    data_lokal=kompas_baca();
    if(data_lokal>127)
    {
        posisi_lokal=data_lokal-45;
        do
        {
            pivot_kiri();
            delay_ms(10);
            stop();
            data_lokal=kompas_baca();
        }
        while(data_lokal>posisi_lokal);
    }

    else
    {
        x_lokal=data_lokal+127;
        posisi_lokal=x_lokal-45;
        do
        {
            pivot_kiri();
            delay_ms(10);
            stop();
            data_lokal = kompas_baca();
            if(data_lokal>127) x_lokal=data_lokal-128;
        }
        while(x_lokal>posisi_lokal);
    }
}
}

```

```

void pvt_90ki_lebih()
{
    int data_lokal, posisi_lokal, x_lokal;

    data_lokal=0;
    posisi_lokal=0;
    x_lokal=0;

    //compass_read();
    data_lokal=kompas_baca();
    if(data_lokal>127)
    {
        posisi_lokal=data_lokal-55;
        do
        {
            pivot_kiri();
            delay_ms(10);
            stop();
            data_lokal=kompas_baca();
        }
        while(data_lokal>posisi_lokal);
    }

    else
    {
        x_lokal=data_lokal+127;
        posisi_lokal=x_lokal-55;
        do
        {
            pivot_kiri();
            delay_ms(10);
            stop();
            data_lokal = kompas_baca();
            if(data_lokal>127) x_lokal=data_lokal-128;
        }
        while(x_lokal>posisi_lokal);
    }
}
/*****
/*****      PROGRAM TES KOMPAS      *****/
/*****

// inisialisasi    1 = pelan
// inisialisasi    2 = kencang
int MajuLurus(int arah, int batas, int kecepatan) {
    int arahVirtual, dataVirtual;
    SRF_aktif();
    if (jrk_dep < batas) {
        stop();
        return 1;
    }
    kompas_aktif();
    if (arah < 32) {
        arahVirtual = arah + 128;
        if (data >= 128) dataVirtual = data - 128;
        else dataVirtual = data + 128;
        if (dataVirtual == 255) dataVirtual = 0; }
}

```

```

else if (arah > 224) {
    arahVirtual = arah - 128;
    if (data < 128) dataVirtual = data + 128;
    else dataVirtual = data - 128;
    if (dataVirtual == 255) dataVirtual = 0;
}
else {
    arahVirtual = arah;
    dataVirtual = data;
}
if ((dataVirtual > arahVirtual + 1) ) {
    if (kecepatan==2) {
        belok_kiri1();
        delay_ms(100);

    } else {
        belok_kiri_lambat();
        delay_ms(100);
    }
}
else if ((dataVirtual < arahVirtual - 1) ) {
    if(kecepatan==2) {
        belok_kanan1();
        delay_ms(100);
    } else {
        belok_kanan_lambat();
        delay_ms(100);
    }
}
else {
    if(kecepatan==1) maju_pelan();
    else maju();
}
delay_ms(100);
return 0;
}

void BelokKanan(int arah) {
int arahVirtual, dataVirtual;
arahVirtual = 0; dataVirtual = 0;
do {
    SRF_aktif();
    kompas_aktif();
    //tampilkan_di_lcd();
    if (arah < 32) {
        arahVirtual = arah + 128;
        if (data >= 128) dataVirtual = data - 128;
        else dataVirtual = data + 128;
        if (dataVirtual == 255) dataVirtual = 0;
    }
    else if (arah > 224) {
        arahVirtual = arah - 128;
        if (data < 128) dataVirtual = data + 128;
        else dataVirtual = data - 128;
        if (dataVirtual == 255) dataVirtual = 0;
    }
}
}

```

```

        else {
            arahVirtual = arah;
            dataVirtual = data;
        }
        belok_kanan();
        delay_ms(100);
    } while(dataVirtual < arahVirtual);
}

void BelokKiri(int arah) {
    int arahVirtual, dataVirtual;
    arahVirtual = 0; dataVirtual = 0;
    do {
        SRF_aktif();
        kompas_aktif();
        //tampilkan_di_lcd();
        if (arah < 32) {
            arahVirtual = arah + 128;
            if (data >= 128) dataVirtual = data - 128;
            else dataVirtual = data + 128;
            if (dataVirtual == 255) dataVirtual = 0;
        }
        else if (arah > 224) {
            arahVirtual = arah - 128;
            if (data < 128) dataVirtual = data + 128;
            else dataVirtual = data - 128;
            if (dataVirtual == 255) dataVirtual = 0;
        }
        else {
            arahVirtual = arah;
            dataVirtual = data;
        }
        belok_kiri();
        delay_ms(100);
    } while(dataVirtual > arahVirtual);
}

void main(void)
{
    // Declare your local variables here
    int ketemuBoneka, langkah, sudutBerikutnya, hasil;
    int arahAwal;
    // Input/Output Ports initialization
    // Port A initialization
    // Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=T State3=T State2=T State1=T
    State0=T
    PORTA=0x00;
    DDRA=0x00;

    // Port B initialization
    // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
    Func1=Out Func0=Out
    // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0
    State0=0
    PORTB=0x00;
}

```

```

DDRB=0xFF;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0
State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=Out Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=0 State2=0 State1=T
State0=T
PORTD=0x00;
DDRD=0x1E;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 10.800 kHz
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x05;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

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

```

```

TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;

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

// ADC initialization
// ADC Clock frequency: 345.600 kHz
// ADC Voltage Reference: AVCC pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x85;
SFIOR&=0xEF;

// I2C Bus initialization
i2c_init();

// LCD module initialization
lcd_init(16);

// Global enable interrupts
#asm("sei")

arahAwal = 0;
ketemuBoneka = 0;
sudutBerikutnya = 0;
langkah = 1;
hasil = 0;
kipas = 0;
servo_centre();
while (!start) {}
while (start) {
    delay_ms(200);
}
kompas_aktif();
delay_ms(200);
arahAwal = data;

```

```

while (1)
{
    // Place your code here
    /*****
    /*****      Program Jalan      *****/
    /*****
    if(!ketemuBoneka) { switch(langkah) {
        case 1:
            hasil = MajuLurus(arahAwal, 350,2);
            if (hasil) {
                sudutBerikutnya = arahAwal+50;
                if (sudutBerikutnya>254) sudutBerikutnya -= 254;
                BelokKanan(sudutBerikutnya);
                langkah = 2; }

            break;
        case 2:
            hasil = MajuLurus(sudutBerikutnya, 280,2);
            if (hasil) {
                sudutBerikutnya = arahAwal+10;
                if (sudutBerikutnya<0) sudutBerikutnya += 254;
                BelokKiri(sudutBerikutnya);
                langkah = 3; }

            break;
        case 3:
            hasil = MajuLurus(arahAwal, 340,2);
            if (hasil) {
                pvt_90ka();
                langkah = 4; }

            break;
        case 4:
            sudutBerikutnya = arahAwal+5;
            if (sudutBerikutnya>=254) sudutBerikutnya -= 254;
            //MajuLurus(sudutBerikutnya, 100 ,2); delay_ms(500);
            do {
                SRF_aktif();
                maju(); }

            while (linetrack_depan() > 18) {
                maju_pelan();
                if(SRF_kiri() < 150) belok_kanan_lambat();
                if(SRF_kiri() > 200) belok_kiri_lambat();
            }

            stop();

            kipas = 1;
            geleng();delay_ms(500);
            kipas = 0;
            langkah = 5;
            break;

```

```

case 5:
    mundur();
    delay_ms(1000);
    pivot_kanan();
    delay_ms(500);
    langkah = 6;
    break;

case 6:
    maju();
    delay_ms(1000);
    stop();
    delay_ms(500);
    SRF_aktif();
    while (jrk_dep > 350) {
        maju_pelan();
        SRF_aktif(); }
    pivot_kiri();
    delay_ms(500);
    langkah = 8;
    break;

case 8:
    while (linetrack_depan() > 18) {
        maju_pelan(); }
    stop(); kipas = 1;
    geleng(); delay_ms(500);
    kipas = 0; mundur(); delay_ms(150);
    langkah = 9;
    break;

case 9:
    pivot_90ka();
    maju_pelan();
    do {SRF_aktif();
        maju_pelan();
    } while (jrk_dep > 200);
    stop();
    langkah = 10;
    break;

case 10:
    pivot_kanan(); delay_ms(560);
    do {
        ping_aktif();
        if (SRF_kiri() > 40) {
            belok_kiri_lambat();
            delay_ms(100); }
        else if (SRF_kiri() < 180) {
            belok_kanan_lambat();
            delay_ms(100); }
        else maju_pelan();
    } while (jrk_dep > 50);
    stop(); langkah = 11;
    break;

```

```

case 11:
    buka_jepit();   angkat();
    jepit();   balikan();
    langkah = 12;
    break;

case 12:
    mundur_pelan();
    delay_ms(1000);
    stop();
    delay_ms(100);
    SRF_aktif();
    while (jrk_dep < 350) {
        mundur_pelan();
        SRF_aktif(); }

    pvt_90ka();
    langkah = 13;
    break;

case 13:
    do {   maju();
        SRF_aktif();} while (jrk_dep > 200);
    stop(); langkah = 14;
    break;

case 14:
    pvt_90ki(); langkah = 15;
    break;

case 15:
    hasil = MajuLurus(arahAwal, 320,2);
    if (hasil) { pvt_90ki();
        langkah = 16;}
    break;

case 16:
    hasil = Majulurus(253, 570, 2)
    if (hasil) { pvt_90ka();
        langkah = 17;}
    break;

case 17:
    do { SRF_aktif(); maju}
    while (jrk_dep<200) pvt_90ki();
    langkah = 18:
    break;

case 18;
    maju(); delay_ms(300) stop();
    buka(); default: break; }
}
};
}

```