Design and Construction of Obstacle Avoiding Robot

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in the Electrical and Electronic Engineering.

Submitted By

Biplob Sarker

ID No: 142-33-1884

Department of Electrical and Electronic Engineering (EEE)

Daffodil International University, Dhaka

Supervised By

Mr. Md. Mahmudur Rahman

Assistant Professor

Department of Electrical and Electronic Engineering (EEE)

Daffodil International University, Dhaka

Daffodil International University

Dhaka, Bangladesh

September, 2018

Certification

This is to certify that this project and thesis entitled "**Obstacle Avoiding Robot Using Ultrasonic Sensor Arduino**" is done by the following students under my direct supervision and this work has been carried out by them in the Department of Electrical and Electronic Engineering under the Faculty of Engineering of Daffodil International University in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering. The presentation of the work was held on September 2018.

Supervised By:

Mr. Md. Mahmudur Rahman
Assistant Professor
Department of Electrical and Electronic Engineering (EEE)
Daffodil International University

Submitted By:

Biplob Sarker
ID: 142-33-1884
Department of Electrical and Electronic Engineering (EEE)
Daffodil International University

ACKNOWLEDGEMENT

Firstly I give thanks to almighty Allah from the bottom of my hearts.

I would like to express my sincere gratitude to my honorable supervisor, **Assistant Professor Mr. Md. Mahmudur Rahman**, **Department of Electrical and Electronic Engineering**, DIU who inspired me in every moment. I am thankful to him for his continuous encouragement, kind co-operation, and scholastic guidance all along the project work. He has always been extremely generous with his time, knowledge and ideas and allowed me great freedom in this research.

I express my humble gratitude to all teachers of Department of Electrical and Electronic Engineering for their support in numerous ways throughout this project work. I am also grateful to the authors whose valuable research papers and books I have considered as reference in this project paper.

Apart from that, I would like to thank my friends for sharing knowledge; information and helping me in making this project a success. Also thanks for lending me some tools and equipment.

Finally I would like to thank my parents who have given us tremendous inspirations and supports. Without their mental and financial supports, I would not able to complete my project.

Author

ABSTRACT

Now day's many industries are using robots due to their high level of performance and reliability and which is a great help for human beings. The obstacle avoidance robotics is used for detecting obstacle and avoiding the collision. This is an autonomous robot. The design of obstacle avoidance robot requires the integration of many sensors according to their task. The obstacle detection is primary requirement of this autonomous robot. The robot gets the information from surrounding area through mounted sensors on the robot. Some sensing devices used for obstacle detection like bump sensor, infrared sensor, ultrasonic sensor etc. Ultrasonic sensor is most suitable for obstacle detection and it is of low cost and has high ranging capability. This can be design to build an obstacle avoidance robotic vehicle using ultrasonic sensors for its movement. A micro-controller (AT mega 2560) is used to achieve the desired operation. An ultrasonic sensor is used to detect any obstacle ahead of it and sends a command to the micro-controller. Depending on the input signal received, the micro-controller redirects the robot to move in an alternate direction by actuating the motors which are interfaced to it through a motor driver.

Table of Contents

List of Figures

List of Table

CHAPTER 1

INTRODUCTION

1.1 Introduction

The project presented here was based on Arduino and was more simple and reliable than simple micro controller based Obstacle Avoiding Robot Using Ultrasonic Sensor Arduino. Here is a Robot which is used to interface with the project to output lock status. "In this project I have an extra advantage that an Obstacle Avoiding Robot is an intelligent robot, which can automatically sense and overcome obstacles on its path". "It contains of a Microcontroller to process the data, and Ultrasonic sensors to detect the obstacles on its path".

1.2 Problem Statement

At the moment, robotic technologies have become more important since a lot of industry is trying to improve their machinery weapons. "This technology has developed year by year to make sure an excellent result". "Recently, by time goes by, a lot of mechanical robots have been designed to help people running their daily life".

1.3 Aim of the Project

- It can be used for home work like automatic vacuum cleanup.
- It can even be utilized in dangerous environments, wherever human penetration may be fatal.
- Design a proper circuit diagram.
- Implement the design.

1.4 Scopes

- These robots can also be used as automatic cars running on roads with embedded magnets
- It can be used commercially in industries for safety of workers and staff.
- To save the lives of workers it is very effective. By using it at commercial level many precious lives can be saved
- Avoid situations that are harmful.

1.5 Methodology

- Collection of project information's from book & internet.
- Required components have been purchased from market.

1.6 Organization of the Report

This project report has seven chapters in total. The first chapter describes an idea about my project "Obstacle Avoiding Robot Using Ultrasonic Sensor Arduino", Brief description of the project, Problem Statement, aim of the project and scopes. The second chapter is about history, block diagram, circuit diagram and list of components. The chapter third about component description, cost analysis of my system. The chapter fourth software analysis & program explanation. The chapter five hardware implementation. Then chapter six describes result & discussion properly. Finally, chapter seven gives the concluding remarks, limitation of my system and suggestion for the future works.

CHAPTER 2

SYSTEM REVIEWS

2.1 Introduction

A robot was a machine that can redact task automatically or with guidance. "Robotics was generally a combination of computational intelligence and physical machines (motors)". Computational intelligence includes the programmed commands in this project.

The project propound robotic vehicle that has an intelligence constructed in it such that it leaders itself whenever an obstacle comes ahead of it. "This robotic vehicle is built, using a microcontroller of Arduino Mega2560 family".

The ultrasonic sensor was used to detect any obstacle ahead of it and sends a command to the Arduino Mega 2560 board. "Ultrasonic sensor was most suitable for obstacle detection and it was very low cost and has high ranging capability". This ultrasonic detector be made of two parts: an emitter, "which produces a 40 kHz sound wave; and a detector", which detects this 40 kHz sound wave and sends the signal back to the Arduino Mega 2560 board.

This concept in future can be comprehensive in such a way that if a destination was fed to the robot, "the robot can map the whole terrain and can reach its destination by deciding a suitable path and avoiding obstacles".

2.2 General Block Diagram

Figure No 2.1: Obstacle Avoiding Robot Using Ultrasonic Sensor Arduino General Block Diagram

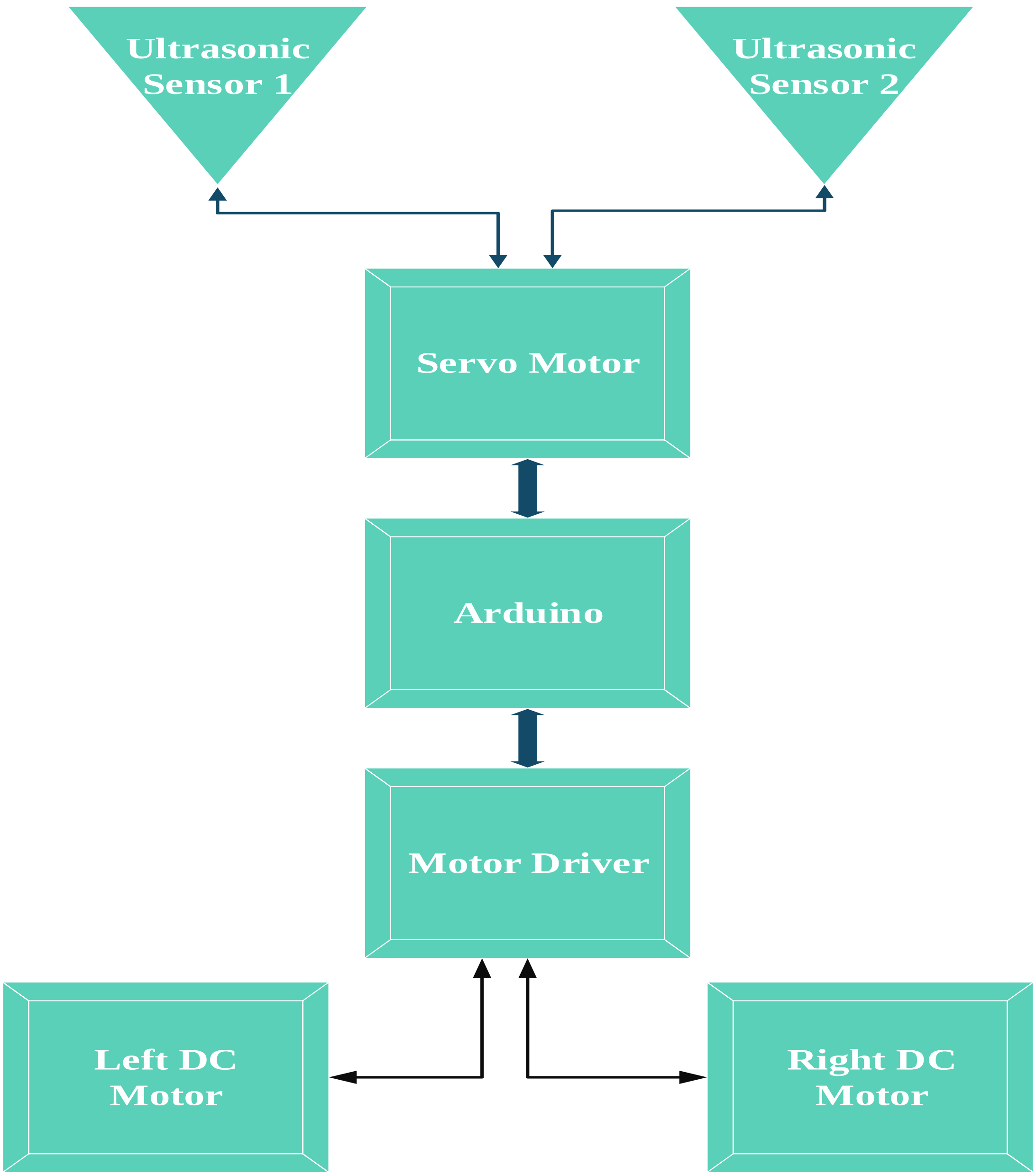


Fig. 2.1: General Block Diagram [1]

2.2.1 Block Diagram Description

In this block diagram, ultrasonic sensor was connected to SG90 servomotor. Arduino was also connected to L298N H-bridge. DC motor was connected to L298N H-bridge. The circuit comprises of Arduino Mega2560 board, L298N H-Bridge, servo motor, ultrasonic sensor and a few other components. Dual Motor Controller Module 2A with Arduino. This allowed to control the speed and direction of two DC motors. The L298N Bridge is connected to the Arduino digital pins 4, 5, 6 and 7. "The 5 Volts for the L298N board was now being supplied from the Arduino 5 Volt output. Trigger pin will be used to send the signal and the Echo pin will be used to listen for returning signal". "The 5V pin of the ultrasonic sensor was connected to the 5V pin on the board, the GND pin was connected to the GND pin, and the SIG (signal) pin was connected to digital pin 10 on the board".

2.3 Circuit Diagram

To complete this project fulfill a circuit diagram is very important. Without circuit diagram a project are not complete properly. Because without a circuit diagram developer cannot understand the problem. One of the most important advantage of circuit diagram is developer can theoretically complete a project very easily to draw a circuit diagram. Circuit diagram connection is given below.

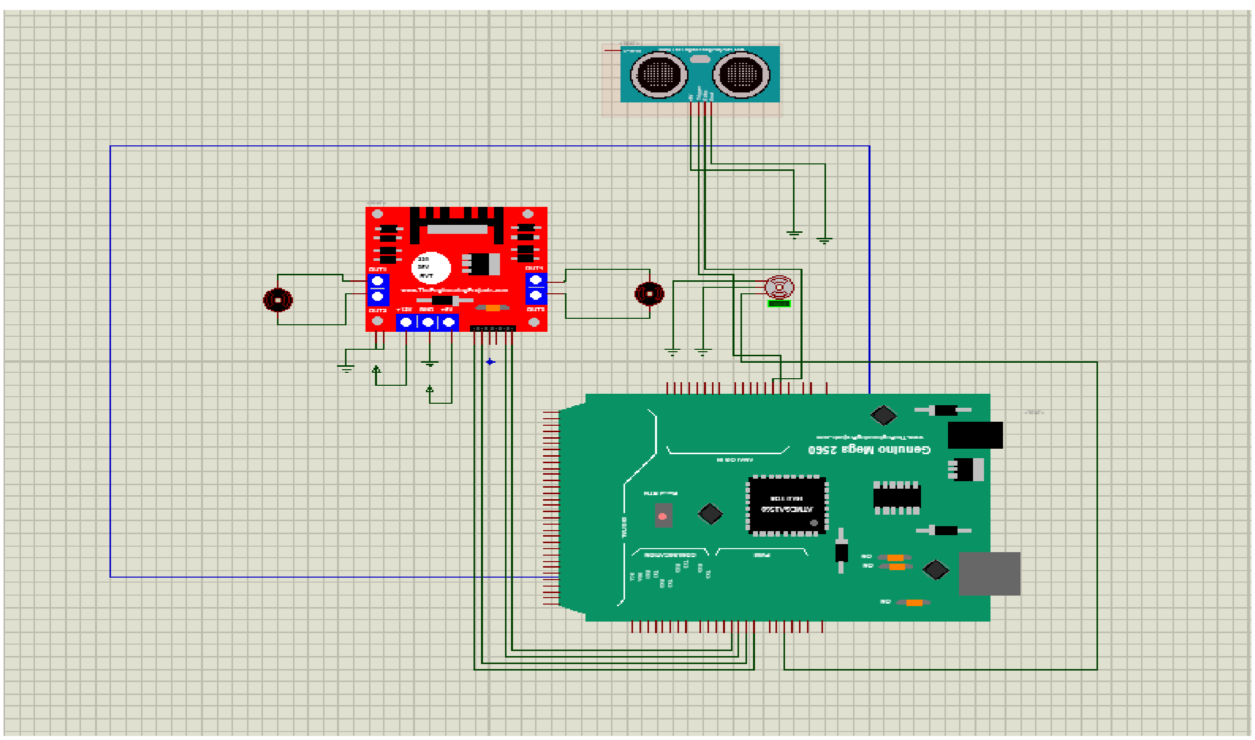


Fig. 2.2: Circuit Diagram

2.3.1 Working Process of the Circuit

Obstacle Avoiding Robot Using Ultrasonic Sensor Arduino. The circuit comprises of Arduino Mega2560 board, L298N H-Bridge and DC Motor, 9g Servo Motor, Ultrasonic Sensor and a few other components. Dual Motor Controller Module 2A with Arduino. This allow to control the speed and direction of two DC motors. "The L298N Bridge was connected to the Arduino digital pins 4 through 7. 5 Volts for the L298N board was now being supplied from the Arduino 5 Volt output". "The Trigger pin will be used to send the signal and the Echo pin will be used to listen for returning signal". The 5V pin of the ultrasonic sensor was connected to the 5Volt pin on the board, the GND pin was connected to the GND pin, and the SIG (signal) pin was connected to digital pin 10 on the board.

2.4 List of Components used in Circuit

Costing was always a vital issue to make any project. Price of electronics is not stable for a developing country like Bangladesh. Because Bangladesh never produces electronics parts but import from other developed country and during import price depends upon the stock of foreign currency. Average price of parts used in this project is given bellow,

Table: 2.1 Components Name of Circuit

No	Component Name	Quantity	Uses
1.	ATmega2560 (Arduino UNO)	**01**	To Control the System.
2.	L298N H-Bridge	**01**	To inputted number.
3.	DC Motor Driver	**01**	To showing message.
4.	5v DC adapter	**01**	Power Supply.
5.	9g Servo Motor	**02**	To view output.
6.	Ultrasonic Sensor	**01**	To vary power.
7.	Jumper Wires	**20**	To connecting.
8.	Breadboard	**01**	Temporary electrical connection**.**
9.	2WD Chassis	**01**	Robot chassis platform.

2.5 Conclusion

The cost involved in developing the system is significantly low and is much less than the cost of commercially available in the market.

CHAPTER 3

COMPONENT DESCRIPTION

3.1 Introduction

System hardware design composed of Arduino Mega2560, L298N H-bridge, HC SR04 Sensor, DC Motor Driver and SG90 Servo. In this chapter I will discuss about component description, features, working procedure and cost analysis of my all component.

3.2 Basic Description of controller unit

In the controller unit, I was using Arduino Mega2560 hardware board (with AVR microcontroller). With the help of Arduino1.6.8 software plate form. "I can easily program AVR IC, as my requirement. Arduino was an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software". It was intended for performers, originators, hobbyists, and anyone interested in creating interactive objects or environments. "The microcontroller board was programmed using the Arduino programming language (based on Wiring) and the Arduino development setting (based on Processing) ".

"The Arduino Mega 2560 was a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button". "It has everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started".

3.2.1 Description of Microcontroller: ATmega2560

The Arduino Mega 2560 was a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), 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 an AC-to-DC adapter or battery to get started"**.** You can tinker with your UNO short of disturbing too much about doing something mistaken, worst case situation you can replace the chip for a few dollars and start over again.

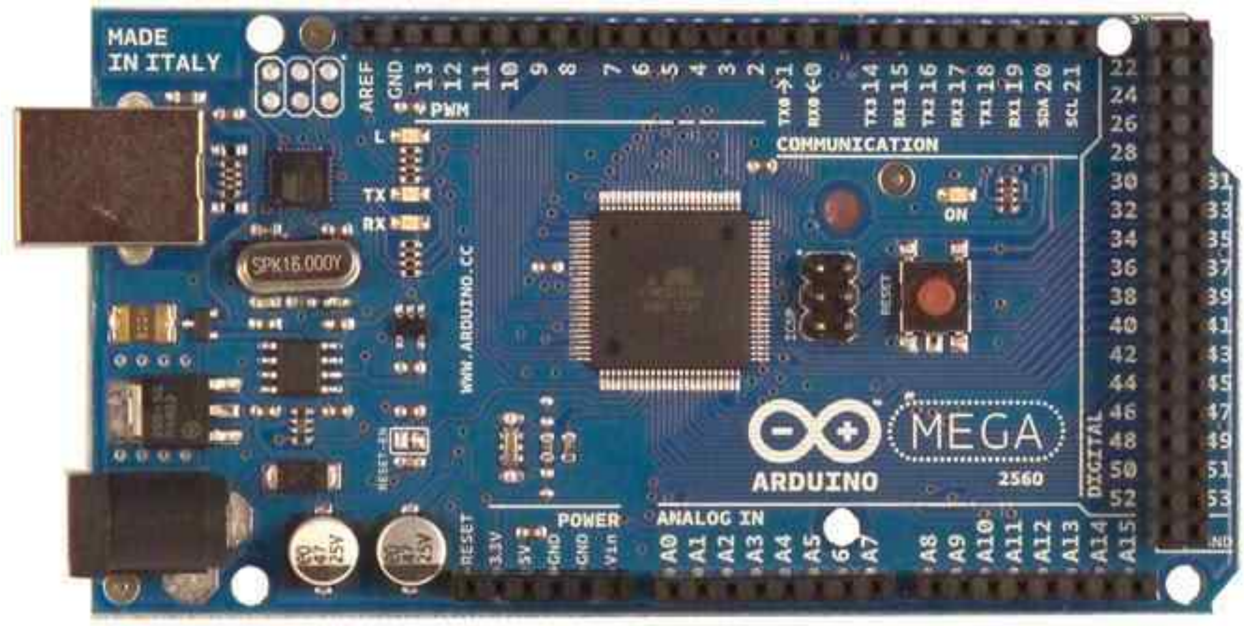


Fig 3.1: Arduino Mega 2560 [2]

"Uno" means one in Italian and is chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. "The Uno board was the first in a series of USB Arduino boards, and the position model for the Arduino stage; for an extensive list of current, past or obsolete boards see the Arduino index of boards".

Arduino was a microcontroller kit for building digital devices and communicating objects that can sense and control physical procedures. "These systems provide set of digital and analog pins that can interface to various expansion boards". "The first Arduino was familiarized in 2005 aiming to provide a low cost, cost in effect device that interacts with environments using sensors and actuators". "An Arduino's microcontroller can be preprogrammed with a boot loader that simplifies the uploading of programs to the on-chip flash memory". Arduino boards were planned by American enterprises. As on 2016, 17 classes of Arduino hardware had been commercially formed. [1]

3.2.2 Specification Arduino Mega2560

Arduino is an open-source prototyping stage based on easy-to-use hardware and software. Arduino consists of both a physical programmable circuit board and a piece of software, or IDE (Integrated Development Situation) that runs on my computer, used to write and upload computer code to the physical Arduino board.

Table No 3.1: Specification Arduino Mega2560 [3]

Microcontroller	ATmega2560
Operating Voltage	5Voltage
Input Voltage (recommended)	7-12Voltage
Input Voltage (limit)	6-20Voltage
PWM Digital Input/output Pins	15
Digital Input/output Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per Input/output Pins	20 mA
DC Current for 3.3Voltage Pins	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM(Static random-access memory)	8 KB (ATmega2560)
EEPROM((Electrically Erasable Programmable Read-Only Memory)	4 KB (ATmega2560)
Clock Speed	16 MHz
LED BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

3.2.3 Block Diagram of Microcontroller – (ATmega2560)

Internal block diagram of Microcontroller.

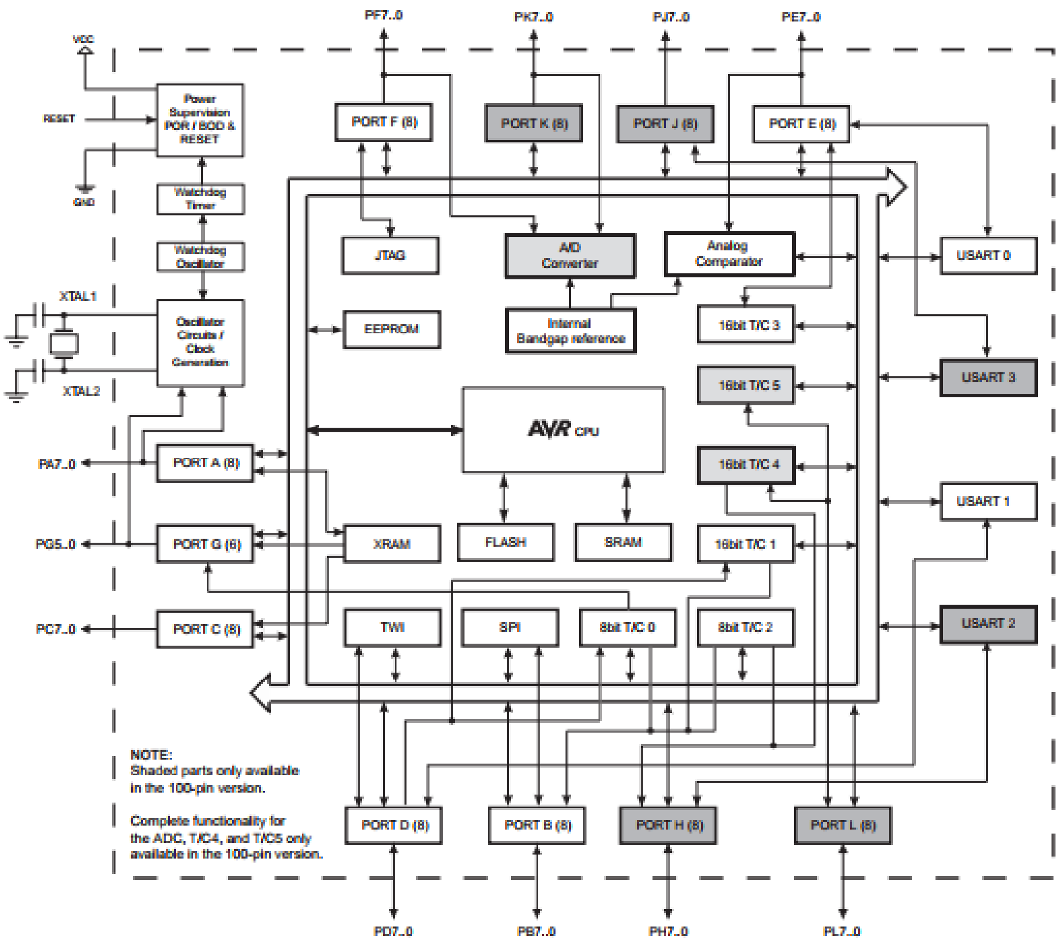


Fig. 3.2: Block Diagram of Microcontroller – (ATmega2560) [4]

3.2.4 Pin Configurations of Microcontroller – (ATmega2560)

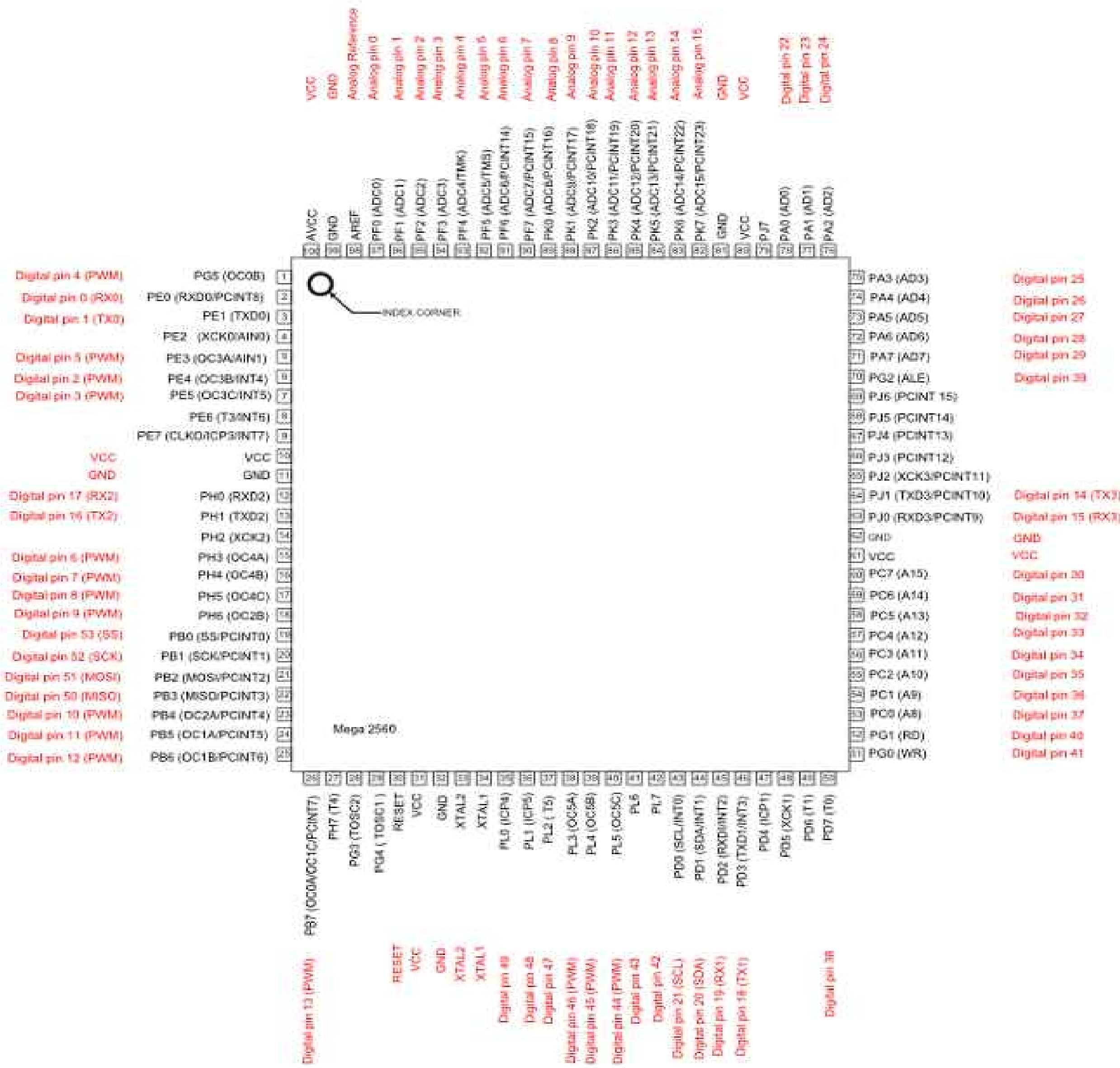


Fig. 3.3: Pin Configurations of Microcontroller – (Atmega2560) [5]

3.2.5 Arduino Mega 2560 PIN mapping table

Table No 3.2: Arduino Mega 2560 PIN mapping table [6]

Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pins 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pins 0 (RX0)
3	PE1 (TXD0)	Digital pins 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pins 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pins 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pins 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)

21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10
24	PB5 (OC1A/PCINT5)	Digital pin 11
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)

44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	

67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8

90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

3.2.6 Pin Descriptions

The below gives a description for each of the pins, along with their function:-

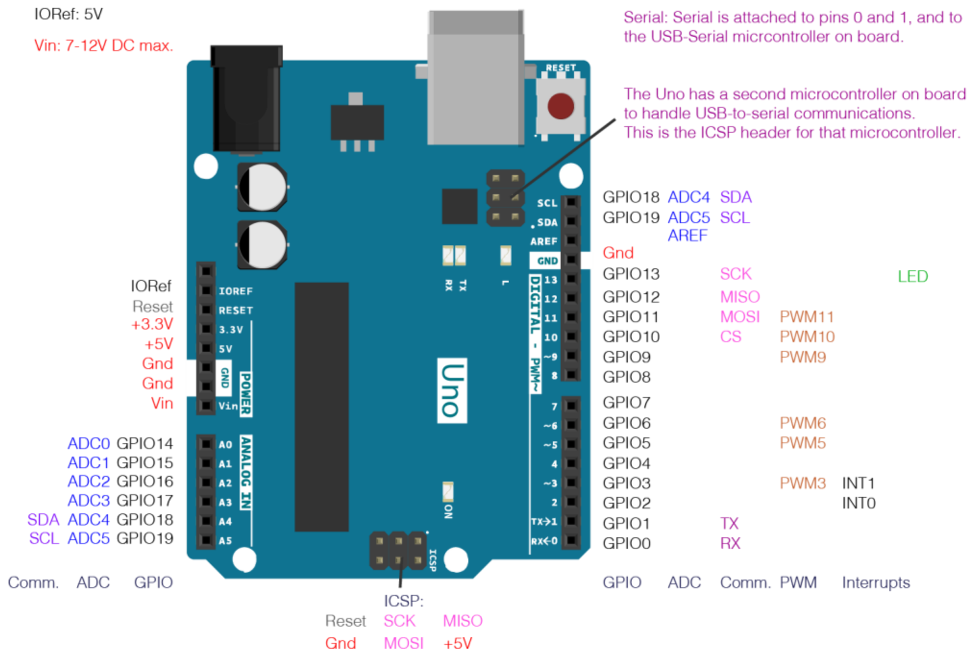


Fig 3.4: Pins of Arduino Mega2560. [7]

Programming

The Arduino Mega2560 "can be programmed with the (Arduino Software (IDE)). Select "Arduino/Genuine Uno from the Tools > Board menu (conferring to the microcontroller on my board) ".

The Arduino mega2560 on the Mega 2560 comes preprogrammed with a bootloader that permits my upload new code to it short of the use of an outside hardware programmer. "It connects using the unusual STK500 procedure (reference, C header files) ".”I can also avoid the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Program design) header using Arduino ISP or similar; see these advices for details. The Arduino mega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code was accessible in the Arduino source". The ATmega16U2/8U2 was loaded with a DFU bootloader, which can be triggered by:

•"On Rev1 boards: attaching the connect jumper on the back of the board (near the map of Italy) and then resetting the 8U2".

•"On Rev2 or later boards: there is a resistor that drawing the 8U2/16U2 HWB line to ground, constructing it easier to put into DFU mode".

"I 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). See this user-contributed tutorial for more information".

Warnings

"The Mega 2560 has a resettable polypus that look after my computer's USB ports from shorts and overcurrent". "Though most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is useful to the USB port, the fuse will mechanically break the connection until the short or overwork is removed".

Differences with other boards

"The Uno changes from all previous boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Arduino mega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter".

Power

•"The Mega 2560 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 become 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 board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source) ".”I can supply voltage through this pin, or, if supplying voltage via the power card, access it through this pin".

•5V. "this pin outputs a regulated 5V from the regulator on the board". "The board can be provided with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). In case voltage via the 5V or 3.3V pins avoids the regulator, and can damage my board. I don't advise it".

•3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

•GND. Ground pins.

•IOREF. "This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the suitable power source or enable voltage convertors on the outputs for working with the 5V or 3.3V".

Memory

"The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library) ".

Input and Output

"See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical".

Each of the 54 digital pins on the Mega can be used as an input or output, using pin Mode (), digital Write (), and digital Read () functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller. In addition, some pins have particular functions:

Serial:

"0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX) ".”Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip".

External Interrupts: "2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be arranged to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the attach Interrupt () function for details".

PWM: "2 to 13 and 44 to 46. Provide 8-bit PWM output with the analog Write () function".

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.

"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".

"TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards".

See also the mapping Arduino Mega 2560 PIN diagram.

"The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values) ".”By default they quantity from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analog Reference () function. There are a couple of other pins on the board":

AREF. Reference voltage for the analog inputs. Used with analog Reference ().

Reset. Bring this line LOW to reset the microcontroller. Normally used to add a reset button to shields which block the one on the board.

Communication

The Mega 2560 board has a number of services for communicating with a computer, alternative board, or other microcontrollers. "The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication". An ATmega16U2 (AT mega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .info file, but OSX and Linux machines will recognize the board as a COM port spontaneously. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Mega 2560's digital pins. The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

Rather than requiring a physical press of the reset key before an upload, the Mega 2560 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 ATmega2560 via a 100 Nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) 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 Mega 2560 board 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 ATMega2560.

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 connects waits a second after opening the connection and before sending this data.

The Mega 2560 board 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.

Revisions

Revision 3 of the board has the following new features:

- 1.0 pinout: SDA and SCL pins - near to the AREF pin - and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the board that uses ATSAM3X8E, that operate with 3.3V. The second one is a not connected pin that is reserved for future purposes.

- ➢ Stronger RESET circuit.
- ➢ Atmega 16U2 replace the 8U2. [7]

3.3 L298N H-bridge

The L298N dual H-Bridge motor driver which "agree to speed and pathway control of two DC motors at the same time". "The element can drive DC motors that have voltages between 5Voltage to 35Voltage, with a peak current up to 2A".

Fig 3.5: L298N Dual H-bridge [8]

3.3.1 Features of L298N H-bridge

- Popular L298N Dual H-Bridge Motor Driver chip.
- Drivers motors from 5-35V at up to 2A per channel.
- Provides 4 LEDs that reflect the state of the control logic.
- Independent direction, speed and braking for each motor.
- Screw terminals for easy connections to motors and power.
- Includes a heavy duty heat sink for maximum performance.
- Easy to interface with most robot controllers.
- Supports current sensing.
- Handy power LED.

3.3.2 Key Specifications

- Input voltage: 5-35V
- Output current: 2A per channel

- Control logic: standard 5V TTL
- Power consumption: 36ma for logic
- Size: 55mm * 60mm * 30mm
- Weight: 33g

3.3.3 How it Works?

An H-Bridge was a circuit that can drive a current in moreover polarity and be controlled by *Pulse Width Modulation (PWM).

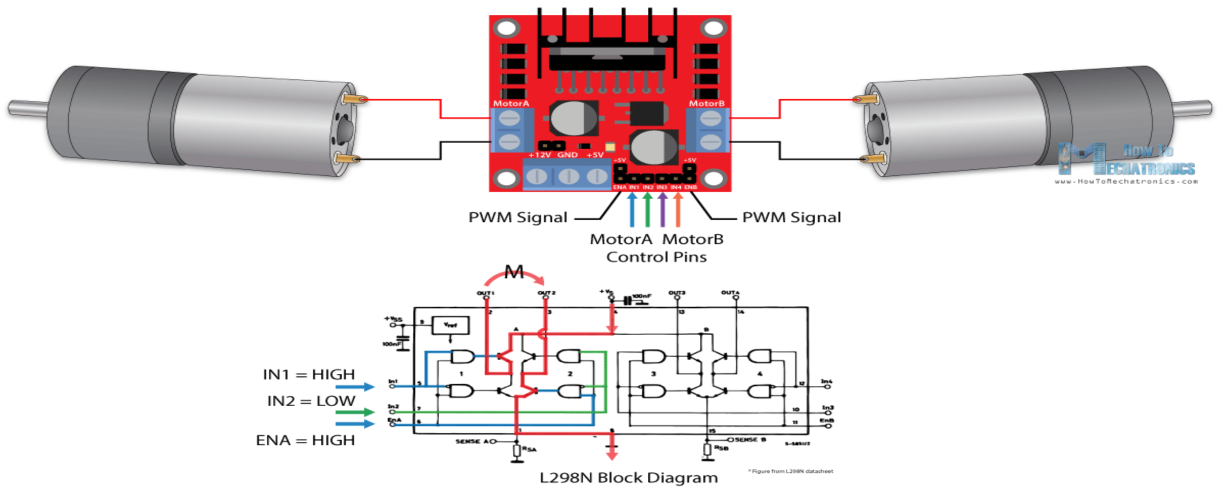


Fig 3.6: L298N Dual H-bridge interface [9]

* "PWM was a means in controlling the duration of an electronic pulse". In motors try to see the brush as a water wheel and electrons as the flowing droplets of water. The voltage would be the water flowing over the wheel at a constant rate, the more water flowing the higher the voltage.

"Motors are rated at certain voltages and can be damaged if the voltage is applied to heavily or if it is dropped quickly to slow the motor down". "Thus PWM. Take the water wheel analogy and think of the water hitting it in pulses but at a constant flow". The longer the pulses the faster the wheel will turn, the shorter the pulses, the slower the water wheel will turn. Motors will last much longer and be more reliable if controlled through PWM. [9]

3.4 Power Requirements

The adapter would be powered with an external power supply that can deliver current between 700mA and 1000mA. "Powering an Arduino Mega2560 and the L298N H-Bridge from a USB connection is not suggested, as USB cannot provide the essential current when the modem is in heavy use". "So instead I have to use 5Voltage connecter. The modem can pull in to 2A of current at peak usage, which can happen during data transmission".

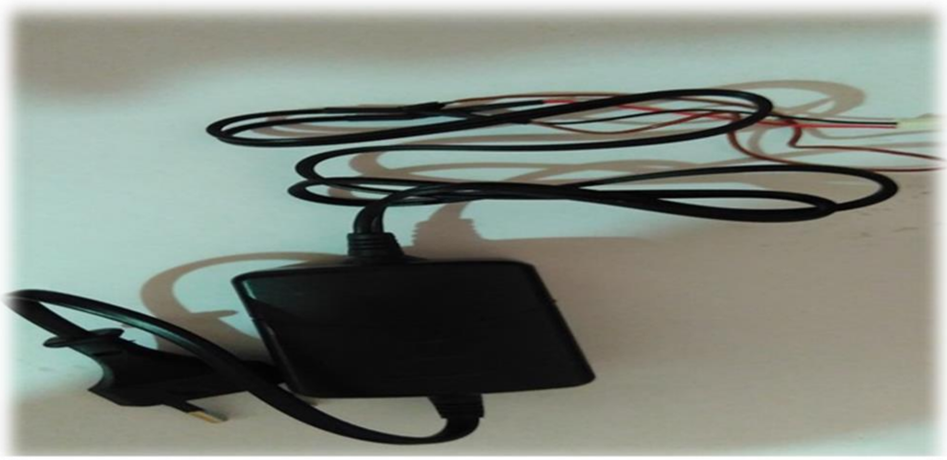


Fig.3.7: 5v Adapter

3.5 Description of HC SR04 Sensor

"The HC-SR04 ultrasonic sensor uses sonar to determine space to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package". From 2cm to 400 cm or 1” to 13 feet. "It’s process is not affected by daylight or black material like high-pitched rangefinders are (while acoustically soft tools like cloth can be testing to detect). It comes complete with ultrasonic transmitter and receiver module". [10]

Fig. 3.8: HC SR04 Ultrasonic Sensor [10]

3.5.1 Features of HC SR04 Sensor

- Power Supply :+5V DC
- Quiescent Current : <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance : 2cm – 400 cm/1″ – 13ft
- Resolution : 0.3 cm

- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm

3.5.2 HC-SR04 Ultrasonic Sensor – Working

The HC-SR04 Ultrasonic (US) sensor was a 4 pin element, whose pin names are Vcc, Trigger, Echo and Ground separately. "This sensor was a very common sensor used in many applications where measuring distance or sensing objects are required". "The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver". The sensor works with the simple high school formula that.

Distance (d) = Time (t) × Speed (s)

Fig. 3.9: HC-SR04 Ultrasonic Sensor

"Now, to calculate the distance using the above formulae, I would know the Speed and time". [10]

Ultrasonic Sensor Pin Configuration:

Table No 3.2: Ultrasonic Sensor Pin Configuration

Pin Number	Pin Name	Description
1	**Vcc**	Vcc pin powers the sensor, normally with +5Voltage.
2	**Trigger**	Trigger pin is an Input pin which is connected to Arduino analog pin A1. "Trig pin has to be kept high for 10us to initialize measurement by sending US wave".
3	**Echo**	Echo pin is an Output pin which is connected to Arduino analog pin A2. "Echo pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor".
4	**GND**	GND pin is joined to the Ground of the structure.

3.5.3 SG90 Servo

"Small and lightweight with high output power. Servo can rotate almost 180 degrees (90 in each direction), and workings just like the normal kinds but smaller".

Fig. 3.10: SG90 Servo [11]

SG90 Servo Wire Configuration

Table No 3.3: SG90 Servo Wire Configuration

Wire Number	Wire Color	Description
1	Brown	GND cable connected to the ground of system
2	Red	Powers the motor normally +5Voltage is used
3	Orange	"Pulse Width Modulation signal is given in through this wire to drive the motor in connected to Arduino 10 pin".

SG-90 Servo Features:

- Operating Voltage is +5Voltage typically
- Torque: 2.5kg/cm
- Operating speed is 0.1s/60°
- Gear Type: Plastic
- Rotation : 0°-180°
- Weight of motor : 9gm
- Package includes gear horns and screws

3.6 DC Motor and Wheel

This was a DC motor and wheel set for building robots. "These vehicles were light weight, high torque and low RPM". "They can increase hills and have outstanding traction, plus you can mount the wheel on moreover side of the motor with its double-sided output shaft".

Fig 3.11: DC Motor Wheel

- Wheel and DC motor-powered kit.
- Motor speed: 100 r / min.

- Torque: 4.5KG. CM1.
- Voltage: 3Voltage - 6Voltage.

3.7 2WD Chassis

"2WD kit provides us everything I need to build the shell of a 2-wheel-drive Platform Mechanical device! I get the metal plates that make up the chassis, two DC drive motors with matching wheels, and a caster ball for stability".

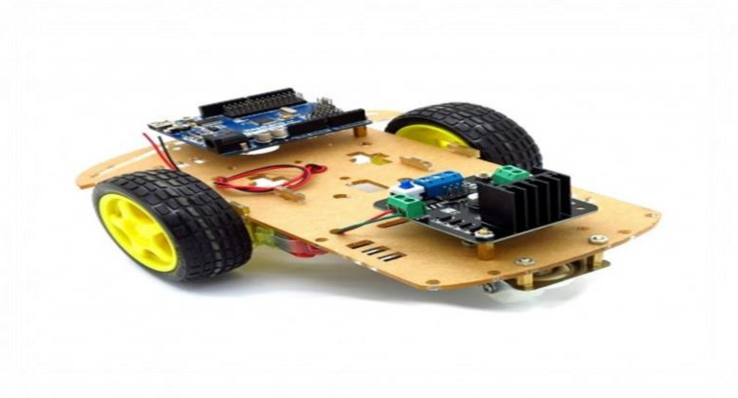


Fig 3.12: 2WD Chassis

3.8 HC SR04 Bracket

- Width :2.8mm - 3.1mm
- Inner length: 16mm
- Diameter of growing shack: 3.8mm
- Quantifiable: acrylic
- Usage: fixed ultrasonic sensor modules and other products

Fig 3.13: HC SR04 Bracket

3.9 Cost Analysis

In this section I will show cost of my project that means cost sheet representation of my project.

3.9.1 Cost Sheet

Table No 3.4: Obstacle Avoiding Robot Using Ultrasonic Sensor Arduino

No	Component Name	Quantity	Purchase Price (TK)
1.	Arduino UNO	1	1000.00
2.	DC Motor	2	1000.00
3.	Micro Servo (SG90)	1	160.00
4.	Motor driver L298N	1	170.00
5.	Wires(m- m),(m-f)	1	40.00
6.	Breadboard	1	100.00
7.	Battery	1	300.00
8.	DC Motor Wheel	3	400.00
9.	Others		1000.00
	Total Cost		**=4170.00**

Comparison:

My all components were available in market. Received all components were very reasonable price.

3.10 Conclusion

Some tools were used in this system to makes it. My all component was very simple & available in my country market.

CHAPTER 4

SOFTWARE ANALYSIS

4.1 Introduction

"In this chapter the software used and the language in which the program code well-defined is declared and the program code putting implements are explained. The chapter also documents the development of the program for the application".

4.2 Description of the Software

"The open-source Arduino mega2560 setting makes it easy to write code and upload it to the Input/output board". "It runs on Windows". The environment is written in Java and based on Processing, avr-gcc, and other open source software. "The screen shot of Arduino 1.0.6 is shown below".

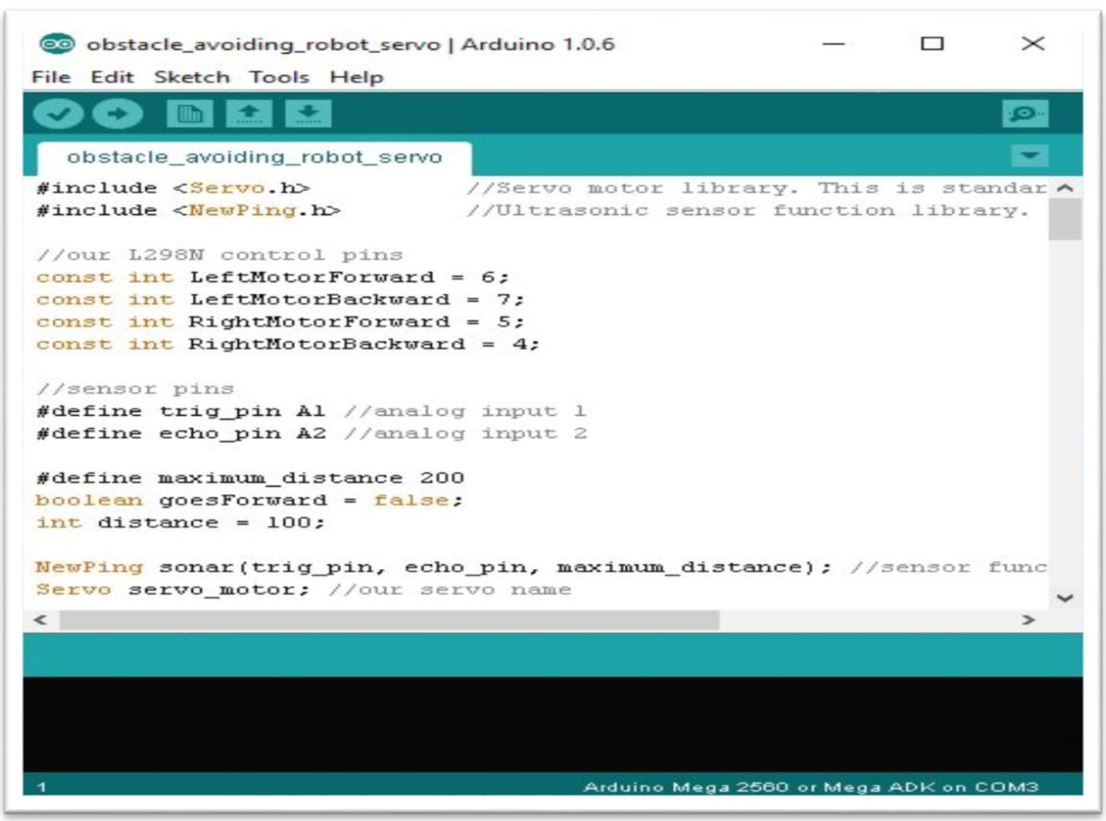


Fig. 4.1: Software Platform

It was also accomplished of compiling and uploading programs to the board with a single click. "There was normally no need to edit make files or run programs on a command-line interface". "Although building on command-line was possible if essential with some third-party tools such as Ino".

"The Arduino comes with a C/C++ library called "Wiring" (from the project of the same name), which makes many common input/output processes much easier. Arduino programs were written in C/C++, although operators only need explain two functions to make a runnable program":

The compiled window of my code is shown below.

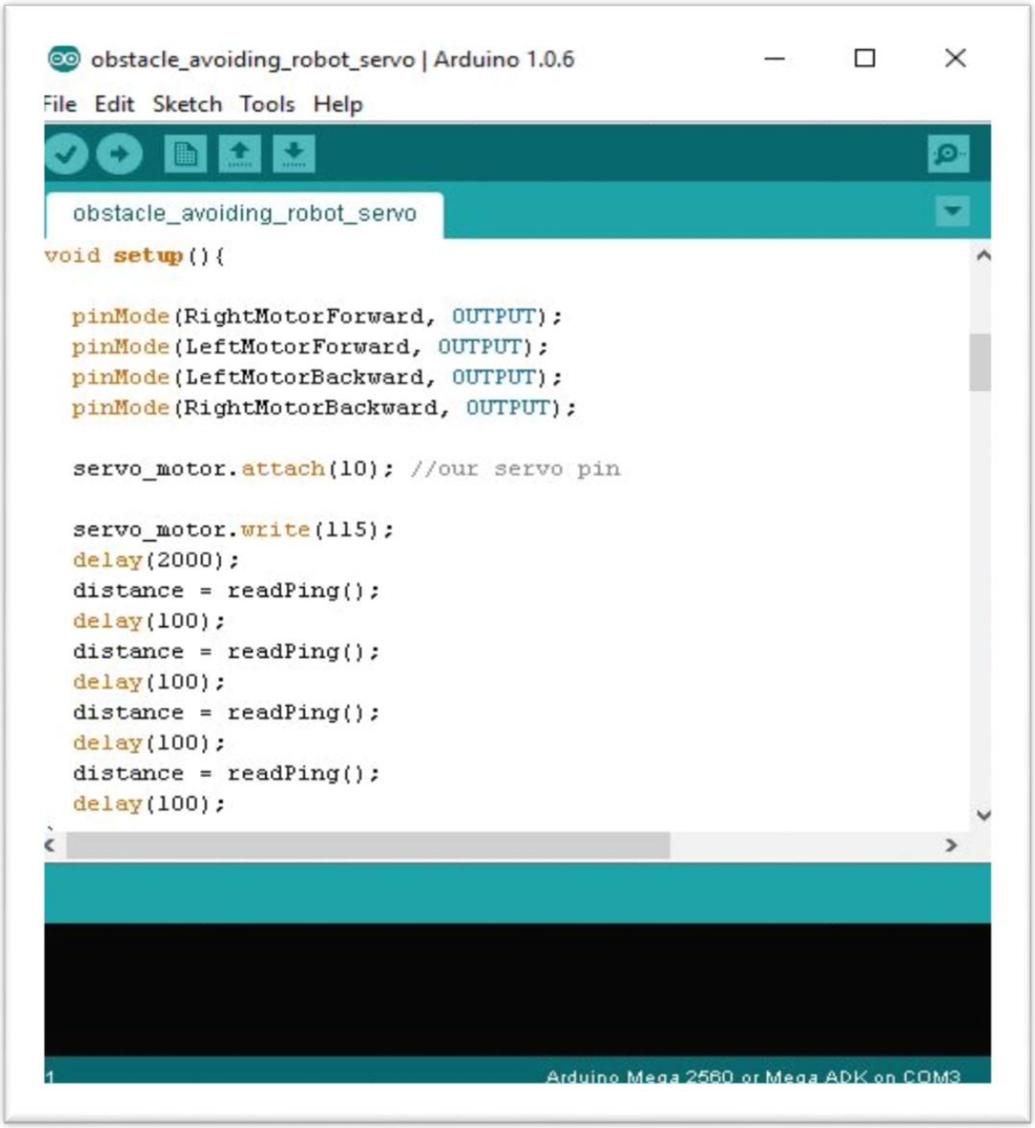
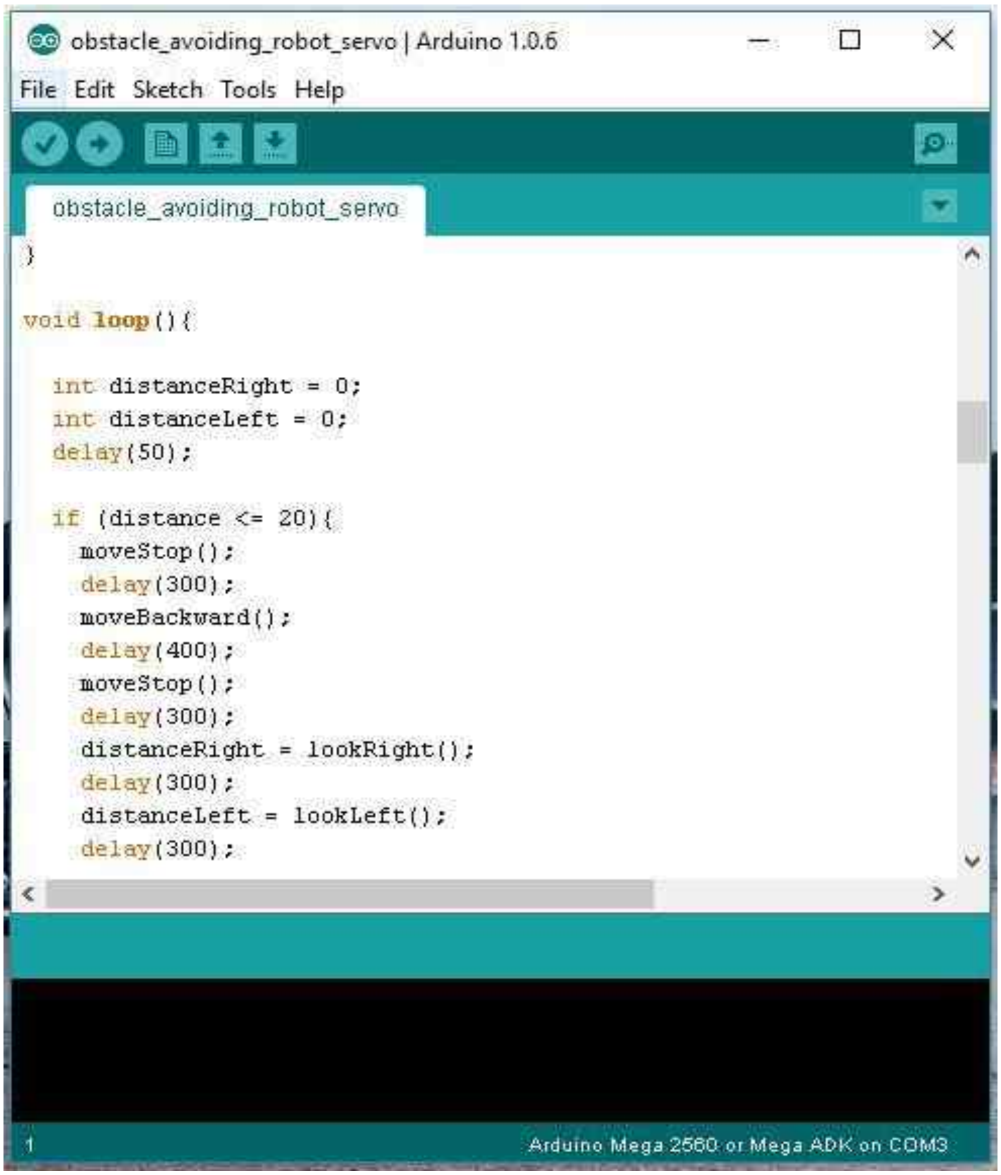
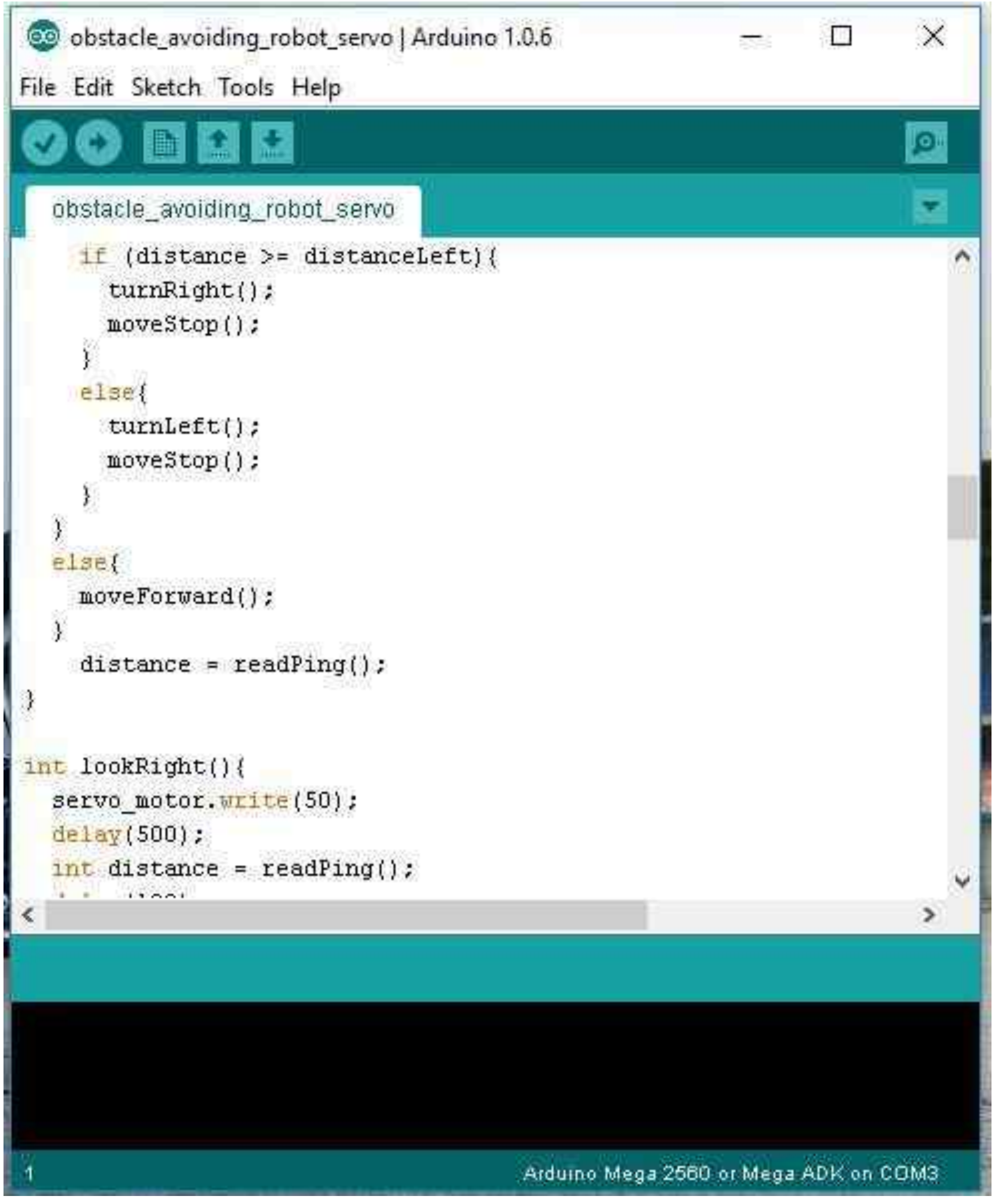
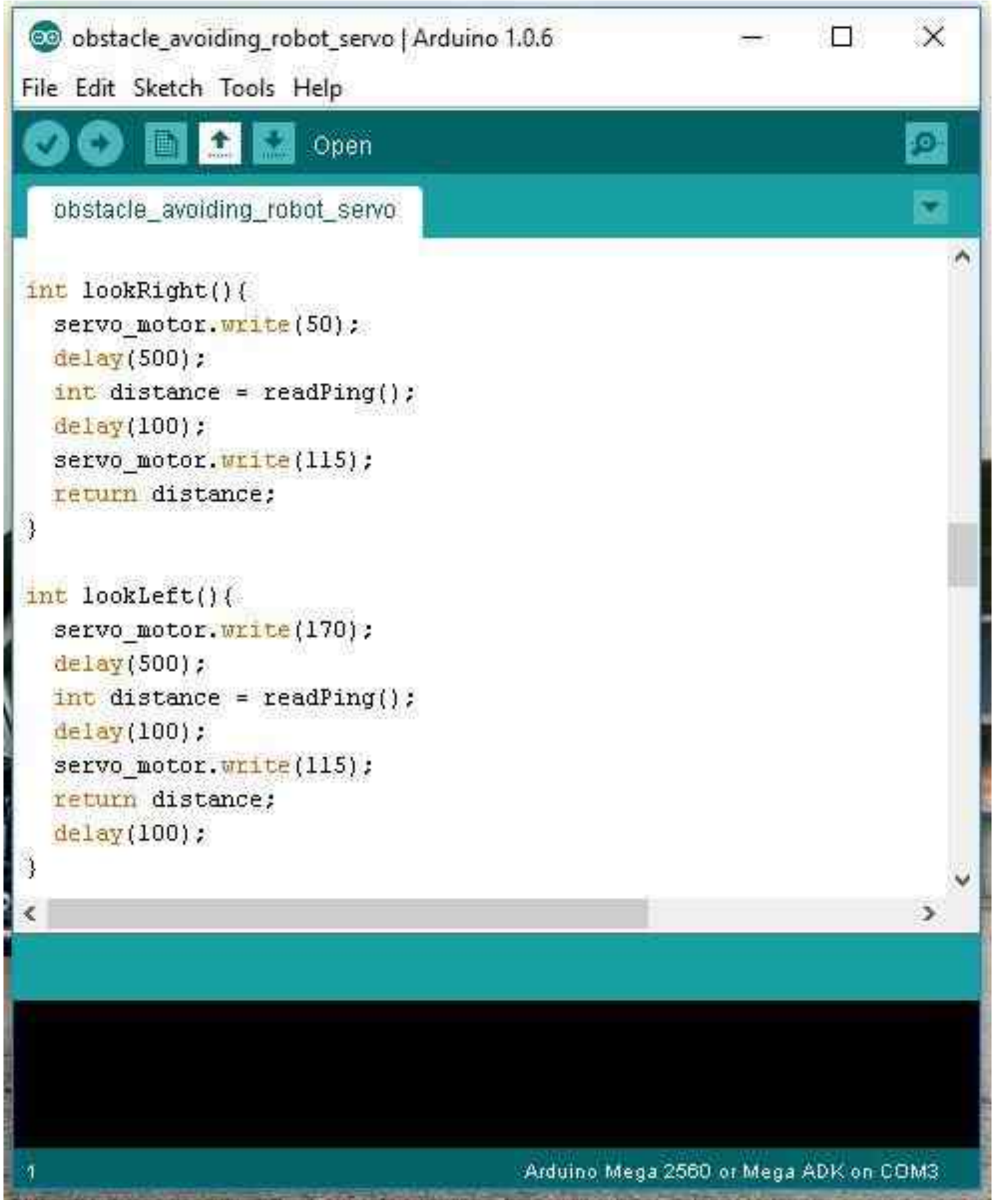


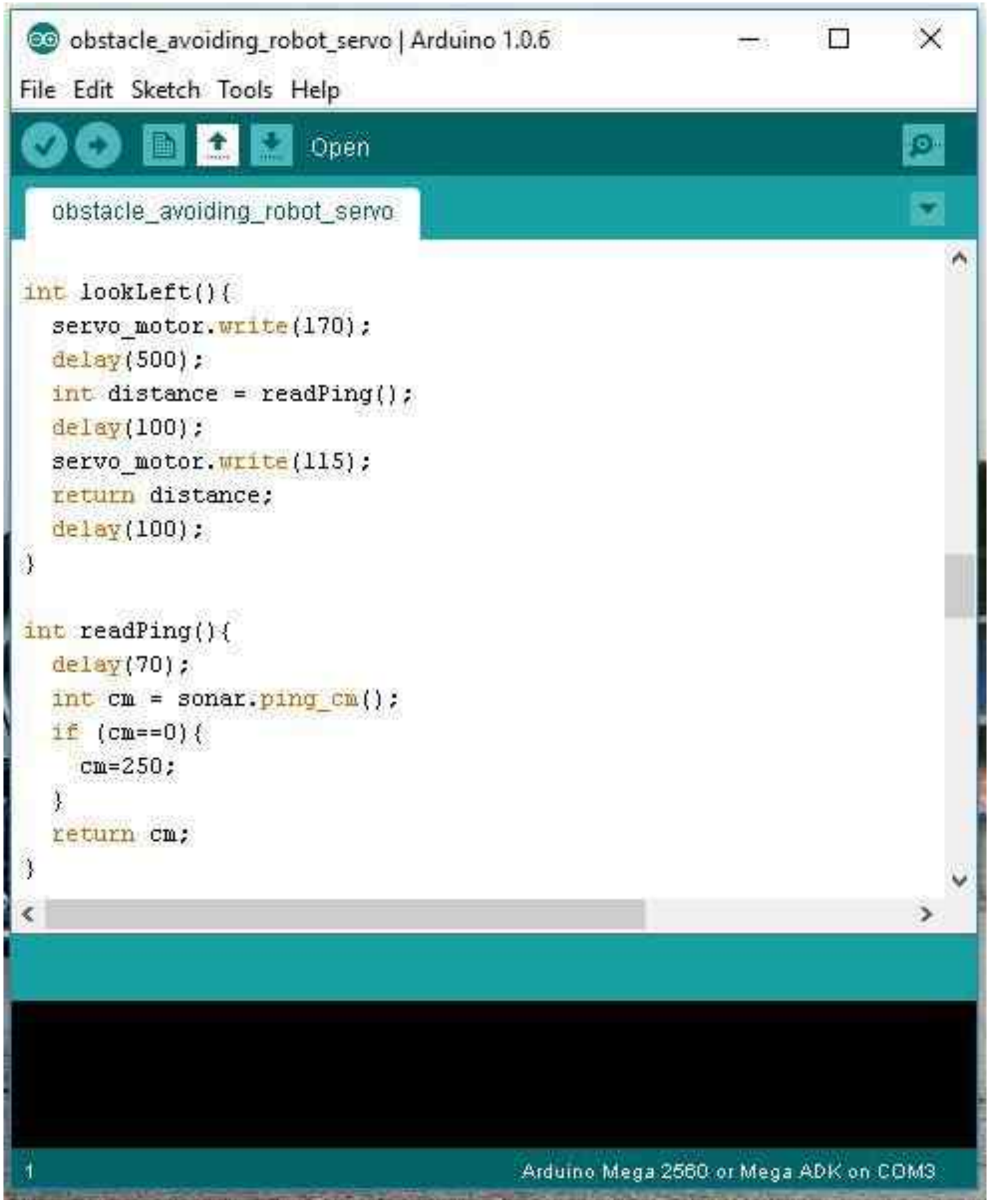
Fig. 4.2: Compiling window

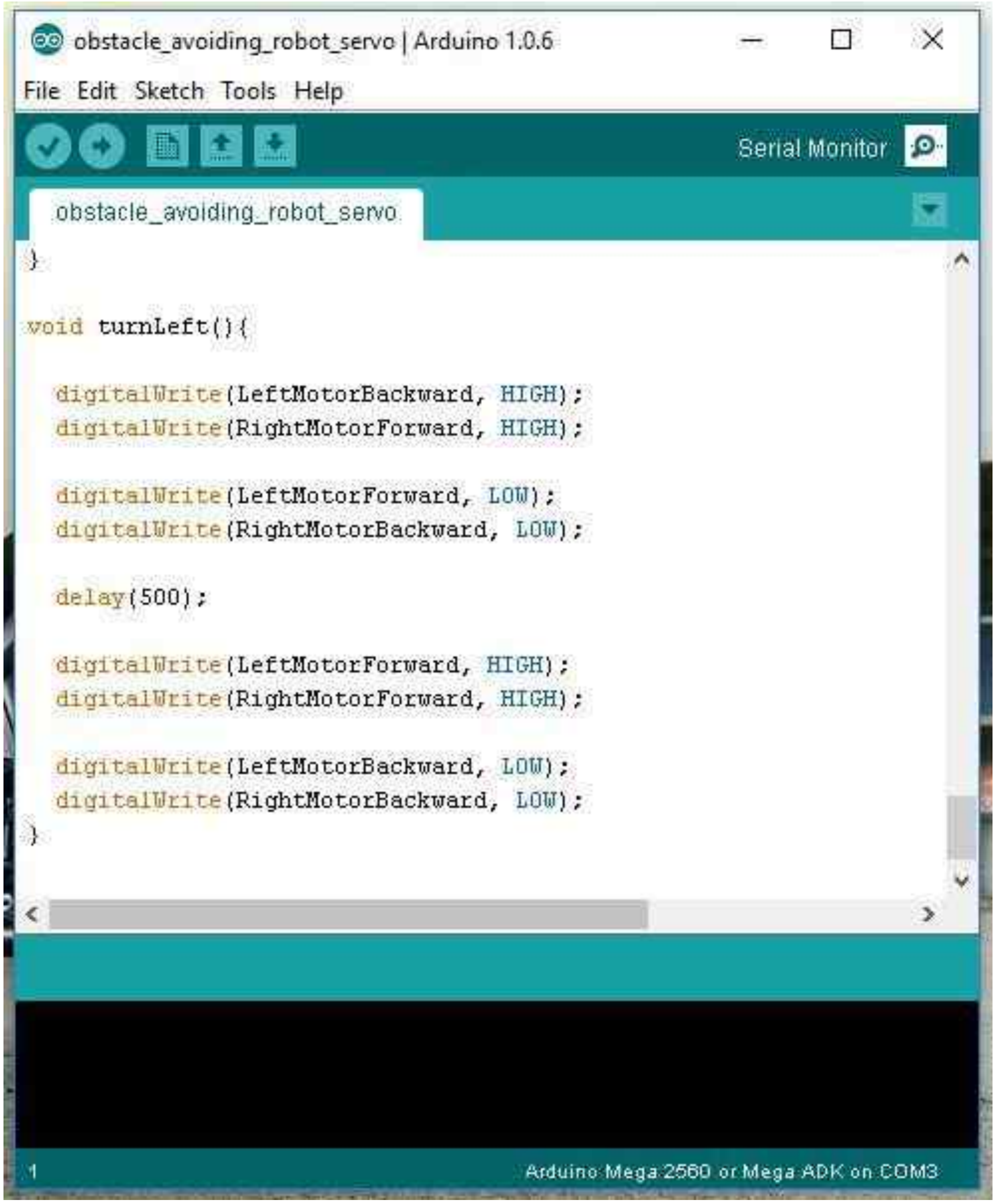
Compiling window show that L298N Dual H-Bridge, Ultrasonic Sensor & Servo motor was connected to the Arduino Pin. "The driver module was created on L298N H-bridge, a high current, high voltage dual full bridge driver. It can drive up to 2 DC motors up to 2Amp each, or drive one stepper motor or 2 solenoids". "The driver can control both motor RPM and direction of rotation". "The RPM was orderly using PWM input to ENA or ENB pins, while of round direction is controlled by providing high and low signal to EN1-EN2 for the primary motor or EN3-EN4 for another motor".




obstacle_avoiding_robot_servo | Arduino 1.0.6
File Edit Sketch Tools Help
obstacle_avoiding_robot_servo
if (distance >= distanceLeft){
turnRight();
moveStop();
}
else{
turnLeft();
moveStop();
}
}
else{
moveForward();
}
distance = readPing();
}
int lookRight(){
servo_motor.write(50);
delay(500);
int distance = readPing();
1
Arduino Mega 2560 or Mega ADK on COM3


obstacle_avoiding_robot_servo | Arduino 1.0.6
File Edit Sketch Tools Help
Open
obstacle_avoiding_robot_servo
int lookRight(){
servo_motor.write(50);
delay(500);
int distance = readPing();
delay(100);
servo_motor.write(115);
return distance;
}
int lookLeft(){
servo_motor.write(170);
delay(500);
int distance = readPing();
delay(100);
servo_motor.write(115);
return distance;
delay(100);
}
1
Arduino Mega 2560 or Mega ADK on COM3


obstacle_avoiding_robot_servo | Arduino 1.0.6
File Edit Sketch Tools Help
Open
obstacle_avoiding_robot_servo
int lookLeft(){
servo_motor.write(170);
delay(500);
int distance = readPing();
delay(100);
servo_motor.write(115);
return distance;
delay(100);
}
int readPing(){
delay(70);
int cm = sonar.ping_cm();
if (cm==0){
cm=250;
}
return cm;
}
1
Arduino Mega 2560 or Mega ADK on COM3


obstacle_avoiding_robot_servo | Arduino 1.0.6
File Edit Sketch Tools Help
Serial Monitor
obstacle_avoiding_robot_servo
}
void turnLeft(){
digitalWrite(LeftMotorBackward, HIGH);
digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(RightMotorBackward, LOW);
delay(500);
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorBackward, LOW);
}
1
Arduino Mega 2560 or Mega ADK on COM3

4.3 Flow Chart Diagram

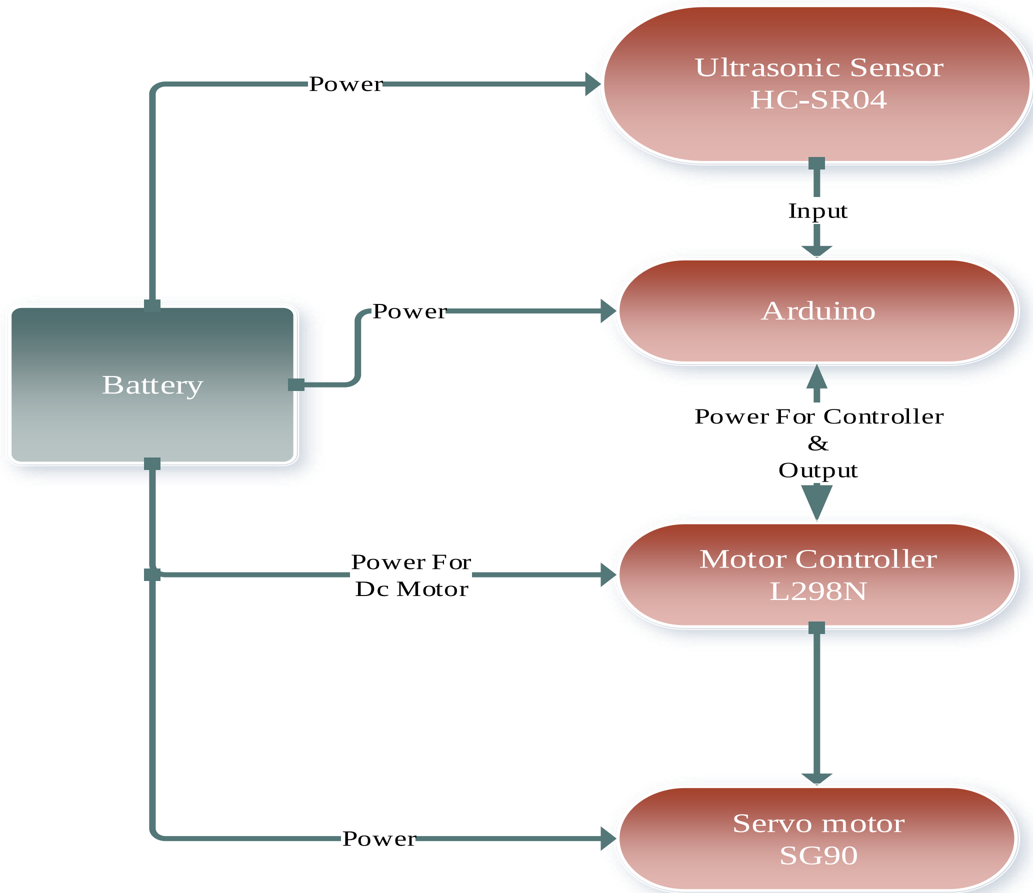


Fig. 4.3: Flow chart of the system

4.4 Conclusion

This project developed an obstacle avoiding Robot to detect and avoid obstacles in its path. The robot is built on the Arduino platform for data processing and its software counterpart helped to communicate with the robot to send parameters for guiding movement. The robot is fully autonomous and after the initial loading of the code, it requires no user intervention during its operation. When placed in unknown environment with obstacles, it moved while avoiding all obstacles with considerable accuracy. In future, the authors of this project intend to test the feasibility of integrating different types of sensors to complement each other’s disadvantages

CHAPTER 5

HARDWARE IMPLEMENTATION

5.1 Introduction

This is one of the most important chapters of this report. In this chapter I will show my completed project's outlook that means Connection layout & operation representation of my project. I will discuss about component's interfacing with Arduino Uno of my system.

5.2 Interfacing of L298N H-bridge

Interface L298N H-bridge to Arduino Mega2560 board was the main part of this project. "Most of the electronics devices use them as user inputs". "Significant how to connect L298N H-bridge to a microcontroller like Arduino was much respected for building marketable products".

Here 2 DC Motor Driver, 1 L298N H-Bridge, Arduino Mega 2560 and 5V Battery.

- DC Motor Driver M1 was connected by OUT1 & OUT2 and DC Motor Driver M2 was connected by OUT3 & OUT4.
- ENA & ENB was connected to Arduino 7 & 4 pin.
- 5V & 5V was connected to Arduino 6 & 5pin.
- L298N H-bridge and Arduino GND was connected to negative GND.
- L298N H-bridge and Arduino 5V was connected to positive GND.
- L298N H-bridge two wire were connected to battery.

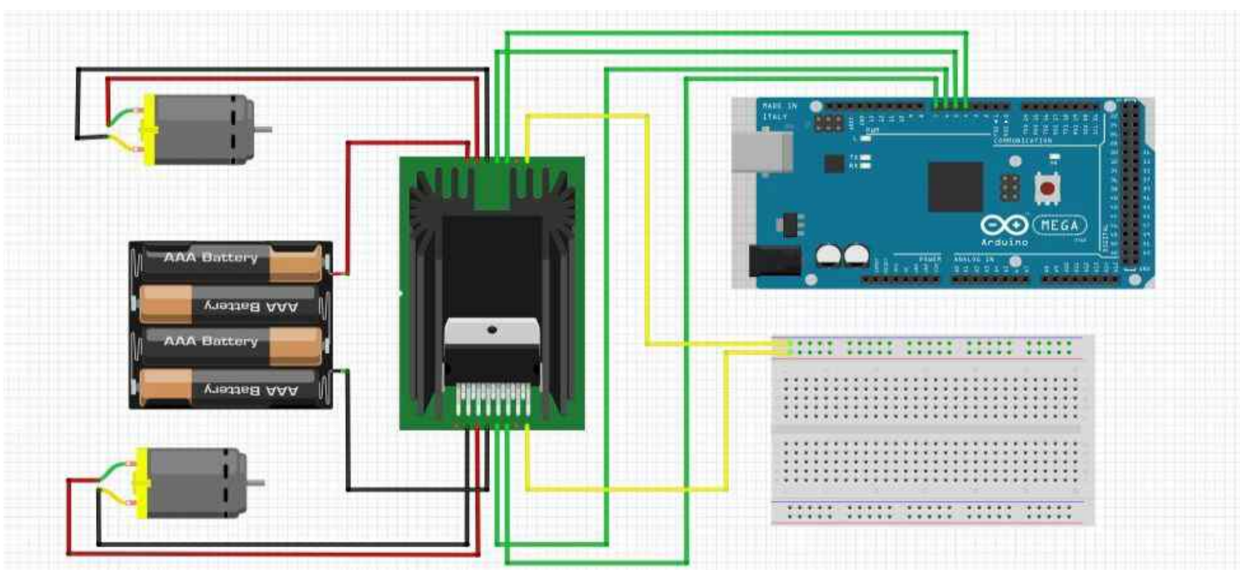
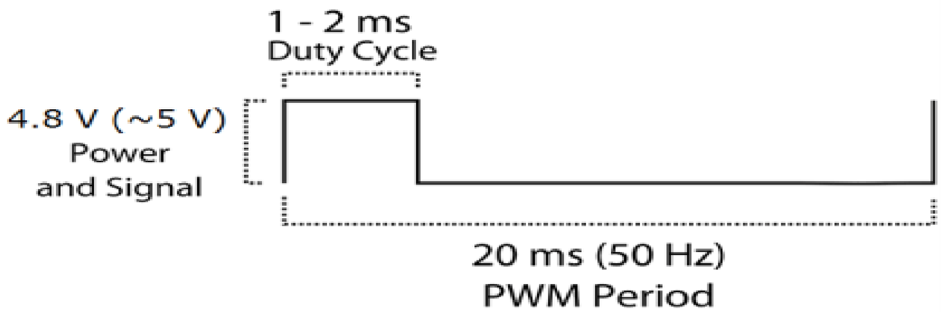


Fig. 5.1: Interfacing Of L298N H-bridge

5.3 SG90 Servo interfacing

SG90 Servo was connected with Arduino & Breadboard. Arduino pin number is 10 respectively on 5 volts power. "I had to connect the brown cable from the SG90 servo to the GND pin". Connect the red cable from the servo to the +5Voltage on Arduino board. "And to finish, attach the orange cable from the SG90 servo to a digital pin 10 on the Arduino mega2560".



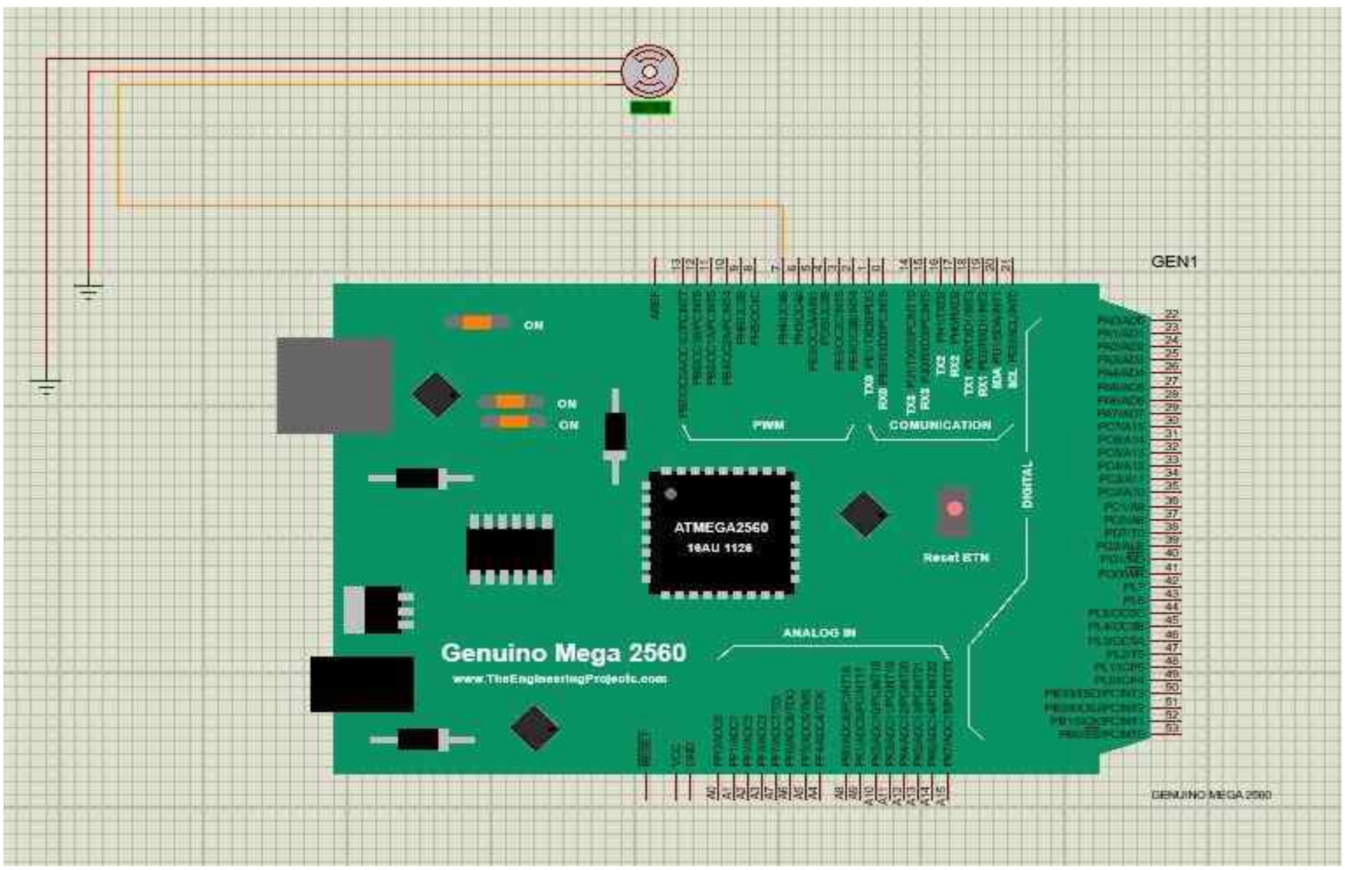


Fig. 5.2: Connecting SG90 Servo motor to Arduino Mega2560

5.4 Connection layout of the System

After the circuits had been devised and tested on Breadboard. Then I was soldering with lead. The circuit diagram shown in section is implementing on this board.

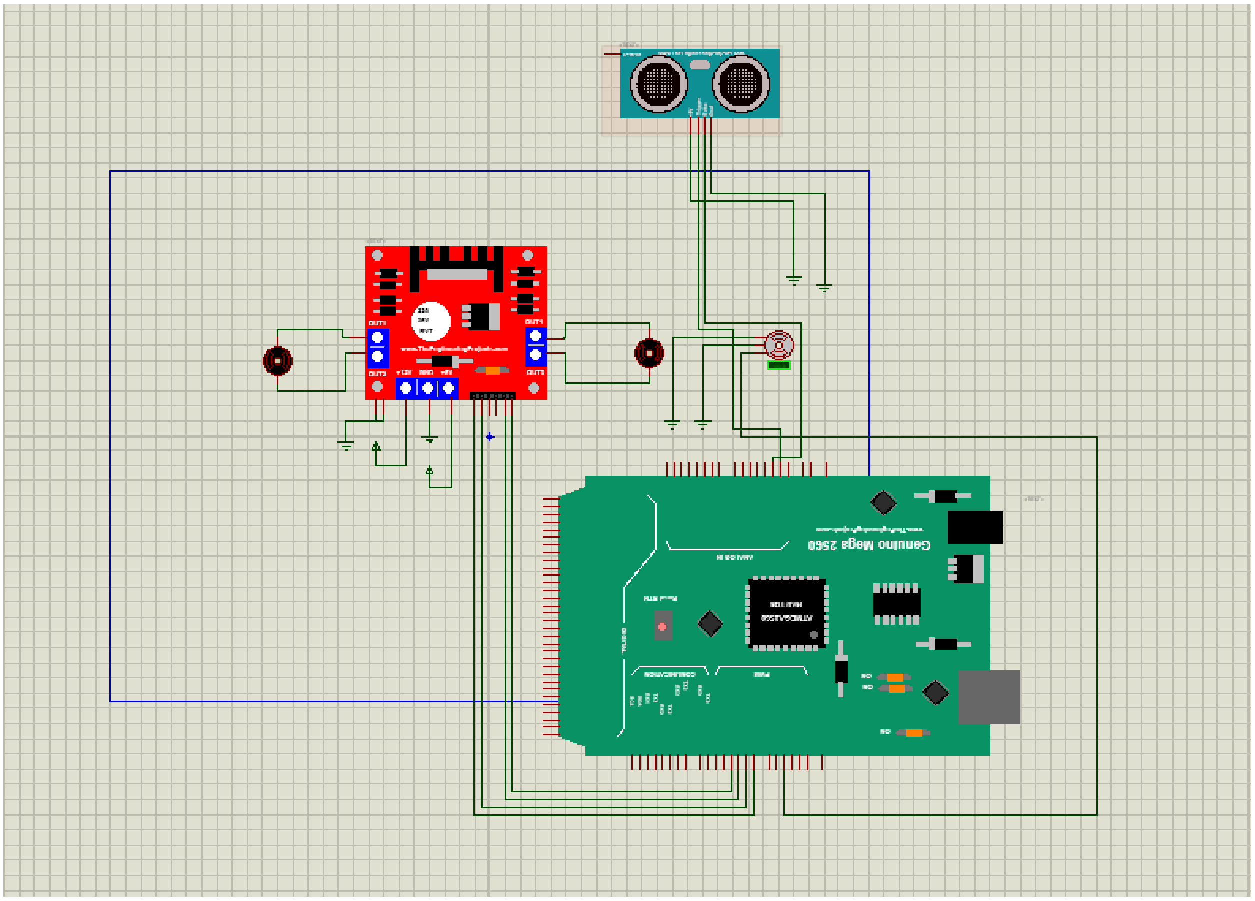


Fig 5.3: Project Design after Connected Components

5.5 System Operation

At the outset, "I was make the joining to the L298N H-Bridge. For joining the DC Motor Driver with the Arduino I was using both analog pins and digital pins. I used digital pins since I need more that 54 digitals pins for this development. Connect first four pins of L298N H-Bridge to digital pins 4, 5, 6 and 7".

"To connect the push pull solenoid with the Arduino mega2560, I will have to use outside power because it requires 6Voltage ~ 12Voltage to control and much more current than the Arduino can deliver. So to do that, I will use a DC power source which can provide 6Voltage ~ 12Voltage. Now connect the positive of power supply to the positive cable of solenoid and the negative of power supply to the ground. Now connect the positive wire of SG90 servo motor to the pin 10 of Arduino mega2560 and negative cable to the ground".

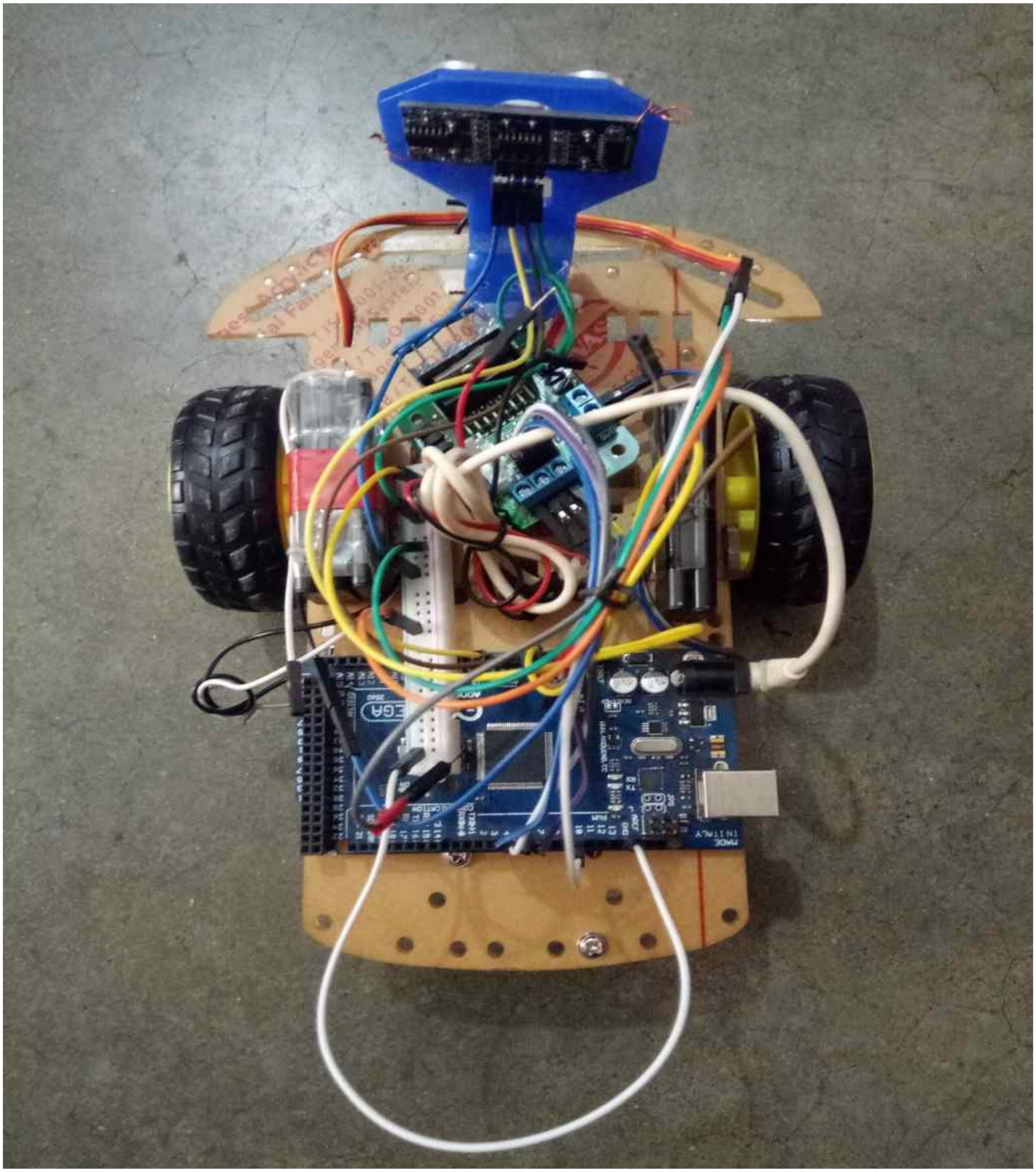


Fig 5.4: Outlook of the System

Now I will connect the L298N H-bridge to the Arduino.

- Connect pin 1 (ENA) to the Arduino 7 pin.
- Connect pin 2 (5v) to the Arduino 6 pin.

•Connect pin 3 (5v) to the Arduino 5 pin.

•Connect pin 4 (ENB) to the pin 4 of the Arduino.

•DC Motor Driver M1 was connected by OUT1 & OUT2.

•DC Motor Driver M2 was connected by OUT3 & OUT4.

•Servo Motor was connected to the Arduino 10 pin for PWM signals to the Orange color wire.

•Servo Motor was connected to the Breadboard have to power the motor with +5V using the Red wire.

•Servo Motor was connected to the Breadboard have to GND using Brown Wire.

The voltage used at the motors V_{cc}. The element have an onboard 5Voltage controller which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V I can enable the 5V regulator and the 5V pin can be used as output, say for powering my Arduino board. But if the motor voltage is greater than 12V I must cut off the jumper because those voltages will cause damage to the onboard 5V regulator.

5.6 Conclusion

To design, interfacing with Arduino Uno of the circuit was the simple persistence of my project. "My Project work was at present completed". "The constructed circuit was very kindly working".

CHAPTER 6

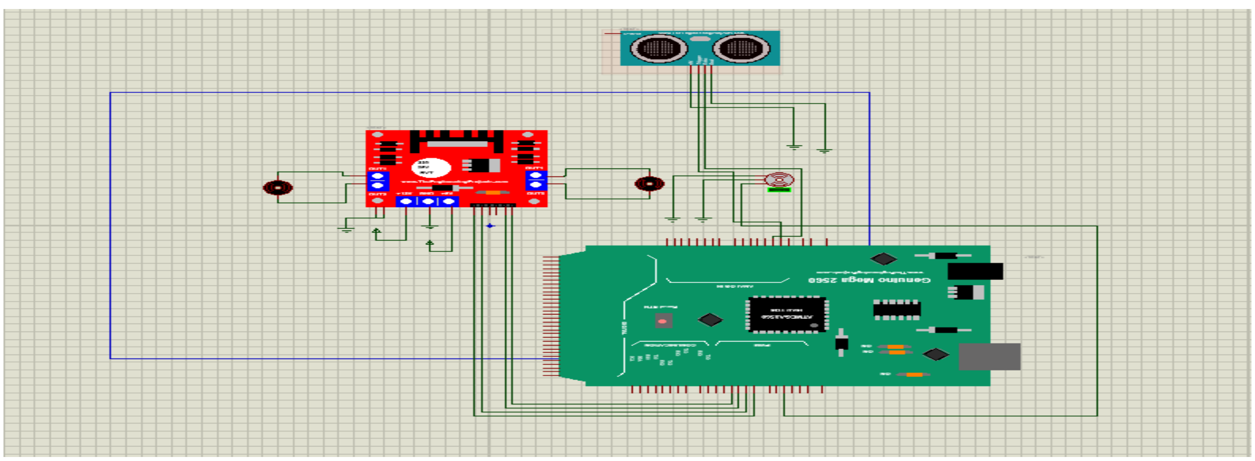
RESULTS AND DISCUSSIONS

6.1 Introduction

"The specific robotic device was an arrangement of both mechanical and electronics engineering. This method was made so that it can be used to sense and avoid obstacles successfully. This car will be just like an electronic car that possesses on moving in straight line until something comes in its path that's when it chooses to turns into some other direction so it successfully avoids the obstacle".

6.2 Experimental Setup

"The Arduino hardware developed contains of Microcontroller Arduino ATMEGA 2560, ultrasonic sensor, two DC motors and motor driver L298N. The Arduino At mega 2560 was the central head of the mechanical device. Figure 6.1 shows the chassis of the car. The ultrasonic sensor was connected to Pin A1, Pin A2 and GND of ATmega2560 the hardware developed consists of Microcontroller Arduino ATMEGA 2560, ultrasonic sensor, two DC motors and motor driver L298N".



"The Arduino ATMEGA 2560 was the central head of the robot. Figure 6.1 shows the frame of the robot. The ultrasonic sensor was connected to Pin A1, Pin A2 and GND of ATMEGA2560. If any object was moving in front of the robot, the ultrasonic sensor output will alert the microcontroller that an obstacle was detected. Motion sensor has 90 degrees field of view and it will be triggered when a warm object interchanges through the area it is facing. It is very sensitive and will trigger with just a hand movement. The Ultrasonic sensor was shown in Figure 6.1".

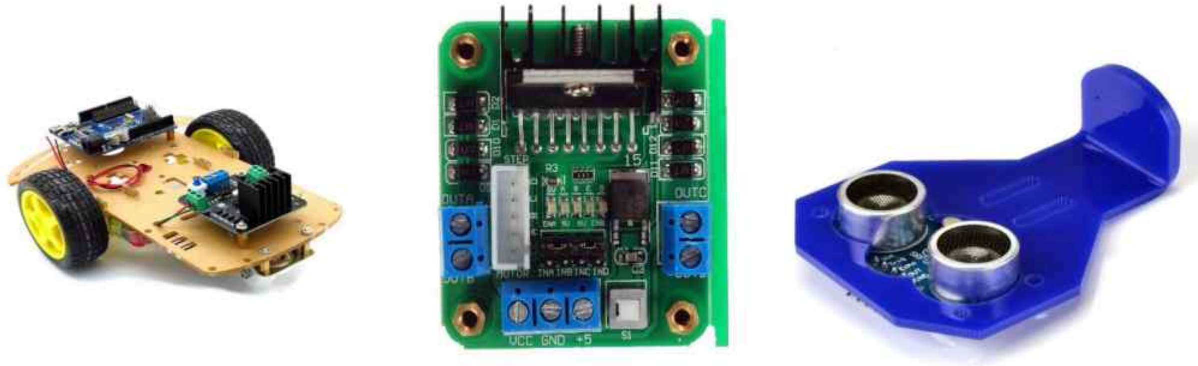


Fig 6.1: Setup of the system

6.3 Testing of the Project

Testing is a process which is ensures the output of project right or wrong. So it was very efficient and very helpful for implementation. Testing are generally two types. There are functional testing and structural testing. I just tested my project very easily. I tested my project through Arduino IDE. I try to show the outcome of my project by some still picture.

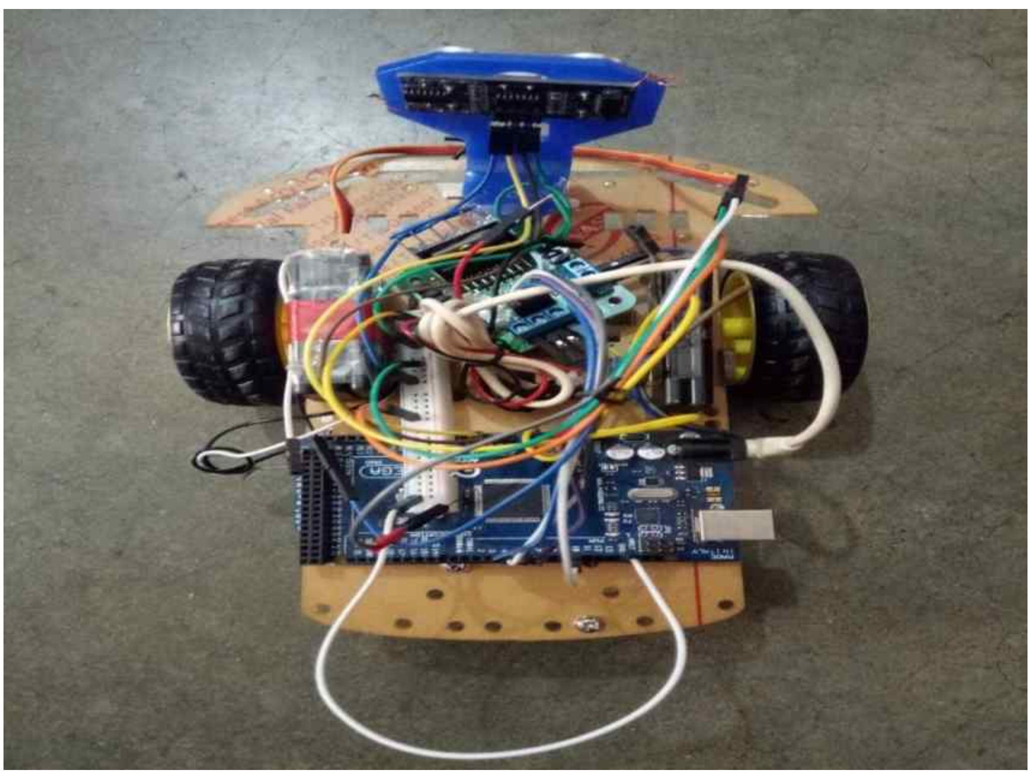


Fig 6.2: Display output of the system

The following figure 6.3 is shows Upload Embedded Code through Arduino IDE.

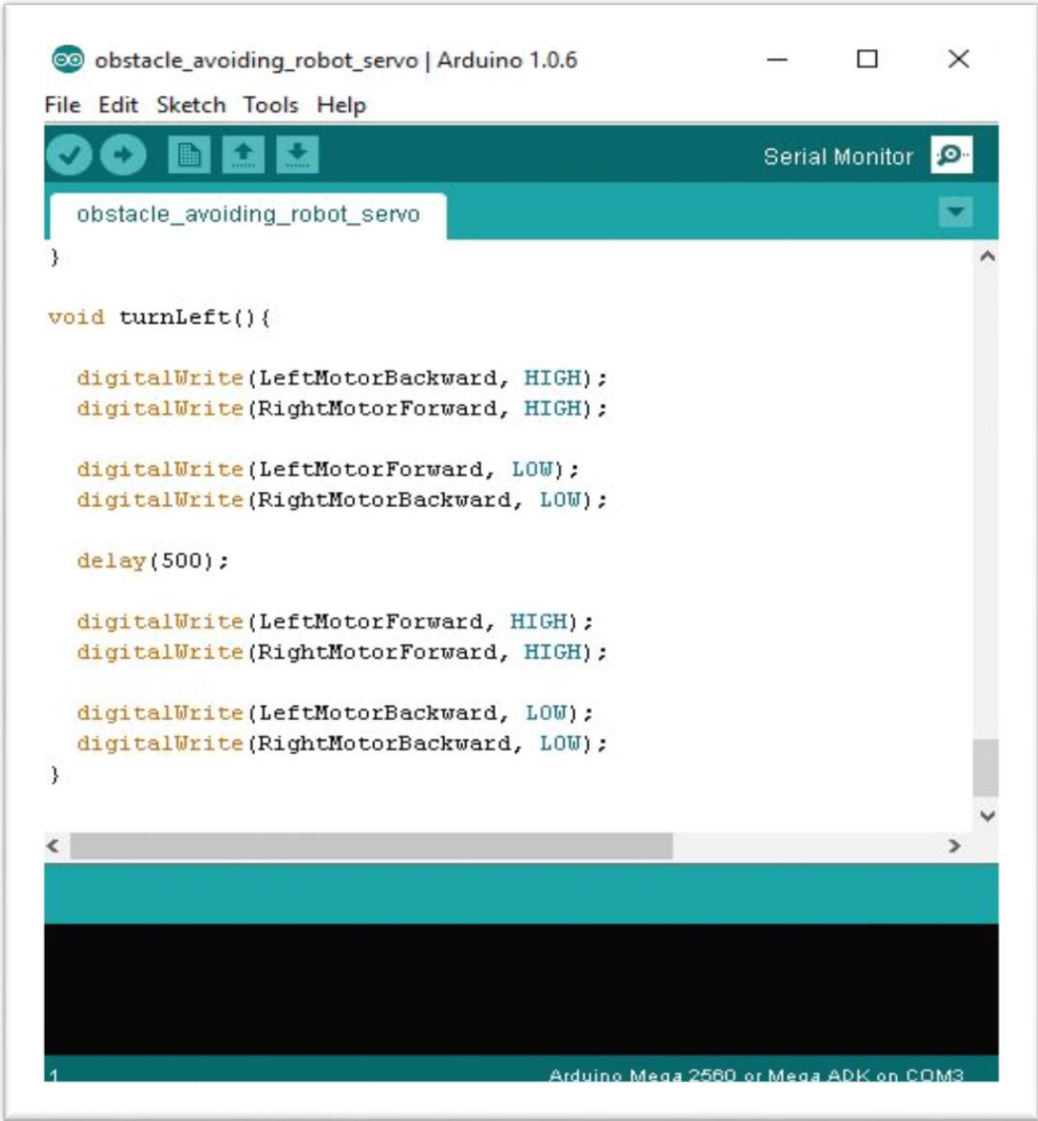


Figure 6.3: Upload Embedded Code through Arduino IDE

6.4 Result

"A robot that can productively detect and avoid obstacles was considered and created". It was found that given a number of obstacles, the robot is able to detect and avoid the obstacle with an average accuracy of 84%. "The equation for calculating the accuracy is given below". "The numerator signifies the total number of times the robot was able to avoid the obstacle it faced". The denominator signifies the total number of test cases.

$$\text{Accuracy} = \frac{\boldsymbol{\Sigma}\ \textbf{Number of Successful Avoidances}}{\boldsymbol{\Sigma}\ \textbf{number of Test Cases}}$$

Type of obstacle	Detected	Avoided	Accuracy
Single solid obstacle	Yes	Yes	100%
Uniform shaped surface	Yes	Yes	100%
Non-Uniform shaped surface	Yes	No	65%
Double solid obstacles	Yes	Yes	71%
			Average = 84%

6.5 Advantages

- Anyone can use this.
- Easy to control.
- Easy to setup.
- Easy to maintain and repair.
- Low power consumption.
- It is easy to change to the systems action according to change in situation.
- The programing of the microcontroller is easy.

6.6 Disadvantages

- This is applicable for flat area.
- It can be traceable by pushing the reset button.
- It is not in human control.

6.7 Conclusion

"Exactly all navigation robot demands the some sort of obstacle detection, hence obstacle avoidance policy is of most importance". Obstacle Avoidance Robot has a vast field of application. "They can be used as facilities robots, for the purpose of everyday work and so many other indoor applications". "Similarly they have great significance in technical exploration and emergency rescue, there may be places that are dangerous for humans or even impossible for humans to reach directly, then we should use robots to help us. In individuals challenging environments, the robots need to gather data about their environments to avoid obstacles". "At the present time, even in normal environments, people involve that robots to detect and avoid obstacles". "For example, an industrial robot in a factory is estimated to avoid workers so that it won't hurt them. In conclusion, obstacle avoidance was widely researched and applied in the world, and it was possible that most robots in the future should have obstacle avoidance function".

CHAPTER 7

CONCLUSION

7.1 Conclusion

"A lot of elements determined the accuracy of the robot I designed. These factors were the Environmental phenomenon in which the robot was verified such a well-lit or a dimly-lit Environment, the number of obstacles current making the test space packed or comparatively less packed the type and shape of the obstacle (the robot is designed for a uniform shaped obstacle) ".

These elements majorly affected the sensors. The accuracy of the robot is dependent on the Sensors used. Thus the nature of the sensor and its accuracy defined the accuracy of my Robot.

7.2 Applications

- Mainly military applications.
- It can be used for city wars.

7.3 Limitations of the Work

There are some limitations of my system. Such as,

- It was a low range circuit, i.e. it was not possible to operate the circuit remotely.
- Further improvement can be achieved by adding sensors on the left and right side of the robot.
- Besides that, computer vision with camera features can be implemented for monitoring applications. For further improvement, to implement an obstacle avoidance in aerospace, well-suited sensors should be used to gather the accurate information about the environment and obstacles.

7.4 Future work

Due to time and methodological constraints, "the development could not be properly implemented on an Arduino board with the proper hardware".

- The project can be further enhanced by GPS
- I can add fire, wind and LPG sensors so that, will automatically work it.
- Sensor will be used to include different applications.

REFERENCES

[1] General Block Diagram. (07/12/17). Retrieved from http://www.apastyle.org/learn/faqs/web-page-no-author.aspx

[2] Description of Microcontroller: ATmega2560. (2017/10/20). Retrieved from https://store.arduino.cc/usa/arduino-mega-2560-rev3

[3] Specification Arduino Mega2560. (2017/10/20). Retrieved from https://store.arduino.cc/usa/arduino-mega-2560-rev3

[4] Block Diagram of Microcontroller – (ATmega2560). (20/2/2016). Retrieved from https://www.researchgate.net/figure/The-block-diagram-of-the-ATMEGA-2560-microcontroller_fig6_292694418

[5] Pin Configurations of Microcontroller – (ATmega2560). (2017/10/20). Retrieved from https://www.arduino.cc/en/Hacking/PinMapping2560

[6] Arduino Mega 2560 PIN mapping table. (2017/10/20). Retrieved from https://www.arduino.cc/en/Hacking/PinMapping2560

[7] Pin Descriptions. (2017/10/20). Retrieved from https://store.arduino.cc/usa/arduino-mega-2560-rev3

[8] L298N H-bridge. (Aug. 28, 2014). Retrieved from http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/

[9] How it Works. Dejan Nedelkovski. (August 9, 2017). Retrieved from https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

[10] Description of HC SR04 Sensor. (August 27, 2017). Retrieved from https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/

[11] SG90 Servo. (January 10, 2016). Retrieved from https://components101.com/servo-motor-basics-pinout-datasheet

Appendix 1: Arduino Code

```
#include <Servo.h>          //Servo motor library. This is standard library
#include <NewPing.h>        //Ultrasonic sensor function library. You
must install this library

//our L298N control pins
const int LeftMotorForward = 6;
const int LeftMotorBackward = 7;
const int RightMotorForward = 5;
const int RightMotorBackward = 4;

//sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2

#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

NewPing sonar (trig_pin, echo_pin, maximum_distance); //sensor
function
Servo servo_motor; //our servo name


void setup(){

  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
```

```
  pinMode(RightMotorBackward, OUTPUT);

  servo_motor.attach(10); //our servo pin

  servo_motor.write(115);
  delay(2000);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
}


void loop(){

  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  if (distance <= 20){
    moveStop();
    delay(300);
    moveBackward();
    delay(400);
    moveStop();
    delay(300);
```

```
    distanceRight = lookRight();
    delay(300);
    distanceLeft = lookLeft();
    delay(300);

    if (distance >= distanceLeft){
      turnRight();
      moveStop();
    }

    else{
      turnLeft();
      moveStop();
    }
  }
  else{
    moveForward();
  }
    distance = readPing();
}

int lookRight(){
  servo_motor.write(50);
  delay(500);
  int distance = readPing();
  delay(100);
  servo_motor.write(115);
  return distance;
}
```

```
int lookLeft(){
  servo_motor.write(170);
  delay(500);
  int distance = readPing();
  delay(100);
  servo_motor.write(115);
  return distance;
  delay(100);
}

int readPing(){
  delay(70);
  int cm = sonar.ping_cm();
  if (cm==0){
    cm=250;
  }
  return cm;
}

void moveStop(){

  digitalWrite(RightMotorForward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorBackward, LOW);
}

void moveForward(){

  if(!goesForward){
```

```
    goesForward=true;

    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);

    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
  }
}

void moveBackward(){

  goesForward=false;

  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorForward, LOW);

}

void turnRight(){

  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorForward, LOW);
```

```
  delay(500);

  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);



}

void turnLeft(){

  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);

  delay(500);

  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);

  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}
```