Sudan University of Science and Technology

College of Engineering

School of Electrical and Nuclear Engineering

Obstacle Avoider Robotic Vehicle

(المركبة الآلية متجنبة العوائق)

Aproject Submitted In Partial Fulfillment for the Requirements of the Degree of b.sc.(Honor) Electrical Engineering

Prepared by:

1. **Abubakr Taha Mohammed Taha**
2. **Abdallah Siddig Ahmed Mohammed**
3. **Abdalla Mohyeldeen Mohammed Ahmed**
4. **Mohammed Salih Mohammed Ali**

Supervised By:

Ust. Mohanad Hamad Eljack Elameen

October 2016

الآيـــة

قال الله تعالى :

﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ ﴾

(سورة المجادلة : الآية ١١)

DEDICATION

To the Song of tenderness and the source of the effort of stayed on my comfort, to the one who inhabit paradise under their feet ...

"' my beloved mother"'

To the shield solid who taught me the meaning of life and grew in me love of science and work and self-reliance

"'dear dad"'

To the one who I am very proud of their presence with me, whom shine Rose in the garden of my life, my happiness was great

"' my friends"'

To each of the extended a helping hand to me and helped me in my research output, I dedicate this research to all of them

ACKNOWLEDGMEN

Praise and thanks first and last resort to God Almighty and offer thanks to the Sudan University of Science and Technology and to each of the lit candle in the paths of our knowledge and to stop on the platforms and give the outcome idea to enlighten our path and we thank Lecturer. Mohanad Hamad Eljack Elameen at the Faculty of engineering , Sudan University Science and Technology, which was approved and honored to oversee this research, his directions was as a lamp enlightening us through knowledge, which he gave his time and effort in order to bring this search to this picture, we do not find words of thanks and appreciation to give him his right. and also thank all those who contributed and helped us on the output of this research.

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. The project is designed to build an obstacle avoidance robotic vehicle using ultrasonic sensors for its movement. An Arduino uno is used to achieve the desired operation. A robot is a machine that can perform task automatically or with guidance. Robotics is generally a combination of computational intelligence and physical machines (motors). Computational intelligence involves the programmed instructions. This robotic vehicle is built, using an Arduino uno. An ultrasonic sensor is used to detect any obstacle ahead of it and sends a command to the Arduino.

مستخلص البحث

كثيراً من الصناعات اليوم تستخدم الربوتات نظراً للأداء العالي الذي يساعد الإنسان والموثوقية العالية . ربوت تجنب العوائق يستخدم للتحقق من وجود العقبة وتجنب الاصطدام بها وهو ربوت ذاتي التحكم. لتصميم الروبوت يجب خلق توافق تام بين مكونات الروبوت .كشف العقبة هي المهمة الأساسية لروبوت المتحكم الذاتي .يحصل الروبوت على المعلومات من البيئة المحيطة من خلال أجهزة الاستشعار الموضوعة عليه . ومن أجهزة الاستشعار المستخدمة لكشف العقبة أجهزة الاستشعار بالأشعة تحت الحمراء وأجهزة الاستشعار بالموجات فوق الصوتية.والمتحسسات فوق الصوتية هي الأكثر مناسبة لرخص ثمنها ومداها العالي مقارنة مع نظائرها .يهدف المشروع إلى بناء مركبة ربوتية تتجنب العقبات أثناء حركتها باستخدام أجهزة الإستشعار بالموجات فوق الصوتية . يتم إستخدام اردوينو اونو لتحقيق العملية المطلوبة . والربوت يمكن أن يؤدي المهمة المطلوبة تلقائياً أو مع التوجيه .الروبوت عموماً هو مزيج من الذكاء الحسابي والمكونات المادية ويشمل الذكاء الحسابي التعليمات المبرمجة .بنيت هذه السيارة باستخدام اردوينو وانو وأجهزة إستشعار بالموجات فوق الصوتية للكشف عن أي عقبة أمامه ويرسل الأوامر إلى الاردوينو.

TABLE OF CONTENT

LIST OF FIGURE

CHAPTER ONE

INTRODUCTION

1.1 General concepts

Many of us are wondering how a robot functions, what types of technologie Are used in a robot and why we need a robot in our life. The aim is To Provide the reader with a clear, simple explanation of robotics. The information is directed to wards engineering students , and engineers who are interested ina robotics.In the beginning ,you will find a general idea and the development of robot technologies, some applications of an indu--strial robot and a nonindustrial robot.How robotics has developed in the last few decades and how it begins to play a vital role in our industrial life[1].

1.2 Origin of the word robot

The word "Robot" comes from the Czech word "Robota" which means "labor doing compulsory manual works without receiving any remuneration " or "to make things manually". Oxford dictionary defines "Robot"as a machine resembling a human being and able to replicate certain human movements and functions automatically." These days, more and more robot is living up to its name. Gone is the time when robots are merely. mechanisms attached to controls. Today's robots are a combination of manipulative, perceptive, communicative, and cognitive abilities. They can be seen welding heavy machinery or repairing nuclear power plants[2].

1.3 definition:

A robot is a re programmable, multi functional manipulate or designed to move material, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks[3].

1.4 What is and what is not a robot:

There are some essential characteristics that a robot must have, and this might help to decide what is and what is not a robot. It will also help to decide what features we will need to build into a machine before it can count as a robot. A robot has these essential characteristics[4]:

1.4.1 Sensing

First of all robot would have to be able to sense it surroundings. It would do this in ways that are not similar to the way that human sense his surroundings .Giving robot sensor: light sensors (eyes), touch and pressure sensors (hands), chemical sensors (nose), hearing and sonar sensors (ears), and taste sensors (tongue) will give robot awareness of its environment.

1.4.2 Movement

A robot needs to be able to move around its environment whether rolling on wheels, walking on legs or propelling by thrusters a robot needs to be able to move. To count as a robot either the whole robot moves, like the sojourner or just parts of the robot moves, like the Canada arm.

1.4.3 Energy

A robot needs to be able to power itself. A robot might be solar powered, electrically powered or battery powered. The way that robot gets its energy will depend on what is a robot needs to do.

1.4.4 Intelligence

A robot needs some kind of smarts. This is where programming enters the pictures. The robot will have to have some way to receive the program so that it knows what it is to do .

1.5 Robotics and Robots

It is a branch of technology and deals with designing construction Operation Application of robot. It also deals with the computer systems for their sensory control,information processing and feedbackThese Technologies deal with automated machines that can replace humans in manufacturing processes or dangerous environments. These robots resemble humans in behavior, and/or cognition . Robotics requires a working knowledge of mechanics, electronics and software [5].

1.6 Robot applications in our lives

1.6.1 Welding

Considered as a dangerous task for a human because of toxic gases emissions. The welding job is quite difficult for a person who is required to weld two pipes from different sides and angles and to sit in a difficult position for a long time. Itcan be hard on ones physic and can cause health problems for the worker.The difficulty for a human is to see all the sides of welded deviceswhen he needs to needs weld around a pipe as he can only see one side of the pipe.

Figure 1.1: welding robot

1.6.2 Painting:

has similar problems to welding due to the use of toxic chemical products. Below is an example picture1.2 of a factory robot painting a car as it moves slowly along a conveyer.

Figure 1.2 :painting robot

1.6.3 Assembly operation

When we assemble a chip we need to be very precise because of very fine wires which require very precise and accurate tasks which ahuman cannot handle but, on the other hand, is easy for a robot.

1.6.4 Medical robot to make surgery

Patient gets fast recovery. The operation is more precise with fewer mistakes.Robot can open small incisions in the body and carry out major operations with minimal damage to the patient .Therefore recovery time is decreased.The equipment is more hygienic and safety.

1.6.4 Mobile robot with legs or wheel

for chemical power plant, under see or remote are as and bombs fields. The advantage in leg robot is that it can avoid step over obstacles Which can be dangerous like bomb or even to protect objects from being destroyed due to robot moving over them.

1.6.5 Robotics aircrafts and boats

without pilot which are guided from a station on The ground, which are used by army or rescue mission.

1.6.6 Robotic toys for entertainment

1.6.7 Robot for cleaning at home and industry

Figure1.3: medical robot

Figure1.4: Mobile robot with legs or wheel

Figure1.5: Robotics aircrafts

CHAPTER TWO

PREVIOUS STUDIES

IN 1970

Freddy and Freddy II, both built in the United Kingdom, were robots capable of assembling wooden blocks in a period of several hours. German based company KUKA built the world's first industrial robot with six electromechanically driven axes, known as FAMULUS. In 1974, David Silver designed The Silver Arm; the Silver Arm was capable of fine movements replicating human hands. Feedback was provided by touch and pressure sensors and analyzed by a computer. The SCARA, Selective Compliance Assembly Robot Arm, was created in 1978 as an efficient, 4-axis robotic arm. Best used for picking up parts and placing them in another location, the SCARA was introduced to assembly lines in 1981. The Stanford Cart successfully crossed a room full of chairs in 1979. The Stanford Cart relied primarily on stereo vision to navigate and determine distances. The Robotics Institute at Carnegie Mellon University was founded in 1979 by Raj Reddy.[6]

IN 1980

KUKA IR 160/60 Robots from 1983 Takeo Kanade created the first "direct drive arm" in 1981. The first of its kind, the arm's motors were contained within the robot itself, eliminating long transmissions. In 1984 Wabot-2 was revealed; capable of playing the organ, Wabot-2 had 10 fingers and two feet. Wabot-2 was able to read a score of music and accompany a person. In 1986, Honda began its humanoid research and development program to create robots capable of interacting successfully with humans. A hexapodal robot named Genghis was revealed by MIT in 1989. Genghis was famous for being made quickly and cheaply due to construction methods; Genghis used 4 microprocessors, 22 sensors, and 12 servo motors. Rodney Brooks and Anita M. Flynn published "Fast, Cheap, and Out of Control: A Robot Invasion of The Solar System". The paper advocated creating smaller cheaper robots in greater numbers to increase production time and decrease the difficulty of launching robots into space.[6]

IN 1990

The biomimetic robot RoboTuna was built by doctoral student David Barrett at the Massachusetts Institute of Technology in 1996 to study how fish swim in water. RoboTuna is designed to swim and resemble a blue fin tuna. Invented by Dr. John Adler, in 1994, the Cyberknife (a stereotactic radio surgery performing robot) offered an alternative treatment of tumors with a comparable accuracy to surgery performed by human doctors. IBM's Deep Blue computer, defeated World Chess Champion Garry Kasparov in 1997. Honda's P2 humanoid robot was first shown in 1996. Standing for "Prototype Model 2", P2 was an integral part of Honda's humanoid development project; over 6 feet tall, P2 was smaller than its predecessors and appeared to be more human-like in its motions. Expected to only operate for seven days, the Sojourner rover finally shuts down after 83 days of operation in 1997. This small robot (only weighing 23 lbs) performed semi-autonomous operations on the surface of Mars as part of the Mars Pathfinder mission; equipped with an obstacle avoidance program, Sojourner was capable of planning and navigating routes to study the surface of the planet. Sojourner's ability to navigate with little data about its environment and nearby surroundings allowed the robot to react to unplanned events and objects. The P3 humanoid robot was revealed by Honda in 1998 as a part of the company's continuing humanoid project. In 1999, Sony introduced the AIBO, a robotic dog capable of interacting with humans, the first models released in Japan sold out in 20 minutes. Honda revealed the most advanced result of their humanoid project in 2000, named ASIMO. ASIMO is capable of running, walking, communication with humans, facial and environmental recognition, voice and posture recognition, and interacting with its environment. Sony also revealed its Sony Dream Robots, small humanoid robots in development for entertainment. In October 2000, the United Nations estimated that there were 742,500 industrial robots in the world, with more than half of the robots being used in Japan.[6]

IN 2001

Roomba vacuum cleaner docked in base station.In April 2001, the Canadarm2 was launched an orbit and attached to the International Space Station. The Canadarm2 is a larger, more capable version of the arm used by the Space Shuttle and is hailed as being "smarter." Also in April, the Unmanned Aerial Vehicle Global Hawk made the first autonomous non-stop flight over the Pacific Ocean from Edwards Air Force Base in California to RAAF Base Edinburgh in Southern Australia. The flight was made in 22 hours. The popular Roomba, a robotic vacuum cleaner, was first released in 2002 by the company iRobot. In 2004, Cornell University revealed a robot capable of self-replication; a set of cubes capable of attaching and detaching, the first robot capable of building copies of itself. On 3 and 24 January the Mars rovers Spirit and Opportunity land on the surface of Mars. Launched in 2003, the two robots will drive many times the distance originally expected, and Opportunity is still operating as of mid 2012. Self-driving cars had made their appearance by the middle of the first decade of the 21st century, but there was room for improvement. All 15 teams competing in the 2004 DARPA Grand Challenge failed to complete the course, with no robot successfully navigating more than five percent of the 150 mile off road course, leaving the $1 million prize unclaimed. In 2005, Honda revealed a new version of its ASIMO robot, updated with new behaviors and capabilities.]In 2006, Cornell University revealed its "Starfish" robot, a 4-legged robot capable of self modeling and learning to walk after having been damaged. In 2007, TOMY launched the entertainment robot, i-sobot, which is a humanoid bipedal robot that can walk like a human beings and performs kicks and punches and also some entertaining tricks and special actions under "Special Action Mode".Robonaut 2, the latest generation of the astronaut helpers, launched to the space station aboard Space Shuttle Discovery on the STS-133 mission. It is the first humanoid robot in space, and although its primary job for now is teaching engineers how dexterous robots behave in space, the hope is that through upgrades and advancements, it could one day venture outside the station to help spacewalkers make repairs or additions to the station or perform scientific work. Commercial and industrial robots are now in widespread use performing jobs more cheaply or with greater accuracy and reliability than humans. They are also employed for jobs which are too dirty, dangerous or dull to be suitable for humans. Robots are widely used in manufacturing, assembly and packing, transport, earth and space exploration, surgery, weaponry, laboratory research, and mass production of consumer and industrial goods. With recent advances in computer hardware and data management software, artificial representations of humans are also becoming widely spread. Examples include Open MRS and EMR Bots.[6]

CHAPTER THREE

COMPONENT OF ROBOT

3.1 Hardware components

3.1.1 Manipulator:

Just like the human arm, the robot consists of what is called a manipulator having several joints and links.

3.1.2 Endeffector

The base of the manipulator is fixed to base support and at its other free end, the End effectors is attached. The End effector is expected to perform tasks normally performed by the palm and finger arrangements of the human arm.

3.1.3 The Locomotion Device

In the case of Human Beings the power for the movement of the arm, the palm and fingers is provided by muscles. For the robot the power for the movement (locomotion) is provided by the motors. The motors used for providing locomotion in robots are of three types depending on the source of energy: Electric, Hydraulic or Pneumatic.

3.1.4 The Controller

The digital computer (both the hardware and the software) acts as a controller to the robot. The controller functions in a manner analogous to the human brain. With the help of this controller, the robot is able to carry out the assigned tasks. The controller directs and controls the movement of the Manipulator and the End effector . In other words, the controller controls the robot.

3.1.5 The Sensors

Without the data supplied by the sense organs, the brain would be incapable of intelligence. In other words the controller (the computer) of the robot cannot do any meaningful task, if the robot is not with a component analogous to the sense organs of the human body. Thus, the fifth and the most important component of the robot is the set of sensors. Sensors are nothing but measuring instruments which measures quantities such as position, velocity, force, torque, proximity, temperature, etc.

3.2 software

The software include the program that robot is working with .that program must take The low of robot as target and that low call Asimov’s Laws (The Three lows) .

3.2.1 Laws of Robotic

Asimov’s Laws (The Three lows) of Robotics constrain robots to server their human masters. These laws of software encompass a host of desiderata and tradeoffs that software developers need to keep in mind, and demonstrate that issues that are typically treated in a fragmented manner are actually strongly intertwined .In 1940, science fiction writer Isaac Asimov formulated the following three Laws of Robotics and later added zero law :

0. Robot may not injure humanity or through inaction,allow humanity to come to harm .

1. *A robot may not injure a human being, or, through inaction, allow a human* being to come to harm, unless this would violate a higher order law .

2. A robot must obey orders given it by human beings, except where such orders would contradict with the First Law .

3. A robot must protect its own existence as long as such protection does not contradict with the First or Second Law**.**

(Asimov's Laws of Robotics Applied to Software Dror G. Feitelson School of Computer Science and Engineering The Hebrew University of Jerusalem91904 Jerusalem, Israel)

3.3 Obstacle avoider robot

3.3.1 Hardware components:

- Arduino UNO .
- HC-SR04 Ultra sonic sensor
- Robot chassis + 2 /4DC Motors with holder + 2 /4 Wheels + 1 Castor Wheel + Screws & Nuts
- L293D motor driver
- Basic electronics kit contains breadboard, connecting wires, battery & other small useful items
- Jumper Wires: Male to Male, Male to Female
- Power source: 5V power bank or 5V battery pack .
- Software: Arduino IDE[7]

3.3.2 software

As we say the soft ware consist the program that robot is working with. there are many program can be used to program robot and here we use arduino ide .

- **Arduino IDE**

The Arduino IDE (Integrated Development Environment) is the program used to write code, and comes in the form of a downloadable file on the Arduino website.

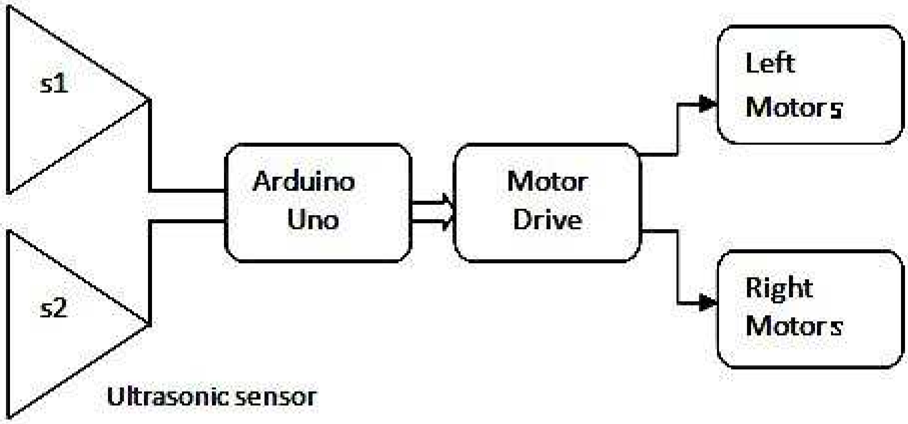

Figure 3.1: obstacle avoider block diagram

3.4 Popular Programming Languages in Robotics

There are over 1500 programming languages in the world, Here are the ten most popular programming languages in robotics at the moment. Each language has different advantages for robotics.

3.4.1 C/C++

Many people agree that C and C++ are a good starting point for new robot cists. Why? Because a lot of hardware libraries use these languages. They allow interaction with low level hardware, allow for real time performance and are very mature programming languages. These days, you'll probably use C++ more than C, because the language has much more functionality. C++ is basically an extension of C. It can be useful to learn at least a little bit of C first, so that you can recognize it when you find a hardware library written in C. C/C++ are not as simple to use as, say, Python or MATLAB. It can take quite a lot longer to implement the same functionality using C and it will require many more lines of code. However, as robotics is very dependent on real time performance, C and C++ are probably the closest thing that we robot cists have to "a standard language".*[8]*

*3.4.*2 Python

There has been a huge resurgence of Python in recent years especially in robotics. One of the reasons for this is probably that Python (and C++) are the two main programming languages found in ROS. Like Java, it is an interpretive language. Unlike Java, the prime focus of the language is ease of use. Many people agree that it achieves this very well. Python dispenses with a lot of the usual things which take up time in programming, such as defining and casting variable types. Also, there are a huge number of free libraries for it, which means you don't have to "reinvent the wheel" when you need to implement some basic functionality. And since it allows simple bindings with C/C++ code, this means that performance heavy parts of the code can be implemented in these languages to avoid performance loss. As more electronics start to support Python "out-of-the-box" (as with Raspberry Pi), we are likely to see a lot more Python in robotics. *[8]*

*3.4.*3 Java

As an electronics engineer, I am always surprised that some computer science degrees teach Java to students as their first programming language. Java "hides" the underlying memory functionality from the programmer, which makes it easier to program than, say, C, but also this means that you have less of an understanding of what it's actually doing with your code. If you come to robotics from a computer science background (and many people do, especially in research) you will probably already have learned Java. Like C# and MATLAB, Java is an interpretive language, which means that it is not compiled into machine code. Rather, the Java Virtual Machine interprets the instructions at runtime. The theory for using Java is that you can use the same code on many different machines, thanks to the Java Virtual Machine. In practice, this doesn't always work out and can sometimes cause code to run slowly. However, Java is quite popular in some parts of robotics, so you might need it. *[8]*

3.4.4 C#/.NET

C# is a proprietary programming language provided by Microsoft. I include C#/.NET here largely because of the Microsoft Robotics Developer Studio, which uses it as its primary language. If you are going to use this system, you're probably going to have to use C#. However, learning C/C++ first might be a good option for long term development of your coding skills. *[8]*

3.4. 5 MATLAB

MATLAB, and its open source relatives, such as Octave, is very popular with some robotic engineers for analyzing data and developing control systems. There is also a very popular Robotics Toolbox for MATLAB. I know people who have developed entire robotics systems using MATLAB alone. If you want to analyze data, produce advanced graphs or implement control systems, you will probably want to learn MATLAB. *[8]*

3.4.6 Assembly

Assembly allows you to program at "the level of ones and zeros”. This is programming at the lowest level (more or less). In the recent past, most low level electronics required programming in Assembly. With the rise of Arduino and other such microcontrollers, you can now program easily at this level using C/C++, which means that Assembly is probably going to become less necessary for most robot cists. *[8]*

3.4.7 Hardware Description Languages (HDLs)

Hardware Description Languages are basically a programming way of describing electronics. These languages are quite familiar to some robot cists, because they are used to program Field Programmable Gate Arrays (FPGAs). FPGAs allow you to develop electronic hardware without having to actually produce a silicon chip, which makes them a quicker and easier option for some development. If you don't prototype electronics, you may never use HDLs. Even so, it is important to know that they exist, as they are quite different from other programming languages. For one thing, all operations are carried out in parallel, rather than sequentially as with processor based languages. *[8]*

3.4.8 LISP

LISP is the world's second oldest programming language (FORTRAN is older, but only by one year). It is not as widely used as many of the other programming languages on this list; however, it is still quite important within Artificial Intelligence programming. Parts of ROS are written in LISP, although you don't need to know it to use ROS. *[8]*

3.4.9 Industrial Robot Languages

Almost every robot manufacturer has developed their own proprietary robot programming language, which has been one of the problems in industrial robotics. You can become familiar with several of them by learning Pascal. However, you are still going to have to learn a new language every time you start using a new robot. *[8]*

ABB has its RAPID programming language. Kuka has KRL (Kuka Robot Language). Comau uses PDL2, Yaskawa uses INFORM and Kawasaki uses

AS. Then, Fanuc robots use Karel, Stäubli robots use VAL3 and Universal Robots use URS script. In recent years, programming options like ROS Industrial have started to provide more standardized options for programmers. However, if you are a technician, you are still more likely to have to use the manufacturer's language. *[8]*

3.4.10 BASIC / Pascal

BASIC and Pascal were two of the first programming languages that I ever learned. However, that's not why I've included them here. They are the basis for several of the industrial robot languages, described below. BASIC was designed for beginners (it stands for Beginners All-Purpose Symbolic Instruction Code), which makes it a pretty simple language to start with. Pascal was designed to encourage good programming practice**s** and also introduces constructs like pointers, which makes it a good "stepping stone" from BASIC to a more involved language. These days, both languages are a bit outdated to be good for "everyday use". However, it can be useful to learn them if you're going to be doing a lot of low level coding or you want to become familiar with other industrial robot languages. *[8]*

3.5 Comparing robots to humans

Manipulation is equal to Arms and fingers driven by motors and other Form so factuation. Vision is equal to camera. Hearing is equal to micro phone. Feeling is equal to tactile sensors. Communication is equal to wires, fiber optics and radio.Brain is equal to computers and microprocessors.Smell and taste are still under development.[9]

CHAPTER FOUR

OBSTACLE AVOIDER ROBOT

In this chapter we will talk about the component of obstacle avoider robot and the steps which we have done to make it.

4.1 Mechanical Tools

There are many important tools such as:

4.1.1 Small screwdriver set

These small screwdrivers are necessary when working with electronics. Don’t force them too much though – their size makes them more fragile.

4.1.2 Regular screwdriver set

All workshops need a multi-tool or tool set which includes flat / Phillips and other screwdriver heads.

4.1.3 Needle nose pliers

A set of needle nose pliers is incredibly useful when working with small components and parts and is a very inexpensive addition to your toolbox. These are different from regular pliers because they come to a point which can get into small areas.

4.2 Component of obstacle avoider robot

4.2.1 A chassis

We chose Wheeled Robots because it has may advantage :

1) Usually low-cost compared to other methods
2) Simple design and construction
3) Abundance of choice

Figure 4.1: chassis

4.2.2 Breadboard

These boards are used to easily create prototype circuits without having to solder. This is good in the event that you have not fully developed your soldering skills or want to quickly put together prototypes and test ideas without having to solder a new circuit each time.

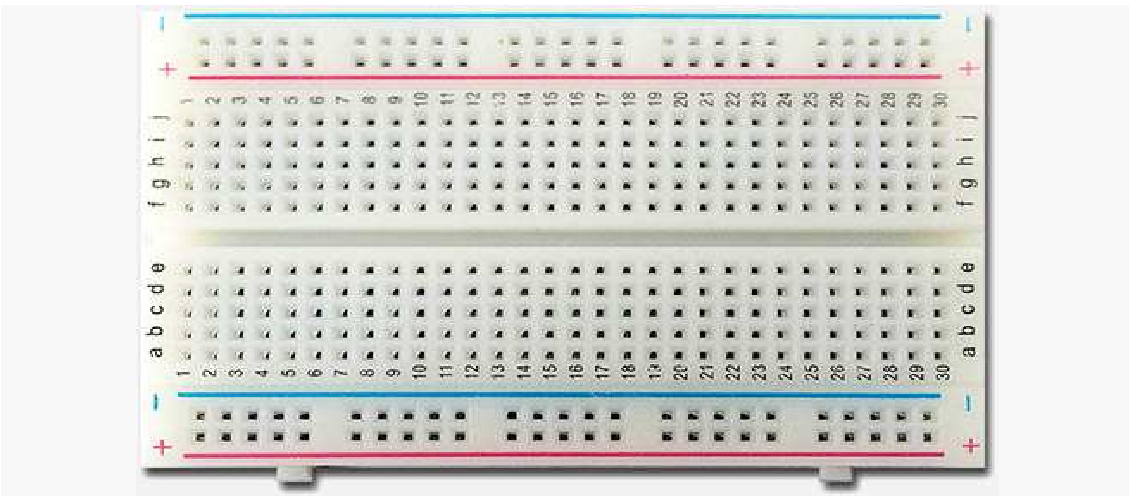

Figure 4.2: Breadboard

4. 2.3 Arduino Uno

It is the brain of the robot and the connection between all component (motor driver, power supply ,ultrasonic sensor …)

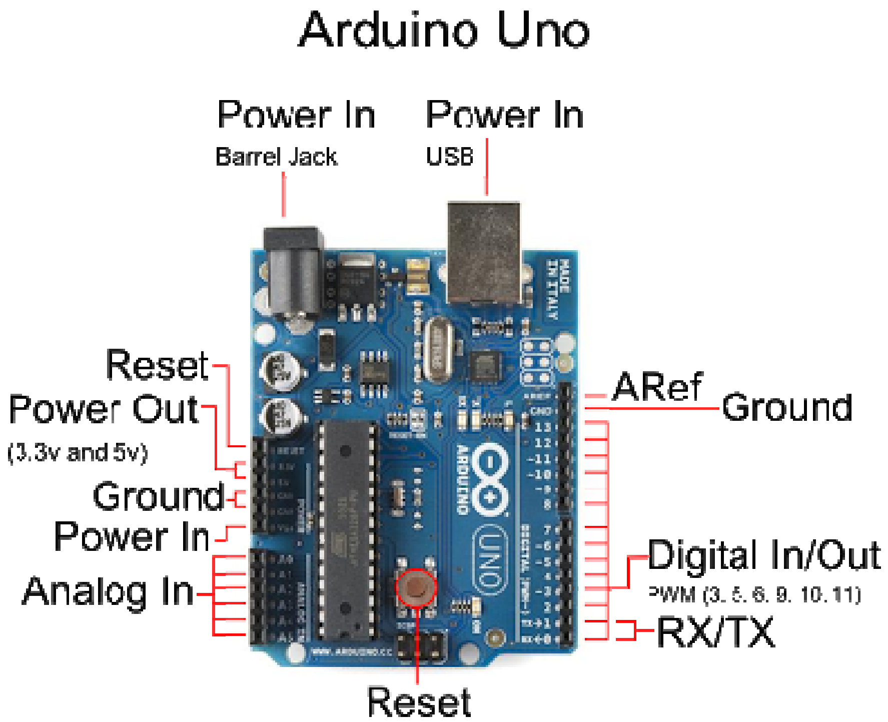

Figure 4.3: Arduino Uno part

➢ Arduino has four possible powering inputs:

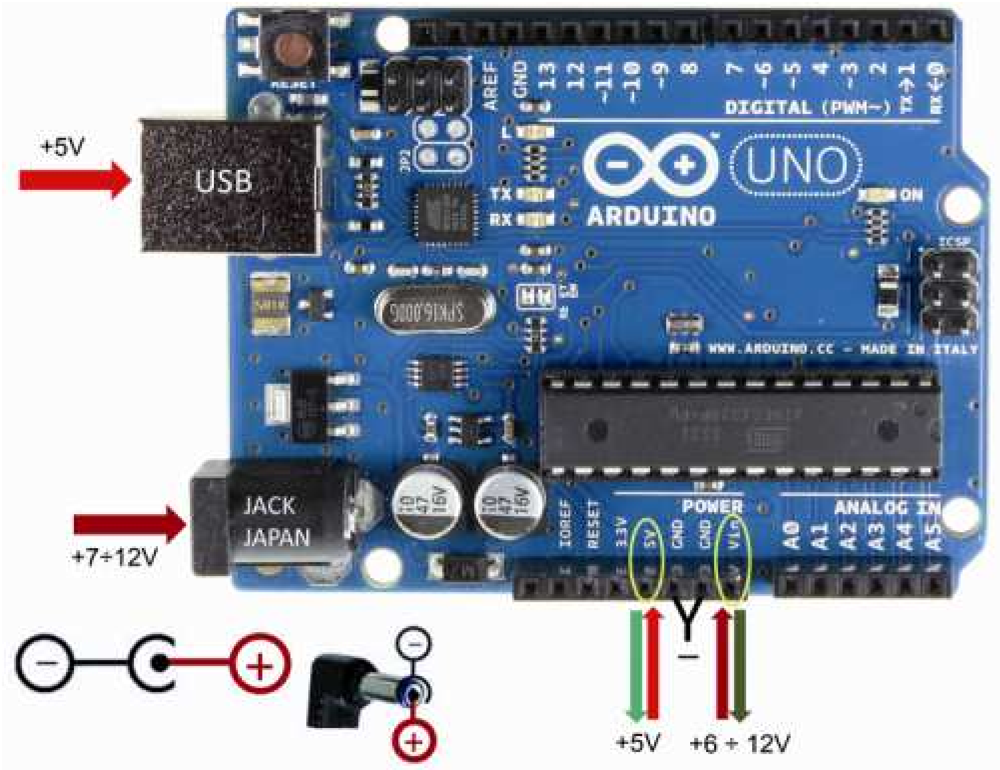

Figure 4.4 : *Arduino's powering inputs*

➢ ***Arduino's powering inputs***

1 . USB Port: 5 V have to reach this socket (different voltages are not allowed, absolutely!), coming from a computer's USB port, or from any power supply that is provided with a USB port (in general, they are small size power supplies, suitable to power devices that are provided with a USB cable). If the powering comes from a computer, there is a current limitation of 250 mA or 500 mA, depending on the USB port of the said computer; if on the other hand you are using an external power supply, the maximum output current (regardless of the one guaranteed by the same power supply, that in general is a maximum of 1 A or 2 A) is anyway limited to 500 mA by the PTC self-resettable protection fuse.

2 . JAPAN JACK socket: an external source (a power supply, usually) must be connected to this socket, with the positive pole going to the central part of the jack, and the value must be ranging between 6 V and 20 V, even though the range recommended by the manufacturer is 7÷12 V, thus it is not advisable to use voltages that are lower than 7 V or greater than 12 V, if not in the case of a real need; 6 V may not guarantee a proper stabilization on the part of the regulator, it is in fact needed to consider the voltage fall of the protection diode, placed in series at the regulator's input (whose purpose is to preserve the board from destruction in the case of polarity inversion on the jack); while values above 12 V would create an excessively high drop-out (an electric potential difference between the regulator's input and output) that would cause a pointless overheating of the regulator, even with low levels of current draw.

3 . Vin socket: this socket has a dual function.

3a – input for external powering, not protected by polarity inversions: in fact the connection goes directly to the regulator's input and below the JACK socket's diode; of course no voltage must be applied to the jack socket, otherwise dangerous conflicts might arise;3b: output from which to draw the voltage applied to the JACK socket, detracting the protection diode's fall. It might prove useful to power small loads, requiring a voltage higher than 5 V and equal to the one applied to the JACK socket (always considering the diode's voltage fall).In both cases the voltage negative pole can be found on the board's GND sockets.

4 . 5 V socket: it is directly connected to the regulator's output, thus the 5 V to power external loads to Arduino can be drawn from it. In the case voltages are not applied to the USB Port or to the JACK socket, the 5 V socket can be even used to power Arduino directly, if having an external stabilized 5 V source. One has to consider that, in general, regulators do not like voltages being applied to their output, but in this particular case this situation turns out to happen even when powering Arduino from the USB port, therefore we may assume that the designers judged this problem as harmless. Even in this case there is no form of protection, since both the diode and the PTC fuse are found above this socket and thus they do not have any active function. As in the case of the Vin socket, the voltage negative pole can be found on the board’s GND sockets.

NOTE: regardless of the input used, Arduino has a 3.3 V output socket to power loads operating at this voltage; in fact a second regulator, right for the purpose of generating 3.3 V, is directly connected to the 5 V. This socket cannot be used as input.

4.2.4 Motor driver l293 module

Arduino boards can't control dc motors by their own, because the currents they are generating are too low. Moreover, the currents they are generating cannot be reversed, so we can't change the direction of the motor. To solve that we will use a motor driver, which helps the arduino control dc motors. The most comfortable way of using motor drivers is through shields.

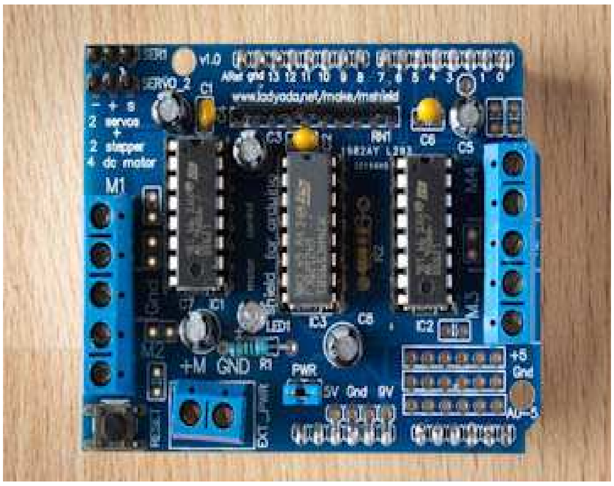

Figure 4.5: l 293 motor driver

4.2.5 Dc motor

DC motor was used in this project for the movement of the robot. The rear wheels were driven by a motor . Depending upon the input of ultrasonic sensor and MSP430, the speed of the motor varies. When the obstacle is at certain distance from the robot, it slows down at first and when it approaches the obstacle it slows down more, and after getting close to the obstacle, it stops.

Figure 4.6 : Dc motor

4.2.6 Ultrasonic sensor

Figure 4.7 : Ultrasonic sensor

Ultrasonic sensor HC-SR04 is integrated with MSP430. This sensor can be treated as one of the major components in the robot. This is used for sensing the obstacle, measuring the distance, giving input to the controller so that the controller can act accordingly and make other components integrated to it function accordingly.

- Scientific Explanation Of the sensor's action

What the sensor does it is calculating the distance from an object by sending bursts of ultrasound towards it and measuring the time it takes to the sound waves to get back (Distance = Velocity * Time). we can use this information in order to determine whether there's a close "obstacle" near the robot and then- avoid it !

4.2.7 The USB Cable

Figure 4.8 : The USB Cable

The Arduino board is connected to a computer via a USB port. The USB connection is used to upload software and also can be used as a power source when testing.

4.2.8 Power supply

Figure 4.9 :Power supply

4.3 Assembling the robot

Step 1: A chassis !

the first step and the base of any robot is a chassis. the chassis has to include a body, four motors and a battery holder and a switch.

Step 2: Brain

Arduino Uno is a compact, comfortable and relatively cheap micro-controller. In addition, it is very common and we can get it literally every-where online.

Step 3: Attaching The Arduino To The Chassis

We use some screws to attach the Arduino to the chassis.

Step 4: Attaching The Sensor To The Chassis

all we have to do next is to actually attach the sensor to the body. We recommend sticking a mini bread-board as well for easier wiring.

Step 5: Connecting The Sensor To The Brain

Step 6: Attaching The Motor Shield To The Ardunio

We take motor shield and simply plug it into the arduino with the sensor's wires crunching inside.

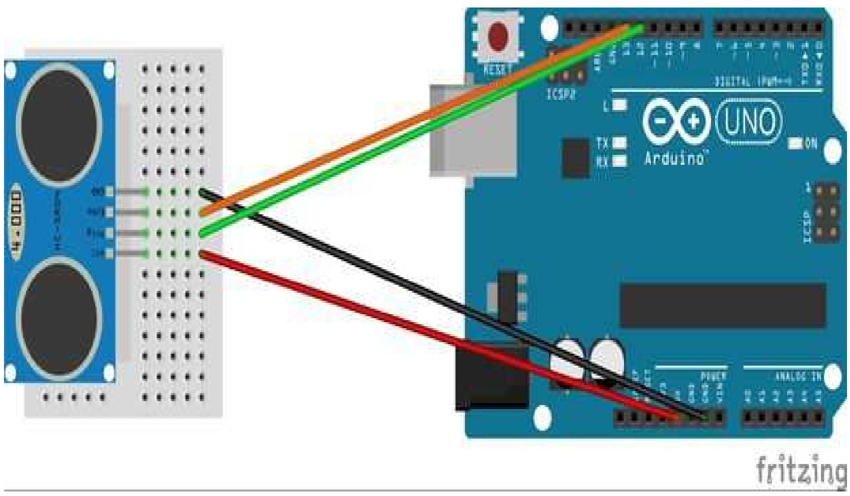

Figure 4.10 :Connecting The Sensor To The Brain

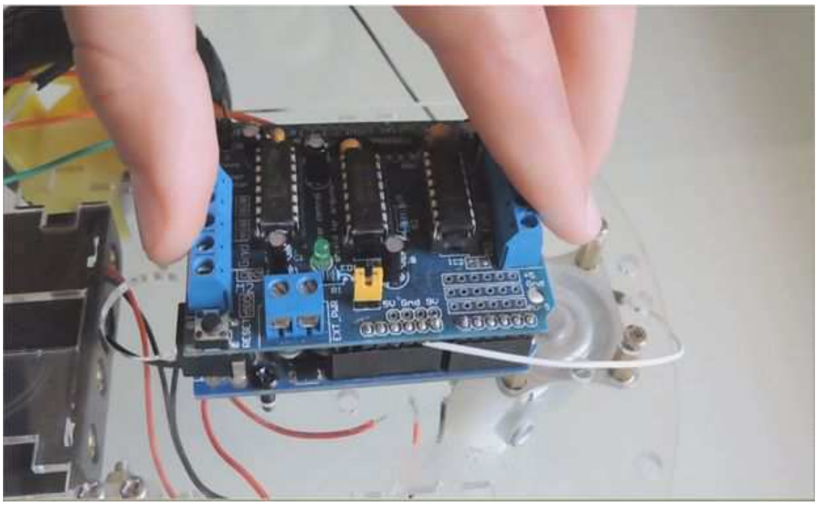

Figure 4.11: Attaching The Motor Shield To The Ardunio

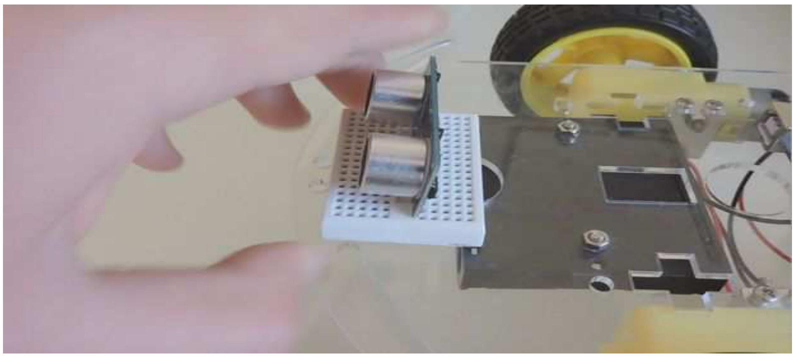

Figure 4.12 :putting The Sensor on breadboard

Step 7: Connecting The Motors To The Shield

Every Motor Shield has five channels, four for the motors, and one for a power source. Locate the channels and plug your chassis' motors to the motors' channel, and whatever power source you are using (e.g. A batteries, Lithium Battery) to the power source's channel.

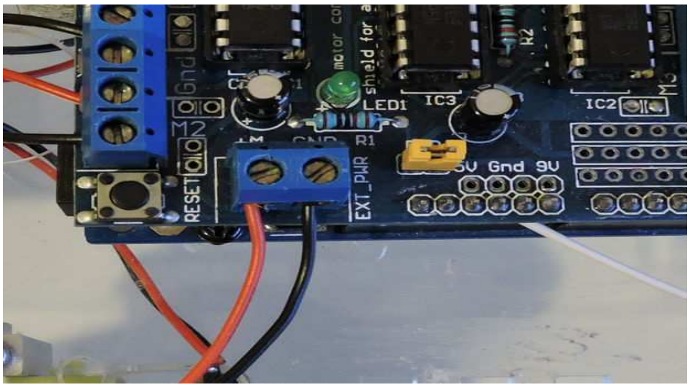

Figure 4.13 : Connecting The Motors To motor driver

Step 8: Programming The Robot

After we've made motor shield connections, From a hardware side, All we have to do next is to plug your arduino to your computer and load the code.

Step 9: getting start

After we loaded the code so the robot is ready to getting start.

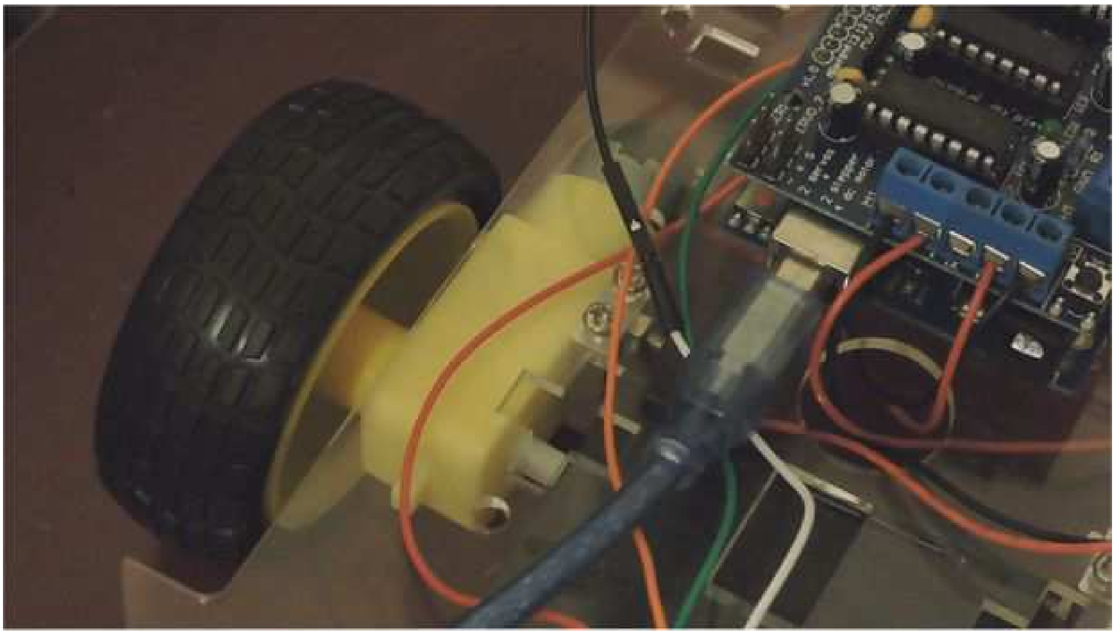

Figure 4.14: Connect arduino to computer

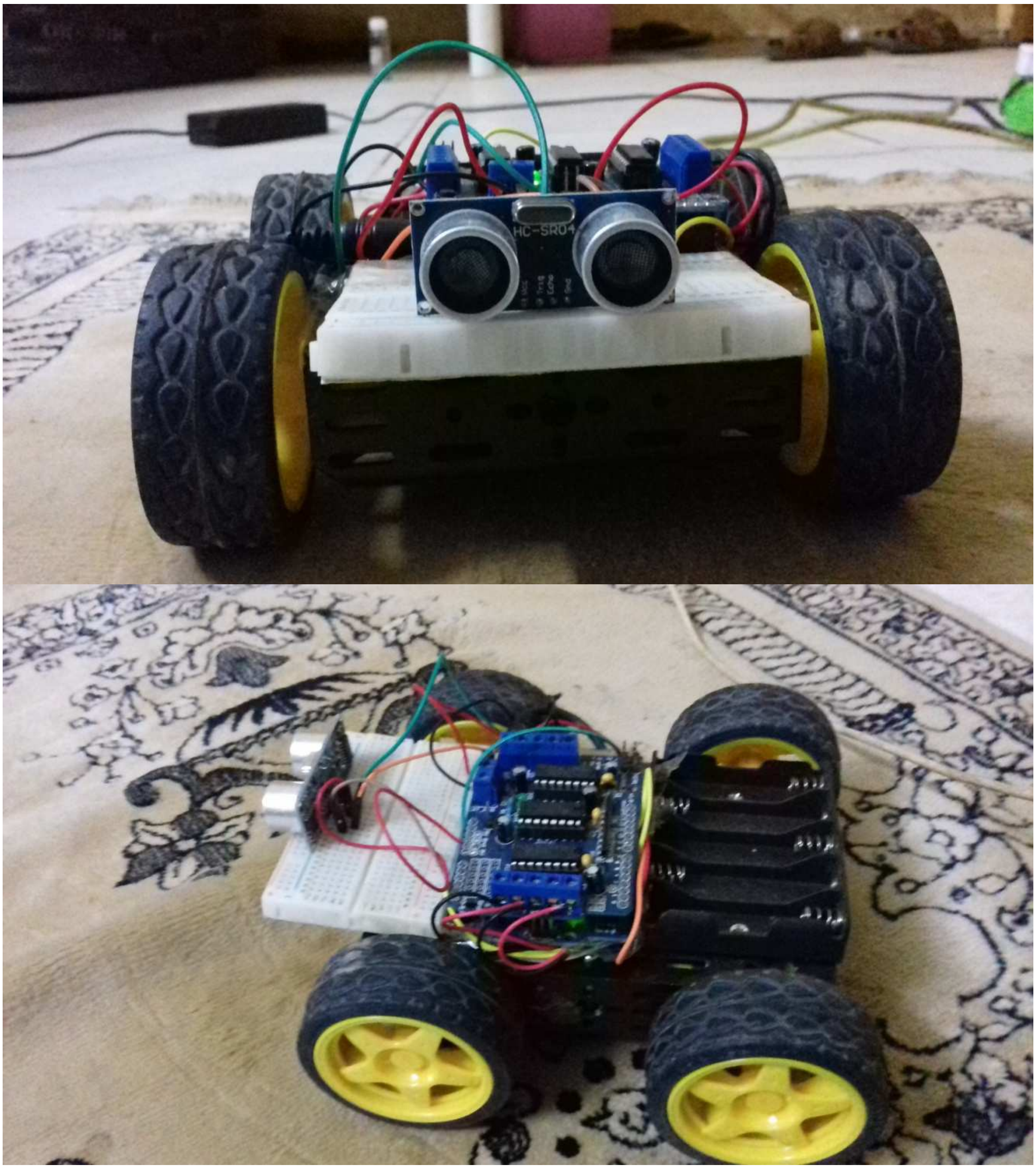

Figure 4.15 the robot

4.4 THE CODE

```
//4wd Avoiding robot Code

#include <AFMotor.h>

 #include  "Ultrasonic.h"

int distance = 0;

 AF_DCMotor r1_motor(1, MOTOR12_64KHZ); // create motor #2, 64KHz

 AF_DCMotor r2_motor(2, MOTOR12_64KHZ); // create motor #2, 64KHz

 AF_DCMotor l3_motor (3, MOTOR12_64KHZ); // create motor #2, 64KHz

 AF_DCMotor l4_motor (4, MOTOR12_64KHZ); // create motor #2, 64KHz

Ultrasonic ultrasonic (12,13);

 void  setup () {

  r1_motor.setSpeed(200);     // set the speed to 200/255

  r2_motor.setSpeed(200);     // set the speed to 200/255

  l3_motor.setSpeed(200);     // set the speed to 200/255

  l4_motor.setSpeed(200);     // set the speed to 200/255

 delay(2000);

 }

 void Forward(int SpeedR,int SpeedL)
```

```
{
 r1_motor.setSpeed(SpeedR);     // set the speed to 200/255
 r2_motor.setSpeed(SpeedR);     // set the speed to 200/255
 l3_motor.setSpeed(SpeedL);     // set the speed to 200/255
 l4_motor.setSpeed(SpeedL);     // set the speed to 200/255
  r1_motor.run(FORWARD);      // turn it on going forward
 r2_motor.run(FORWARD);      // turn it on going forward
 l3_motor.run(FORWARD);
 l4_motor.run(FORWARD);
 }
void right(int SPEED,int timer)
{
 r1_motor.setSpeed(SPEED);     // set the speed to 200/255
 r2_motor.setSpeed(SPEED);     // set the speed to 200/255
 l3_motor.setSpeed(SPEED);     // set the speed to 200/255
 l4_motor.setSpeed(SPEED);     // set the speed to 200/255
 r1_motor.run(RELEASE);      // turn it on going forward
 r2_motor.run(RELEASE);      // turn it on going forward
```

```
 l3_motor.run(FORWARD);
 l4_motor.run(FORWARD);
   delay(timer);
}
void left(int SPEED,int timer)
{
 r1_motor.setSpeed(SPEED);     // set the speed to 200/255
 r2_motor.setSpeed(SPEED);     // set the speed to 200/255
 l3_motor.setSpeed(SPEED);     // set the speed to 200/255
 l4_motor.setSpeed(SPEED);     // set the speed to 200/255
  r1_motor.run(FORWARD);      // turn it on going forward
 r2_motor.run(FORWARD);      // turn it on going forward
 l3_motor.run(RELEASE);
 l4_motor.run(RELEASE);
 delay(timer);
}
void backward()
{
```

```
 r1_motor.run(BACKWARD);      // turn it on going forward
 r2_motor.run(BACKWARD);
 l3_motor.run(BACKWARD);
 l4_motor.run(BACKWARD);
}
void Stop()
{
 r1_motor.run(RELEASE);      // turn it on going forward
 r2_motor.run(RELEASE);      // turn it on going forward
 l3_motor.run(RELEASE);
 l4_motor.run(RELEASE);
}
void  loop ()
{
 distance=ultrasonic.Ranging(CM);
 delay(80);
 if(distance>30)
{
```

```
   Forward(110,130);
 }
 if(distance<=30 && distance>15)
{
    Forward(100,120);
 }
 if(distance<=20)
{
  Stop();
  delay(15);
  right(120,15);
   }
 }
```

step

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The purpose of our project was to create an inexpensive and high-efficient obstacle avoidance robot for detecting obstacle and avoiding the collision. We built a robotic vehicle which moves in different directions like Forward, Backward, Left, and Right when input is given to it.The goal of our project is to create a autonomous robot which intelligently detects the obstacle in his path and navigate according to the actions that we set for it.

5.2 RECOMMENDATION

We recommend improving this robot making it moves from the starting point to the target point so beyond any obstacle in its path and then return to the same path until it reaches the target point

REFERENCES

[1] The basics of robotics byFareed Shakhatreh

[2] FUNDAMENTALS OFROBOTICS BY H. Bunke (Univ. Bern, Switzerland)

[3] Robot Institute of America, 1979

[4] http://www.google.org/robotics/intro.html

[5] Obstacle Detection And Avoidance by a Mobile Robot BY sejal jaiswal)

[6] https://en.wikipedia.org/wiki/History_of_robots

[7]http://tech.endeepak.com/blog/2016/01/02/simple-obstacle-avoiding-robot-using-arduino

[8] http://blog.robotiq.com/what-is-the-best-programming-language-for-robotics

[9] THE BASICS OF ROBOTICS Fareed Shakhatreh

APPENDLX

The code of the robot:

```
//4wd Avoiding robot Code
#include <AFMotor.h>
 #include  "Ultrasonic.h"
int distance = 0;
 AF_DCMotor r1_motor(1, MOTOR12_64KHZ); // create motor #2, 64KHz
 AF_DCMotor r2_motor(2, MOTOR12_64KHZ); // create motor #2, 64KHz
 AF_DCMotor l3_motor (3, MOTOR12_64KHZ); // create motor #2, 64KHz
 AF_DCMotor l4_motor (4, MOTOR12_64KHZ); // create motor #2, 64KHz
 Ultrasonic ultrasonic (12,13);
 void  setup () {
  r1_motor.setSpeed(200);     // set the speed to 200/255
  r2_motor.setSpeed(200);     // set the speed to 200/255
  l3_motor.setSpeed(200);     // set the speed to 200/255
  l4_motor.setSpeed(200);     // set the speed to 200/255
 delay(2000);
 }
```

```
void Forward(int SpeedR,int SpeedL){
 r1_motor.setSpeed(SpeedR);     // set the speed to 200/255
 r2_motor.setSpeed(SpeedR);     // set the speed to 200/255
 l3_motor.setSpeed(SpeedL);     // set the speed to 200/255
 l4_motor.setSpeed(SpeedL);     // set the speed to 200/255
  r1_motor.run(FORWARD);      // turn it on going forward
 r2_motor.run(FORWARD);      // turn it on going forward
 l3_motor.run(FORWARD);
 l4_motor.run(FORWARD);
 }
void right(int SPEED,int timer)
{
 r1_motor.setSpeed(SPEED);     // set the speed to 200/255
 r2_motor.setSpeed(SPEED);     // set the speed to 200/255
 l3_motor.setSpeed(SPEED);     // set the speed to 200/255
 l4_motor.setSpeed(SPEED);     // set the speed to 200/255
 r1_motor.run(RELEASE);      // turn it on going forward
 r2_motor.run(RELEASE);      // turn it on going forward
```

```
 l3_motor.run(FORWARD);
 l4_motor.run(FORWARD);
   delay(timer);
}
void left(int SPEED,int timer){
 r1_motor.setSpeed(SPEED);     // set the speed to 200/255
 r2_motor.setSpeed(SPEED);     // set the speed to 200/255
 l3_motor.setSpeed(SPEED);     // set the speed to 200/255
 l4_motor.setSpeed(SPEED);     // set the speed to 200/255
  r1_motor.run(FORWARD);      // turn it on going forward
 r2_motor.run(FORWARD);      // turn it on going forward
 l3_motor.run(RELEASE);
 l4_motor.run(RELEASE);
 delay(timer);
}
void backward(){
r1_motor.run(BACKWARD);      // turn it on going forward
 r2_motor.run(BACKWARD);
```

```
 l3_motor.run(BACKWARD);
 l4_motor.run(BACKWARD);
 }
void Stop()
{
 r1_motor.run(RELEASE);      // turn it on going forward
 r2_motor.run(RELEASE);      // turn it on going forward
 l3_motor.run(RELEASE);
 l4_motor.run(RELEASE);
 }
void  loop ()
 {
 distance=ultrasonic.Ranging(CM);
 delay(80);
 if(distance>30){
   Forward(110,130);
 }
if(distance<=30 && distance>15){
```

```
    Forward(100,120);
}
if(distance<=20){
  Stop();
  delay(15);
  right(120,15);
 }
}
```