INDEX

Chap 1. INTRODUCTION

Our aim is to make a vehicle which can be controlled by the voice command of a person. Normally these types of systems are called as Speech Controlled Automation System (SCAS). Our design is a prototype of the above mentioned system. The idea is to create a sort of vehicle which going to be driven by voice commands. The car is remotely controlled by a mobile phone; there are many articles that show the communication between a robot and smart phone. Smart phone is a very good interface for remotely automating the robot.

It contains many features that can be helpful. In this design, an android application with a micro controller is used for the required task. The connection between the application and the robot is facilitate with Bluetooth technology. The commands issued will be relayed over through the channel and will be received by the module. The objective of Arduino voice controlled vehicle (AVCV) is to listen and act on the commands of the user. Here the system requires accent training, post which the device will start understanding the commands issued; and the commands have been added by codes.

The main motive to build a AVCV is to analyze the human voice and act according to the programmed commands. The most basic commands are backward, forward, right , left and also stop the car. The vehicle is to be controlled wirelessly with the use of android smartphone; our intention is to make a robotic vehicle with use of advanced smartphone technology in a very simple and economic way. In current scenario vehicles are manually controlled and all are done by the person who is driving the vehicle. Every action like start and stop, applying brake, gear transmission, acceleration requires human effort.

But nowadays new technologies have been developed that can be integrated with the conventional vehicles to new vehicle form. In the technology era, the space between the physical and the digital world is brought closer by the introduction of gesture concept. For all dangerous tasks, we prefer technology rather than people. Even though these robots are being controlled manually in the early stages, these can now be controlled via voice and gestures. This technology of gesture and voice recognition can be defined by the interaction between the computer and the body language of human beings. This constructs the communication link between technology and mankind. The target of this work is to upgrade the complete security to the robot and to simplify the controlling mechanism. The voice directions are handled, utilizing an advanced mobile phone. The individual human right hand robot is created on a smaller scale controller based stage and can know about its present area.

The viability of the voice control conveyed over a separation is estimated through numerous examinations. Execution assessment is completed with consequences of the underlying investigations. The developments to be forecasted are possibly referring to the applications in ventures, medical clinics and how, including the environmental laboratories .Providing human labor is the biggest problem all over the world. With the help of this device, they can move in a wheelchair on their own by just giving voice commands through Bluetooth. If any obstacle is detected while moving it informs the user and stops. It also detects if any fire and smoke occur due to emergency, detects and informs the prescribed number to get the required help. A prototype is developed by incorporating all the features in a single module.

The hardware part consists of the mechanical design of the robot, the adequate choice of the motors, and the electronic devices to properly drive the robot joints. The software part contains the high level algorithms that convert the desired word to a sequence of target points, and the control algorithms that ultimately make the robot move according to the specifications. Here the writing mechanism is made by speech recognition technique. This speech recognition can be provided through either by using microphone or by using android applications. The purpose is to implement a voice-controlled system as an Intelligent Personal Assistant (IPA) that can perform numerous tasks or services for an individual. This golem is specially designed for this cluster of individuals as its main purpose is to supply help to associate senior or disabled person.

It is to develop a robotic vehicle using Arduino and to controls the vehicle with the help of voice based information. The whole mechanism of the project is based upon the device namely Arduino. The feature of “hand gesture” helps it to move wirelessly depending upon the Radio Frequency, which is placed upon the hand of the user. The voice command was given to the Android smartphone that converts speech to text and it communicates with the control unit through Bluetooth.

To control the robot, it is necessary to compile a library of words that the operator can input to achieve the set goals. The main elements of this library are discussed and defined through the article. To provide voice control, a sequence of actions is provided, which must be followed by the operator. Also, the commands that may be needed in the case of interactive correction of actions are considered. The Android application (AMR – Voice) is connected to Bluetooth Module (HC – 05), which is directly connected to Arduino Uno R3. We give command to the robot and it performs work according to the given command. Voice Control Robot is much useful for those areas where humans can’t reach. Robot can work in all type of situations like toxic area, in fire situations, polluted area and also on hills. This robot is very useful for those who is physically challenged.

The Bluetooth module HC-05 will be interfaced with Arduino on the car for receiving the voice command. The Android app will be used for providing the specific command to the car. Now, on the receiving phase, a Bluetooth module HC-05 will receive the commands. The main goal is to control the vehicle in a desired position and also to control the robot by the voice or push buttons. Human Robotic Interaction is achieved. The goal of voice controlled Car is to listen and act on the commands received from the user.

1.1 EXISTING SYSTEM

The current systems are robots like line follower robot, edge averting robot, DTMF robot, gesture controlled robot. These type of robots are not efficient since they require more power to run, cost is also very high. In the existing system they don’t use voice commands, making it not possible for physically handicapped people to drive.

The voice commands are interpreted via an offline server in real time. The commands are at once transmitted to the server directly by the means of a wired network. The car is built primarily on a platform based on a microcontroller.

Some of the fields that can likewise be equally enhanced are the effect of the mouth-microphone range on the robotic, the overall performance (scope) of the robot and the effect of noise on the translation of speech to textual content. In the existing system Bit Voicer Server is used, it's a database for speech processing and automation synthesis.

It was designed to make voice operation possible with simple gadgets having low processing power. Microcontrollers usually do not have enough storage and computing ability to perform sophisticated speech treatment and synthesis. By doing the tough work Bit Voicer Server removes the consequences of these limitations so that the microcontroller can assign its key functionality to most of its origin sources.

1.2 PROPOSED SYSTEM

In this proposed device we perform a variety of research on control style variants for robots. It shows that it’s feasible to study to successfully manipulate actual world objects with solely voice (human voice) as a control mechanism. The reason of this lookup is to provide simple robotic hardware architecture so that this shape can focally point on Bluetooth connection infrastructure.

It is also beneficial for academic robotics due to the fact human beings can construct their personal robots with low cost. When the app is operating in the system, a microphone on the mobile is used to identify user voice commands. Commands are interpreted and the program utilizes Google's speech-recognition software to translate voice to text within the app.

Arduino is an open source programmable circuit board that can be integrated into a wide variety of makerspace projects both simple and complex. This board contains a microcontroller which is able to be programmed to sense and control objects in the physical world. Arduino IDE is a special software running on your system that allows you to write sketches (synonym for program in Arduino language) for different Arduino boards

The text will then be sent with the aid of Bluetooth to the receiver part. The microcontroller Arduino UNO R3 has 32kB of ISP flash memory, 2kB of RAM and 1kB of EEPROM. The panel incorporates serial communication connectivity with UART, SPI and I2C. The MCU will operate at 16MHz clock speed. The digital Arduino I / O pins 3, 4, 5 and 6 are programmed as output pins in this design. For serial communication with the Bluetooth unit, pins 0 and 1 of Arduino are used. Text obtained with the aid of Bluetooth is forwarded to Arduino UNO microcontroller panel by the usage of UART serial conversation protocol.

The voice commands to the robotic device are dispatched via Bluetooth with the aid of an Android device. These commands are received on the robotic device by using Bluetooth module set up on it. The motor driver circuit is used to manipulate the velocity of the car. The complete circuitry is powered by the usage of a 12V rechargeable battery hooked up on the system.

Chap 2. LITERATURE SURVEY

2.1 LITERATURE REVIEW OF PROJECT

Arduino Uno :- The Arduino Uno is a microcontroller board based on the ATmega328. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery.

TT Gear Motor :- These durable (but affordable!) plastic gearbox motors (also known as 'TT' motors) are an easy, low-cost way to get your projects moving. This is a TT DC Gearbox Motor with a gear ratio of 1:48, and it comes with 2 x 200mm wires with breadboard-friendly 0.1" male connectors. Perfect for plugging into a breadboard or terminal blocks.

Servo Motor :- A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor.

Ultrasonic Sensor :- An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object’s proximity.

HC-05 Bluetooth Module :- It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications. It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions

IR Proximity Sensor :- A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact. or return signal. The object being sensed is often referred to as the proximity sensor's target

2.2 SYSTEMATIC REVIEW

Arduino Uno :- The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

Ultrasonic Sensor :- Ultrasonic transducers and ultrasonic sensors are devices that generate or sense ultrasound energy. They can be divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

HC-05 Bluetooth Module :- HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds. This module works on 3.3 V.

IR Proximity Sensor :- Proximity sensors are sensors that detect the movement/presence of objects without physical contact and relay that information captured into an electrical signal. It can also be defined as a proximity switch, a definition given by the Japanese Industrial Standards (JIS) to all contactless detecting sensors

Chap 3. SCOPE OF PROJECT

When the app is running in the smartphone, the user’s voice commands are detected by the phone microphone.

The circuit consist of Arduino UNO Board, HC-05/HC-06 Bluetooth Module, L293D Motor Driver IC, a pair of DC Geared Motors of 200 RPM and a 9V Battery.

The TX, RX pins of Arduino is connected to Rx, Tx pins of Bluetooth Module. The Bluetooth Module is supplied with 5V. Similarly, left DC motor is connected to pin no 3 & 6 of L293D and right DC motor to pin no 14 & 11 of L293D. Arduino digital pins 3,4,5,6 is connected to L293D 2, 7, 10, 15 respectively.

The L293D IC Pins 2, 5, 12, 13 is GND pins and 9, 1, 16 is supplied with 5V. But pin 8 of L293D

is directly supplied with 9V.

Android smartphone with app. Android speech-recognition app (voice control.apk) used here was developed using MIT App Inventor. When the app is running in the smartphone, user’s voice commands are detected by the microphone present in the phone.

Commands are processed, and speech-to-text conversion is done within the app using Google’s speech-recognition technology. Text is then sent to the receiver side (that is, robotic car) via Bluetooth.

Chap 4. METHODOLOGY

Arduino Uno

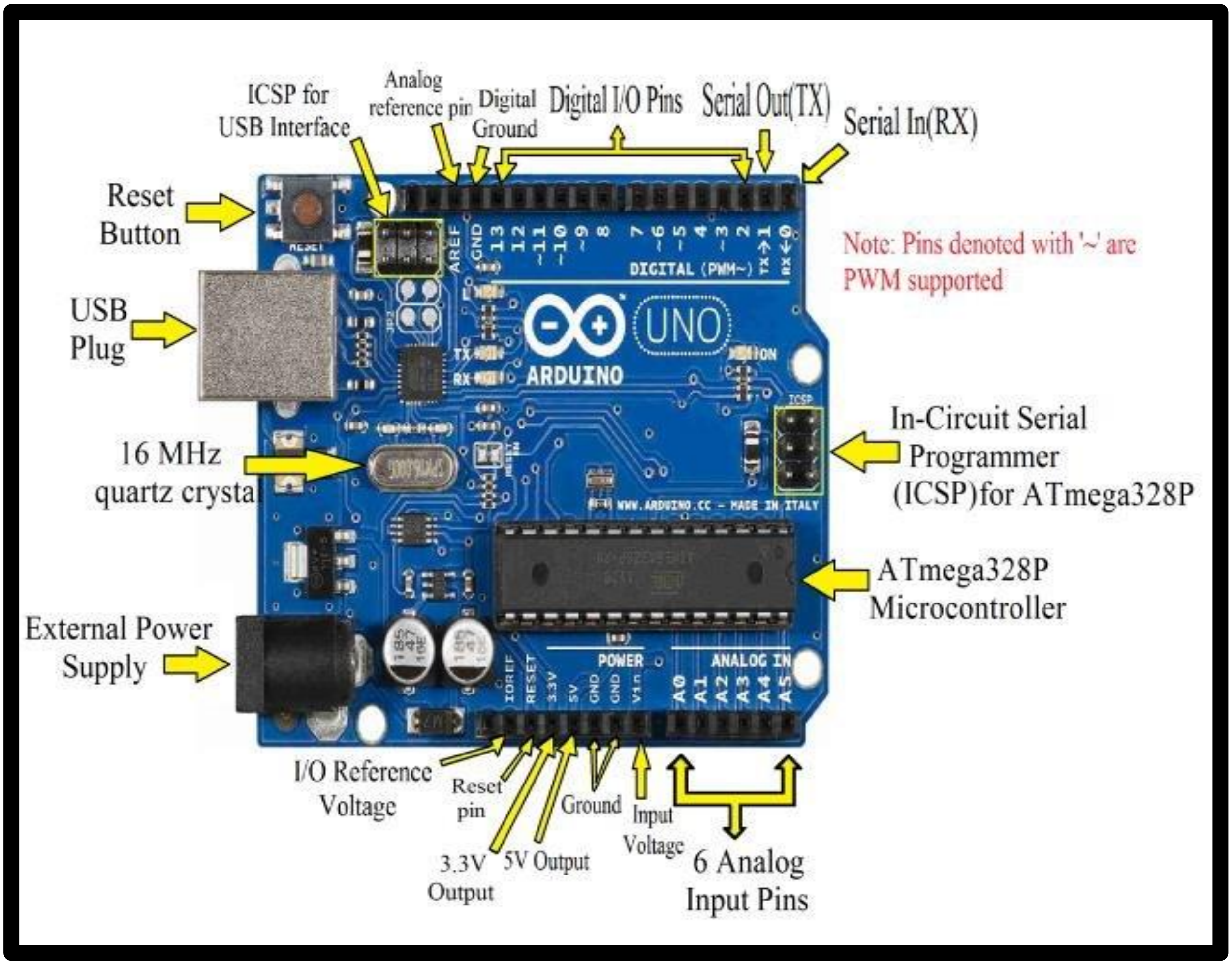


Fig.4.1 Arduino Uuno

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was 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 is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

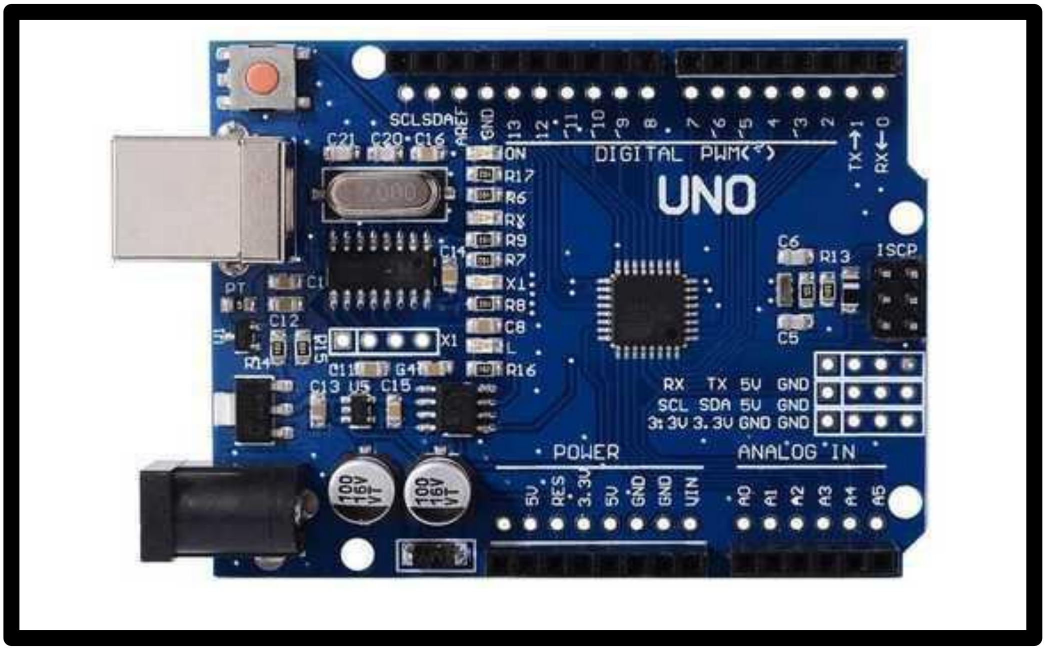


Fig.4.2 Arduino Uuo

Programming

The Arduino/Genuino Uno can be programmed with the (Arduino Software (IDE)). Select "Arduino/Genuino Uno from the Tools > Board menu (according to the microcontroller on your board).

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

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

1. On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then rese ing the 8U2.
2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

Warnings

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

Differences with other boards

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Power

The Arduino/Genuino Uno board 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 from 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.

Memory

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library). Input and Output. The mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm.

A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

The power pins are as follows:

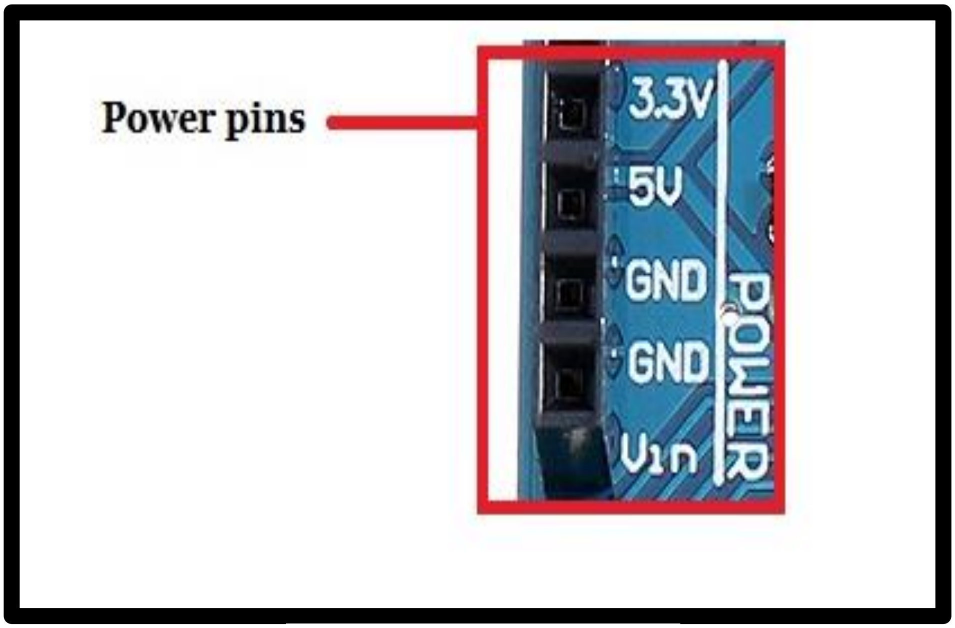


Fig.4.3 Power Pins

1. Vin. The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
2. 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
3. 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
4. GND. Ground pins.

IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

In addition, some pins have specialized functions:

1. Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
2. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
3. PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
4. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
5. LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
6. TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.
There are a couple of other pins on the board:

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

Communication

Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer.

The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. 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 USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Motor Driver Shield

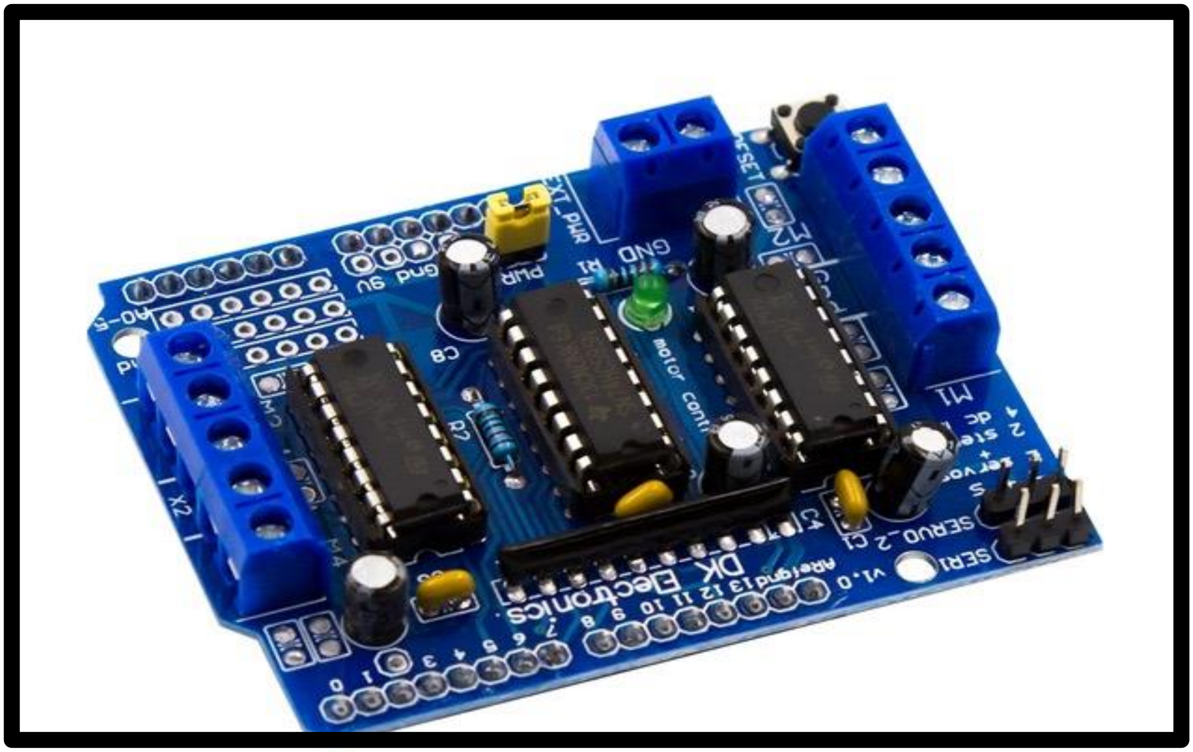


Fig.4.4 Motor Driver Shield

The Arduino L293D motor driver shield guide is a robobtics project that involves driving various types of motors. The most common types used for robotic applications include DC, servo, and stepper motors.

However, these motors typically cannot be driven directly by Arduino or another microcontroller. This is because of their higher current and power ratings, so motor shields or driver ICs are used instead. These shields or ICs isolate a motor’s power supply and use control logic from the microcontroller circuitry.

One of the most popular motor driver shields used with Arduino is the L293D. The full-featured L293D motor driver shield can control up to four bi-directional DC motors with 8-bit speed selection, two stepper motors, and two servo motors.

The motor driver

The L293D is a dual-channel H-bridge motor driver that can control two DC motors or a stepper motor at one time. As there are two L293D ICs on the shield, it’s technically capable of controlling a total of four DC motors.

This is ideal for two and four-wheel robot platforms. The IC consists of two H-bridge to control the motors, each delivering p to 0.6A to a motor.

The shift register

The 74HC595 is an 8-bit serial input and serial/parallel output shift register. It’s used to extend four Arduino GPIO (or another microcontroller) to eight direction control pins for two of the L293D motor driver ICs.

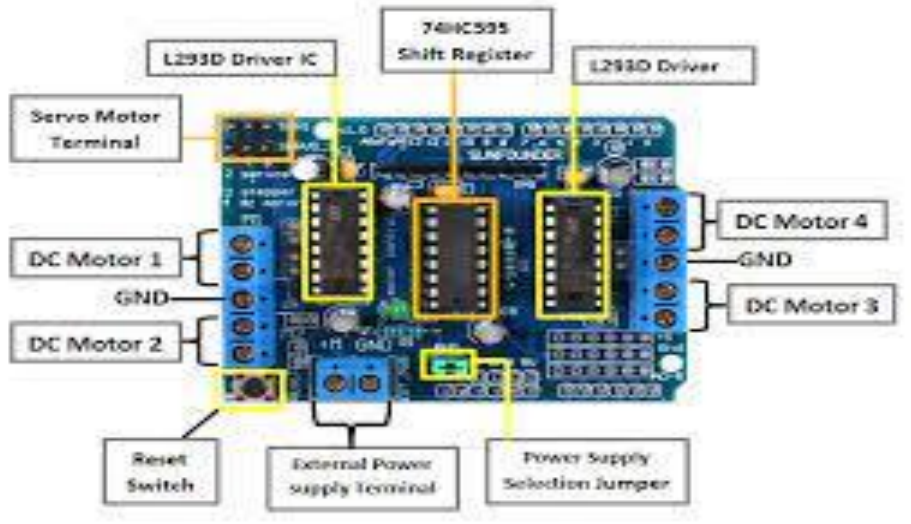


Fig.4.5 Motor Driver Shield

TT Gear Motor

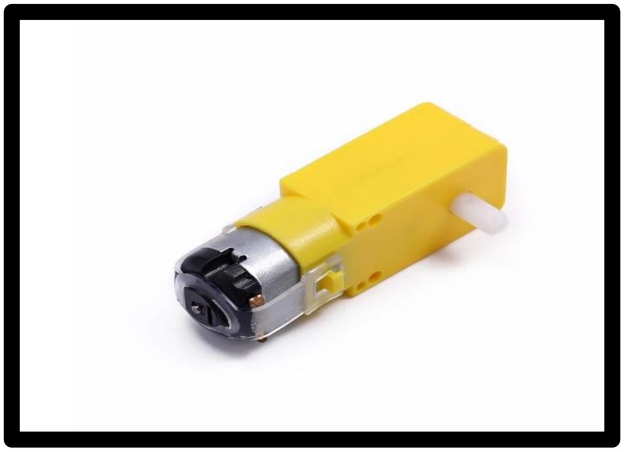


Fig.4.6 TT Gear Motor

These durable (but affordable!) plastic gearbox motors (also known as 'TT' motors) are an easy, low-cost way to get your projects moving. This is a TT DC Gearbox Motor with a gear ratio of 1:48, and it comes with 2 x 200mm wires with breadboard-friendly 0.1" male connectors. Perfect for plugging into a breadboard or terminal blocks.

You can power these motors with 3VDC up to 6VDC, they'll of course go a little faster at the higher voltages. We grabbed one motor and found these stats when running it from a bench-top supply

TECHNICAL DETAILS

1. Rated Voltage: 3~6V
2. Continuous No-Load Current: 150mA +/- 10%
3. Min. Operating Speed (3V): 90+/- 10% RPM • Min. Operating Speed (6V): 200+/- 10% RPM
4. Torque: 0.15Nm ~0.60Nm
5. Stall Torque (6V): 0.8kg.cm

Servo Motor

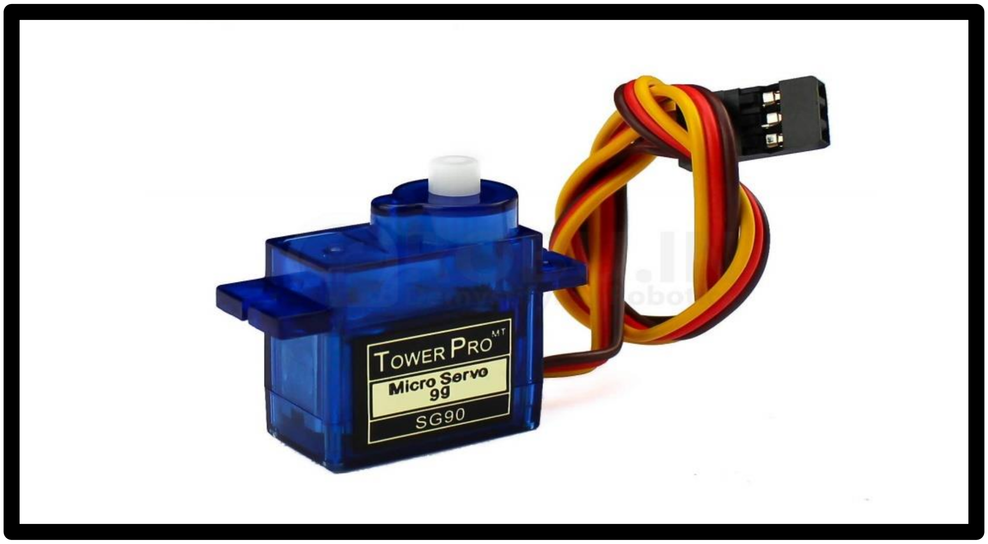


Fig.4.7 Servo Motor

A servo motor is a type of motor that can rotate with great precision. Normally this type of motor consists of a control circuit that provides feedback on the current position of the motor shaft, this feedback allows the servo motors to rotate with great precision. If you want to rotate an object at some specific angles or distance, then you use a servo motor.

It is just made up of a simple motor which runs through a servo mechanism**.** If motor is powered by a DC power supply then it is called DC servo motor, and if it is AC-powered motor then it is called AC servo motor. For this tutorial, we will be discussing only about the DC servo motor working. Apart from these major classifications, there are many other types of servo motors based on the type of gear arrangement and operating characteristics.

A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages. Due to these features, they are being used in many applications like toy car, RC helicopters and planes, Robotics, etc.

Ultrasonic Sensor

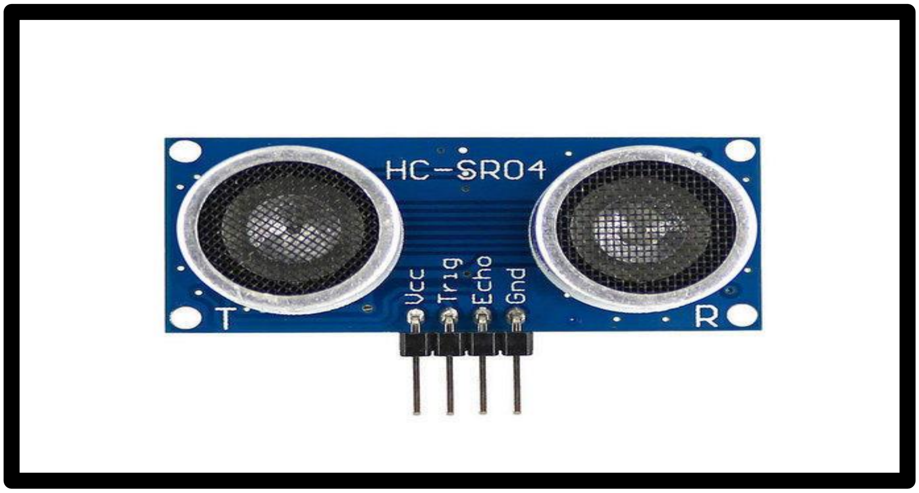


Fig.4.8 Ultasonic Sensor

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves.An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object’s proximity. High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

How Ultrasonic Sensors Work.

Ultrasonic sensors work by sending out a sound wave at a frequency above the range of human hearing. The transducer of the sensor acts as a microphone to receive and send the ultrasonic sound. Our ultrasonic sensors, like many others, use a single transducer to send a pulse and to receive the echo. The sensor determines the distance to a target by measuring time lapses between the sending and receiving of the ultrasonic pulse.

The working principle of this module is simple. It sends an ultrasonic pulse out at 40kHz which travels through the air and if there is an obstacle or object, it will bounce back to the sensor. By calculating the travel time and the speed of sound, the distance can be calculated.Ultrasonic sensors are a great solution for the detection of clear objects. For liquid level measurement, applications that use infrared sensors, for instance, struggle with this particular use case because of target translucence.

18650 Li battery

Fig.4.9 18650 Li Battery

An 18650 battery is a lithium-ion battery. The name derives from the battery’s specific measurements: 18mm x 65mm. For scale, that’s larger than an AA battery. The 18650 battery has a voltage of 3.6v and has between 2600mAh and 3500mAh (mili-amp-hours). (Osborne, 2019)

These batteries are used in flashlights, laptops, electronics and even some electric cars because of their reliability, long run-times, and ability to be recharged hundreds of times over. 18650 batteries are what would be considered a “high drain battery.” This means that the battery is designed to generate high output voltage and current to meet the power demands of the portable device in which it is being used.

Male and Female Jumper Wires

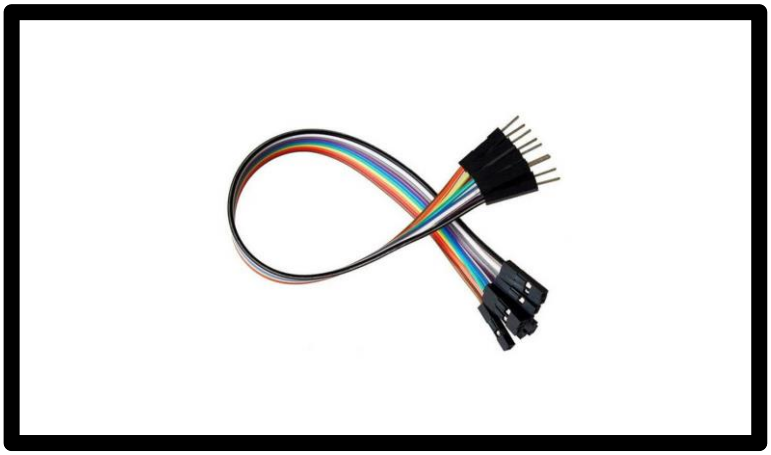


Fig.4.10 Male and Female Jumper Wires

A jump wire (also known as jumper, jumper wire, DuPont wire) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

HC-05 Bluetooth Module

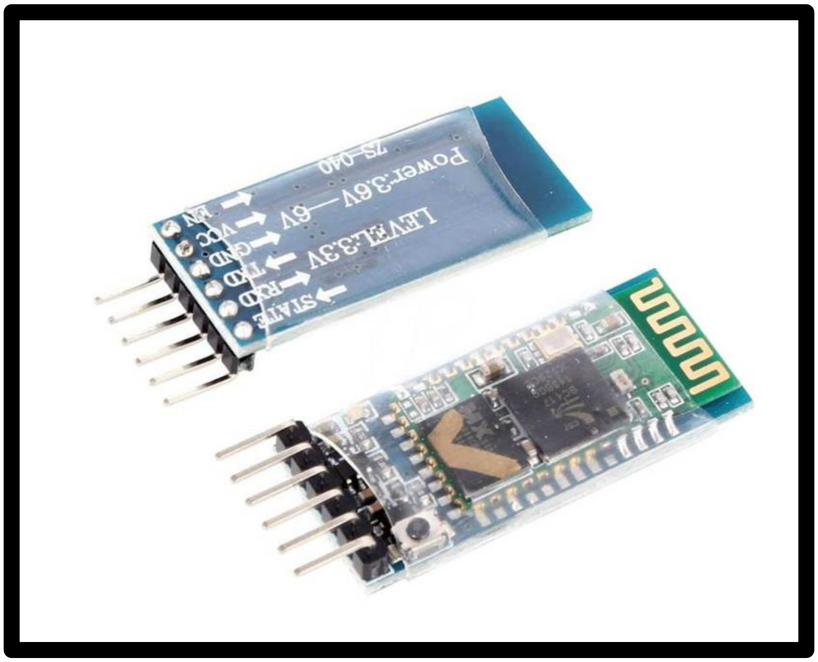


Fig.4.11 HC-05 Bluetooth Module

1. It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
2. It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network (PAN). It uses frequency-hopping spread spectrum (FHSS) radio technology to send data over air.
3. It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

Pin description

1. Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.
2. It has 6 pins,
3. 1. **Key/EN:** It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.
4. HC-05 module has two modes

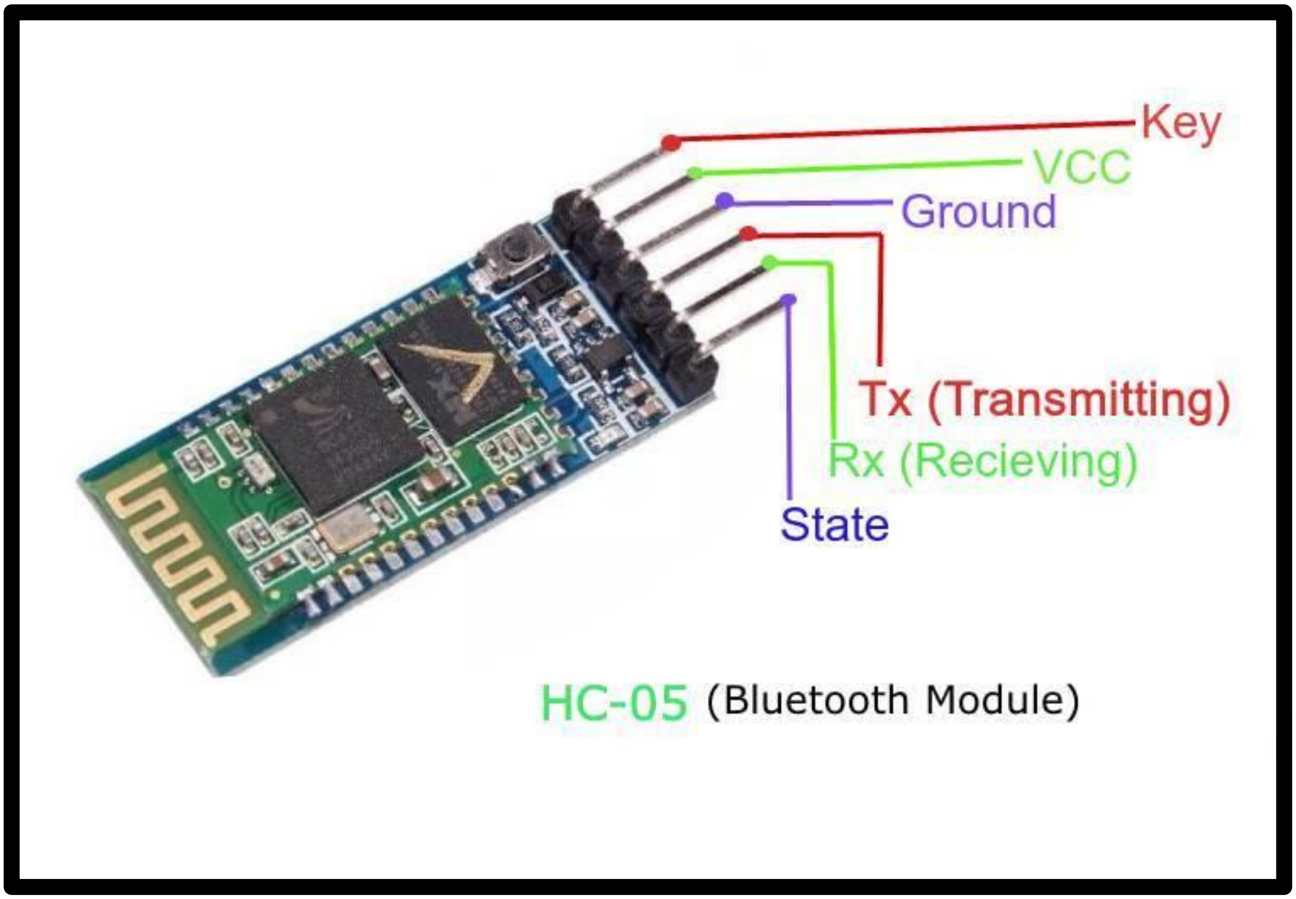


Fig.4.12 HC-05 Bluetooth Module

1. **Data mode**

   Exchange of data between devices.

2. **Command mode**

   It uses AT commands which are used to change setting of HC.

- To send these commands to module serial (USART) port is used.

a. **VCC:** Connect 5 V or 3.3 V to this Pin.
b. **GND:** Ground Pin of module.

c. **TXD:** Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)

d. **RXD:** Receive data serially (received data will be transmitted wirelessly by Bluetooth module).

e. **State:** It tells whether module is connected or not.

IR Proximity Sensor

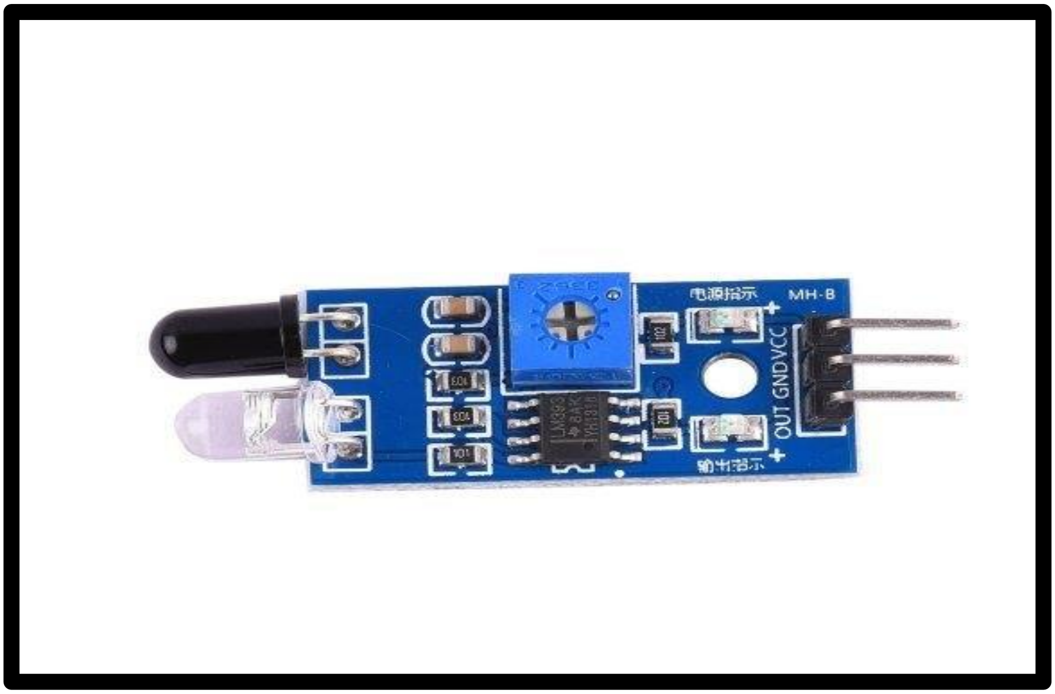


Fig.4.13 IR Proximity Sensor

A proximity sensor is a sensor able to detect the presence of nearby objects without any physical contact.

A proximity sensor often emits an electromagnetic field or a beam of electromagnetic radiation (infrared, for instance), and looks for changes in the field or return signal. The object being sensed is often referred to as the proximity sensor's target.

Different proximity sensor targets demand different sensors. For example, a capacitive proximity sensor or photoelectric sensor might be suitable for a plastic target; an inductive proximity sensor always requires a metal target.

Proximity sensors can have a high reliability and long functional life because of the absence of mechanical parts and lack of physical contact between the sensor and the sensed object.

Proximity sensors are also used in machine vibration monitoring to measure the variation in distance between a shaft and its support bearing. This is common in large steam turbines, compressors, and motors that use sleeve-type bearings.

A proximity sensor adjusted to a very short range is often used as a touch switch.

Chap 5. Details of Design, Working And Processes

Step 1. Connecting TT Gear Motor with Jumper Wires

We have to take a TT gear motor with male and female jumper wires and has to solder it with two if its end points.

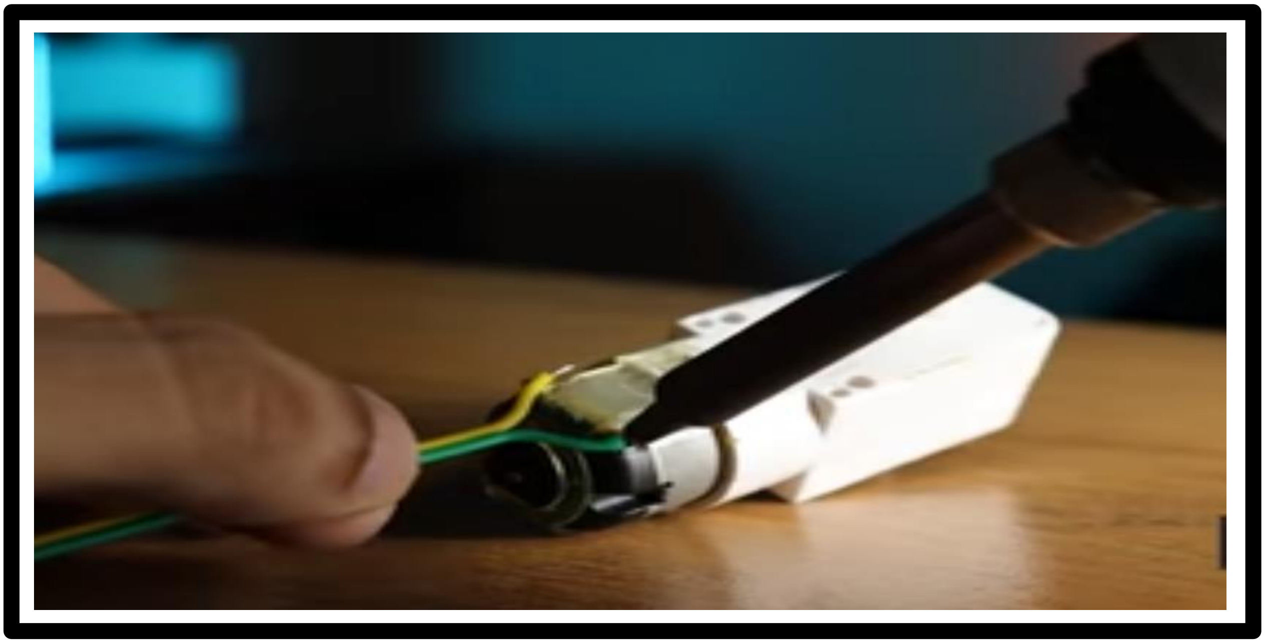


Fig 5.1. Soldering of TT Gear Motor

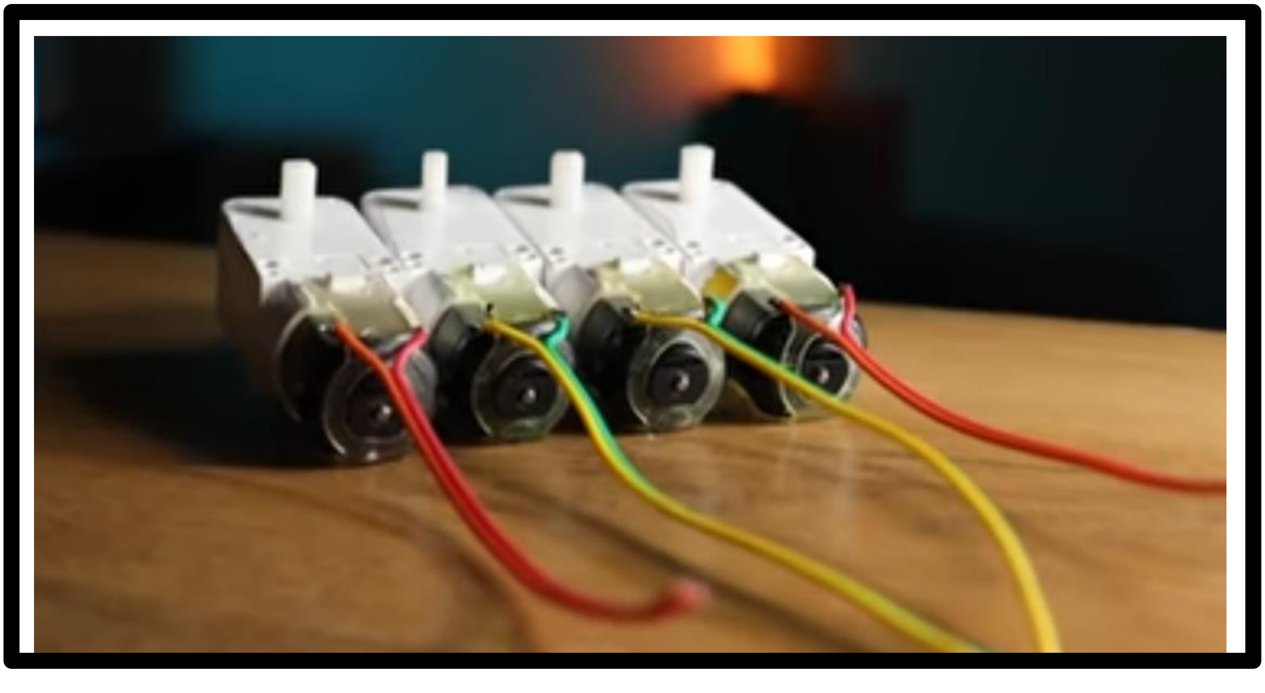


Fig 5.2. Set of soldered TT Gear Motor

Step 2. Designing of the sheet with the attachment of the motors

We have to use a acrylic sheet for the base of the car and have to cut according to the measurements and then we have to attach the four TT gear motors to the sheet with the help of a glue gun.

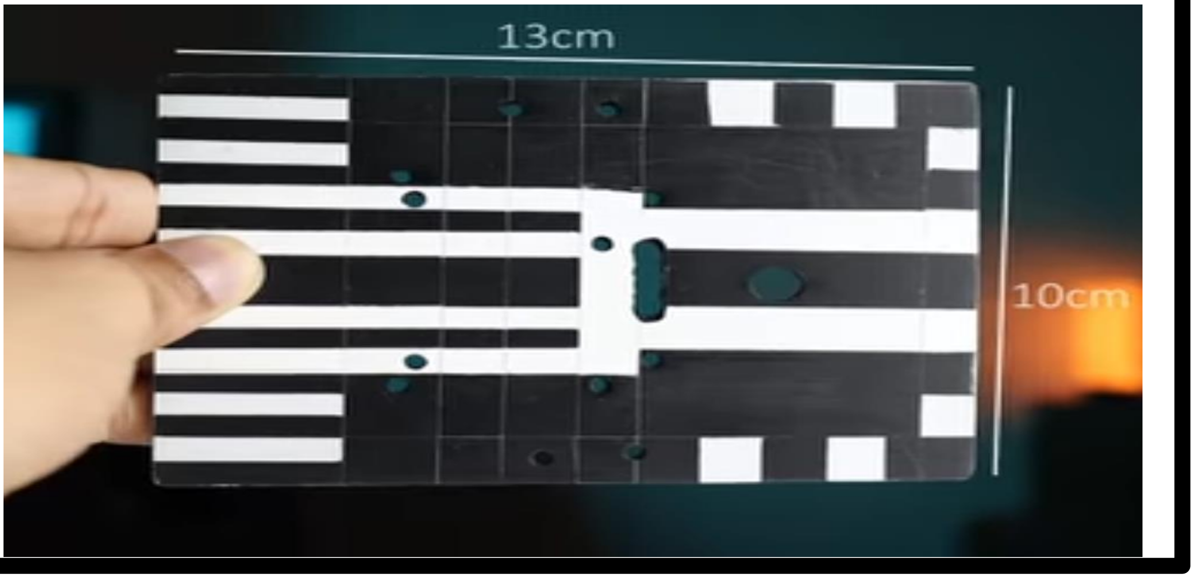


Fig 5.3. Designed Sheet

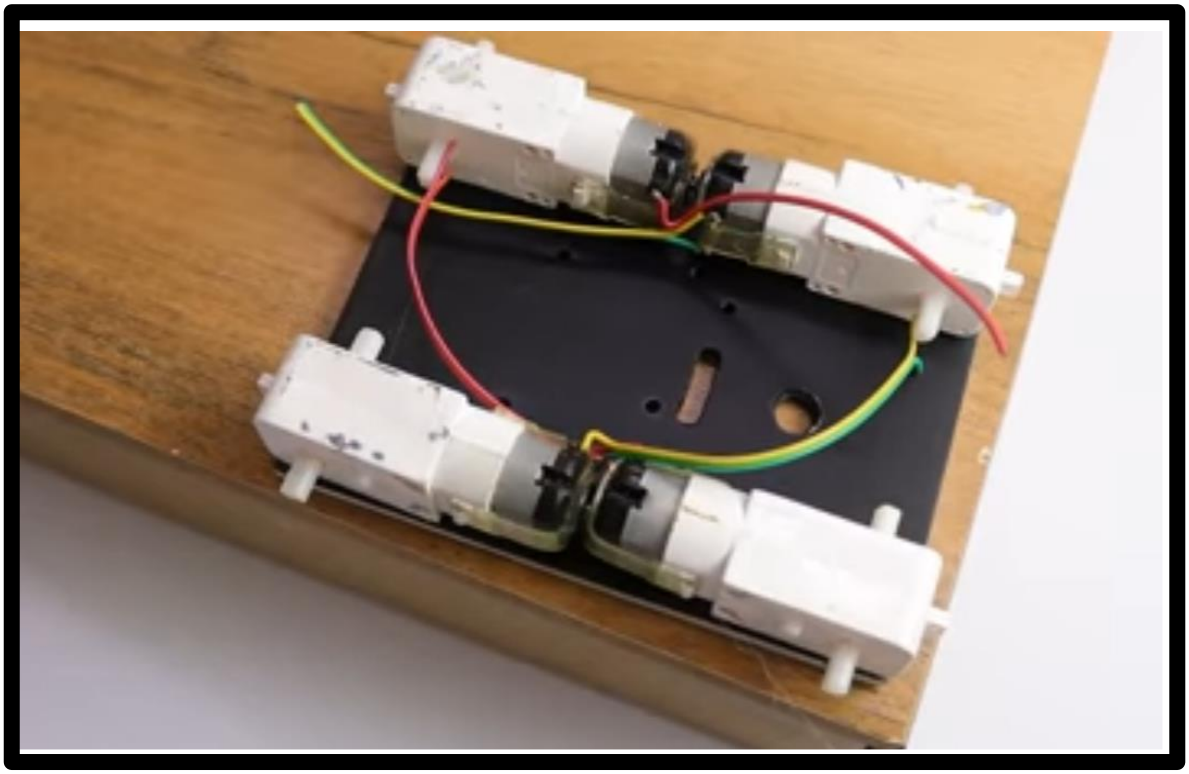


Fig 5.4. Attaching TT Gear Motor to the sheet

Step 3. Connect Arduino Uno and Driver Motor Shield

We have to screw the Arduino uno board on the acrylic sheet and the we have to mount the driver motor shield upon the Arduino uno board and then connecting the motor wire with the uno driver shield.

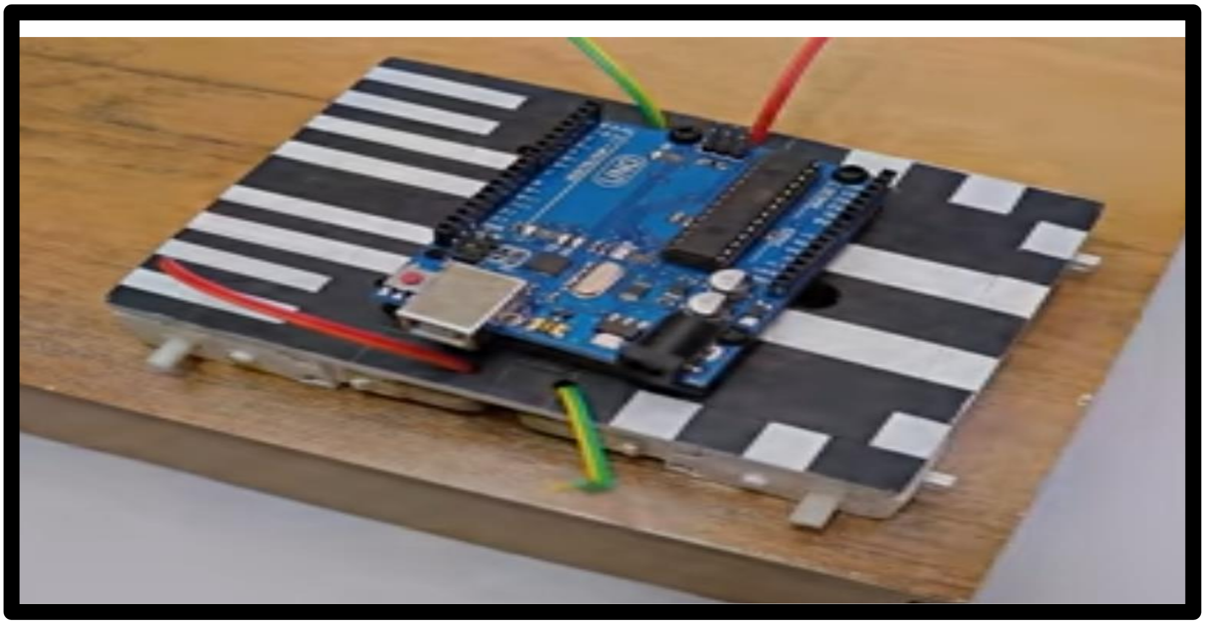


Fig 5.5. Connecting Arduino Uno

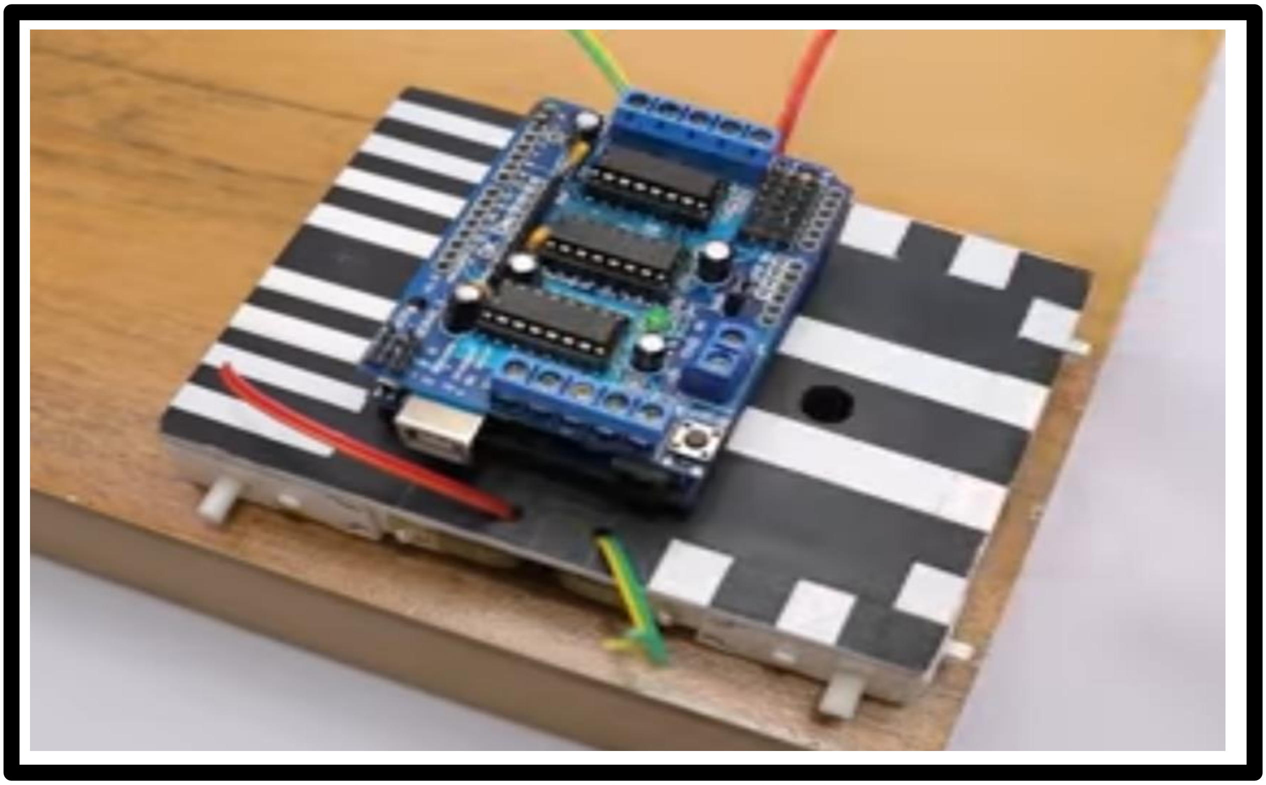


Fig 5.6. Mounting Driver Motor Shied to Uno

Step 4. Connecting the HC-05 Bluetooth Module and Wheels

We have to connect the connector with Bluetooth module and then we have to stick the module to thee sheet with double side tape and then we have to solder the connector wires to the points of the driver shield.

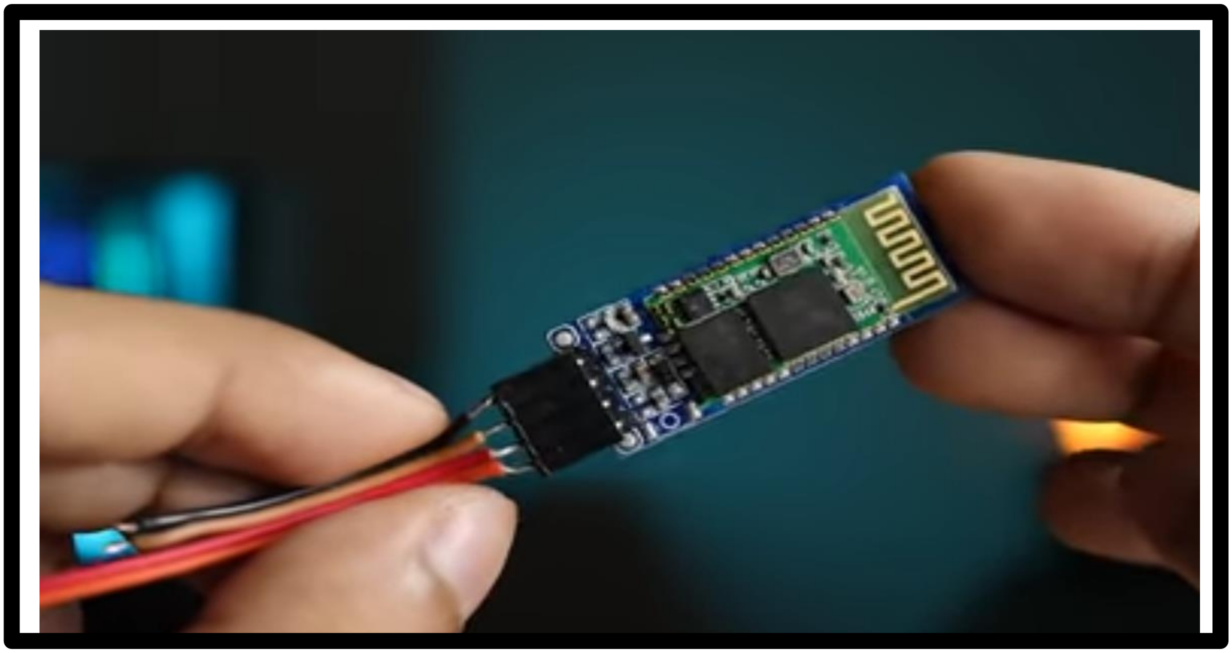


Fig 5.7. Connecting HC-05 Bluetooth Module

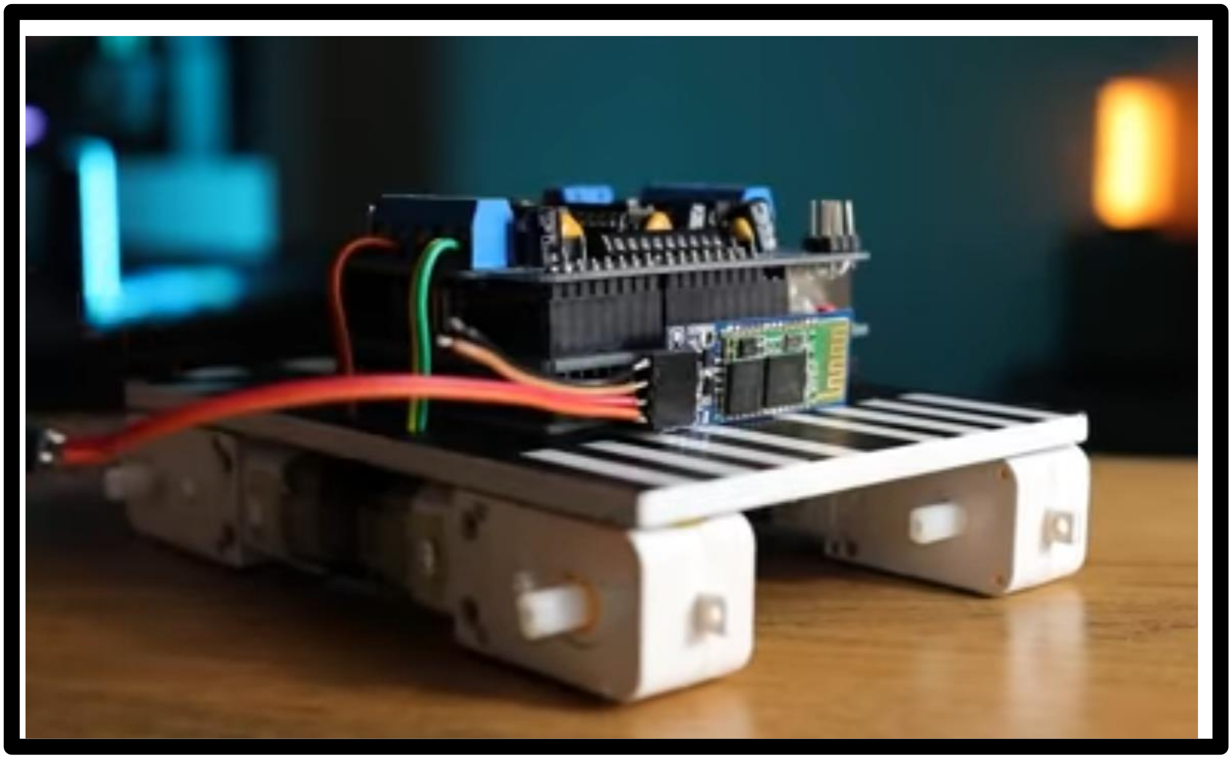


Fig 5.8. Attaching it to the sheet

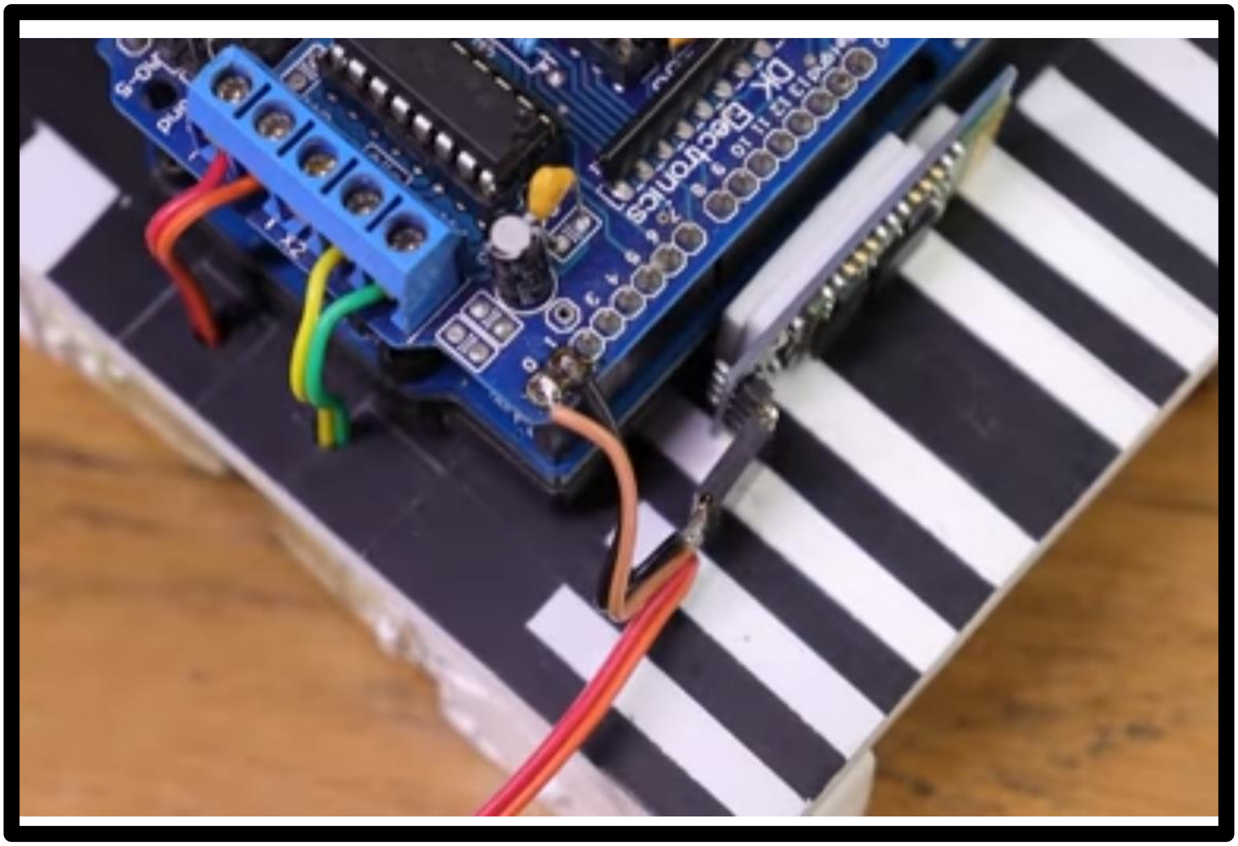


Fig 5.9. Mounting its connector

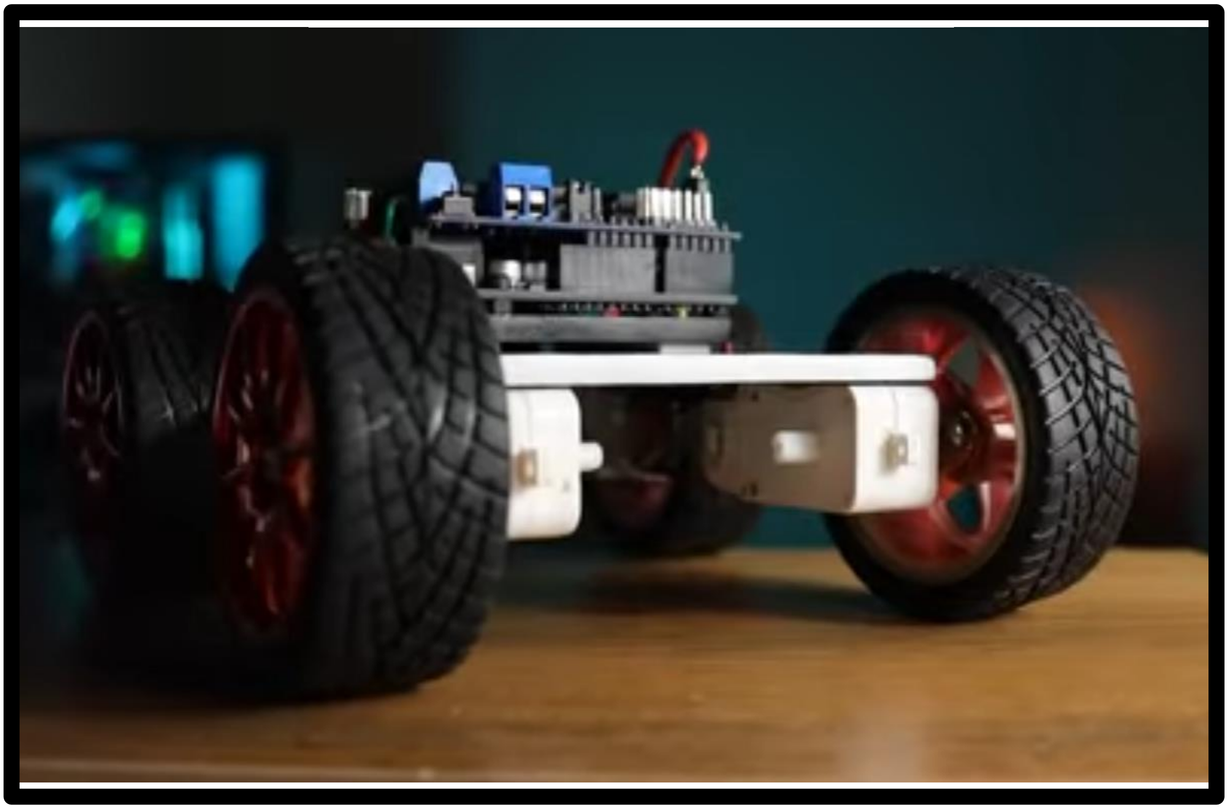


Fig 5.10. Adding Wheels

Step 5. Servo Motor Connections

We have to stick the servo motor to the acrylic sheet and then connect its wire to the driver motor shield.

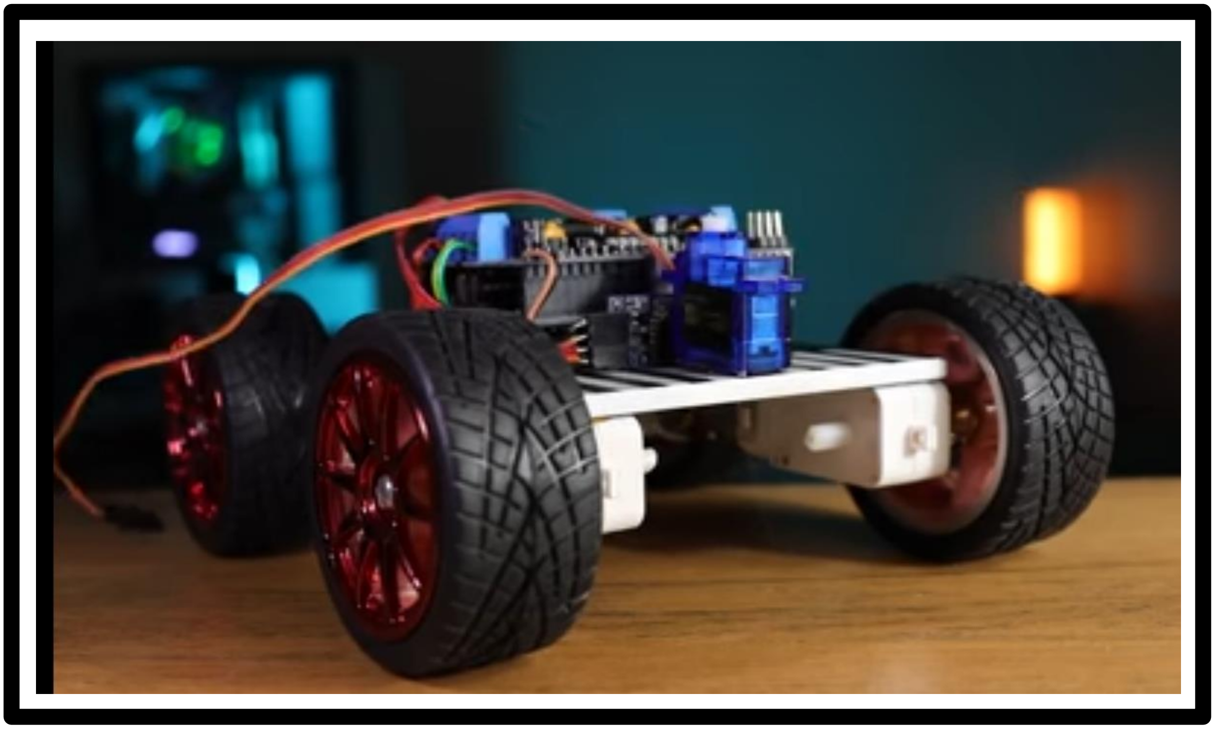


Fig 5.11. Connecting Servo Motor

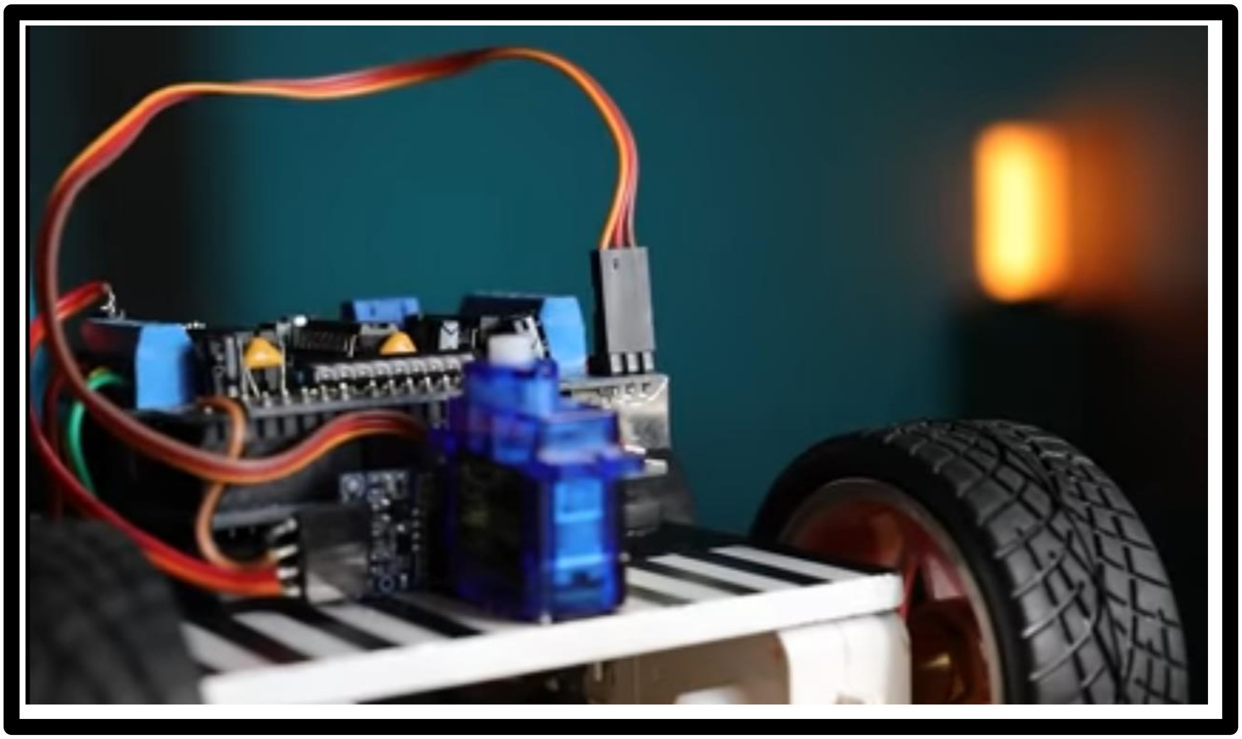


Fig 5.12. Mounting with the connector

Step 6. Attaching IR Proximity Sensor

At the backside of the base of the car where the gear motors have been placed there we have to stick the IR proximity sensor and the we have to mount it with the motor shield.

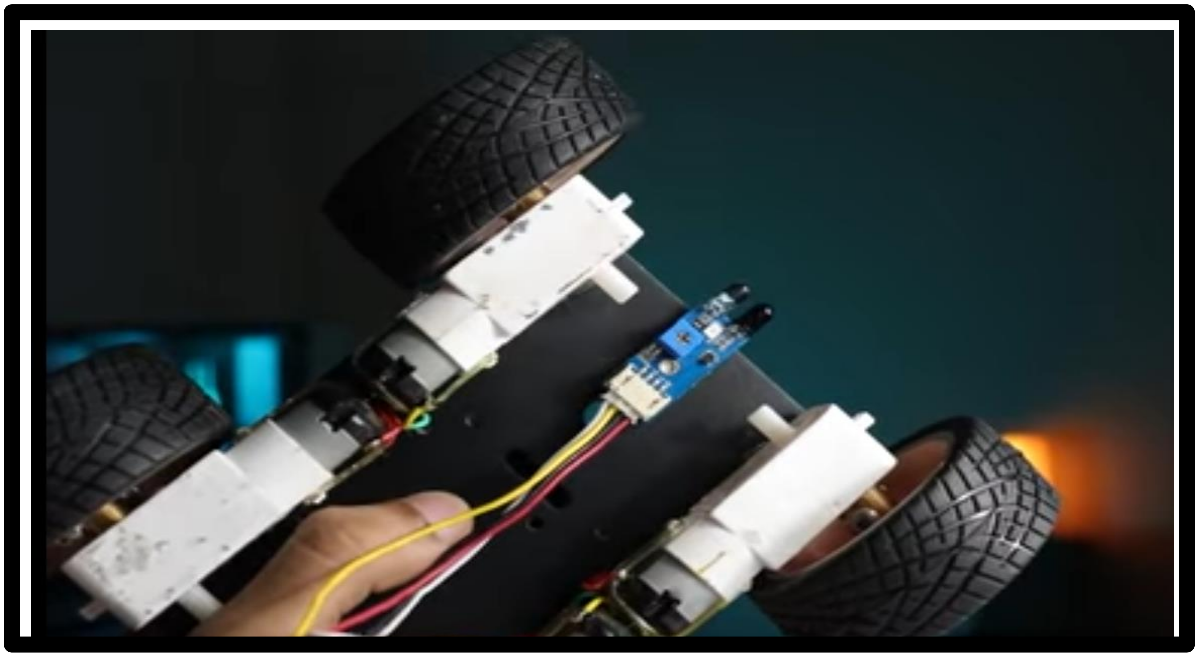


Fig 5.13. Attaching IR Sensor Module

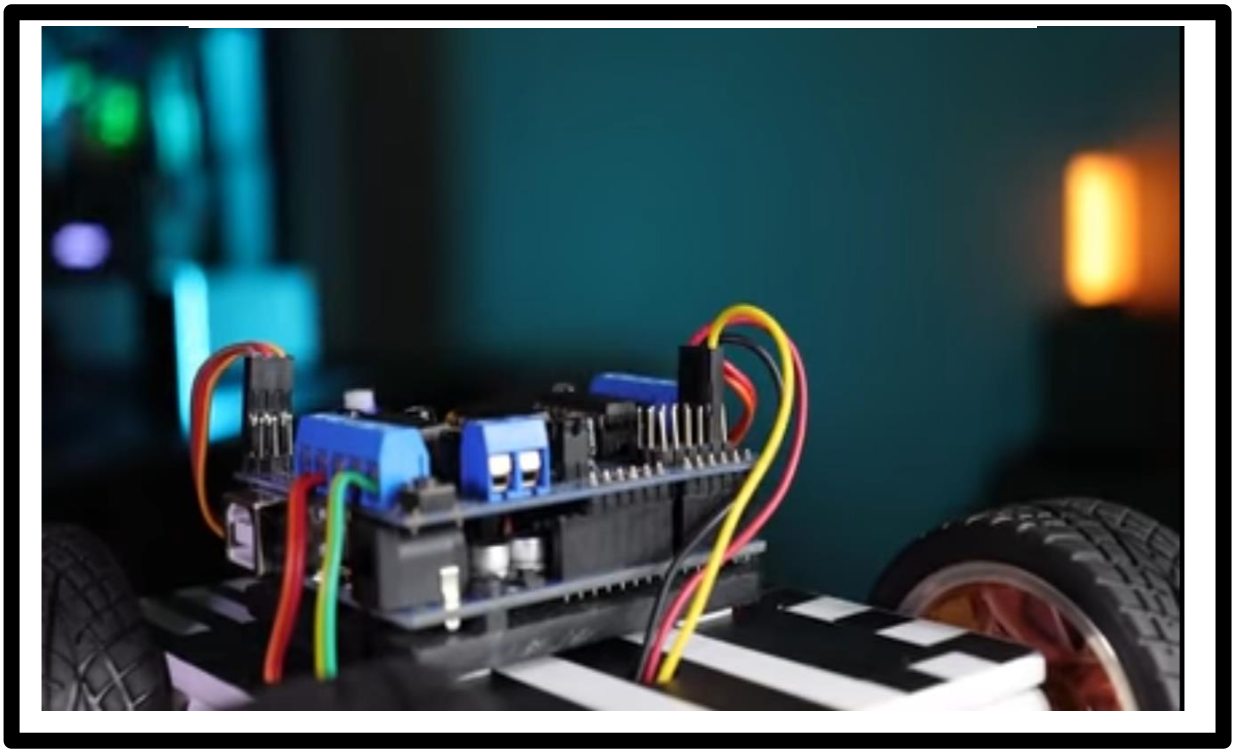


Fig 5.14. Mounting it with the connector to the shield

Step 7. Connecting Ultrasonic Sensor and Holder

We have to first connect the ultrasonic sensor to its holder then we have to connect the wires to the sensor and then wires have to be mount with the motor driver shield.

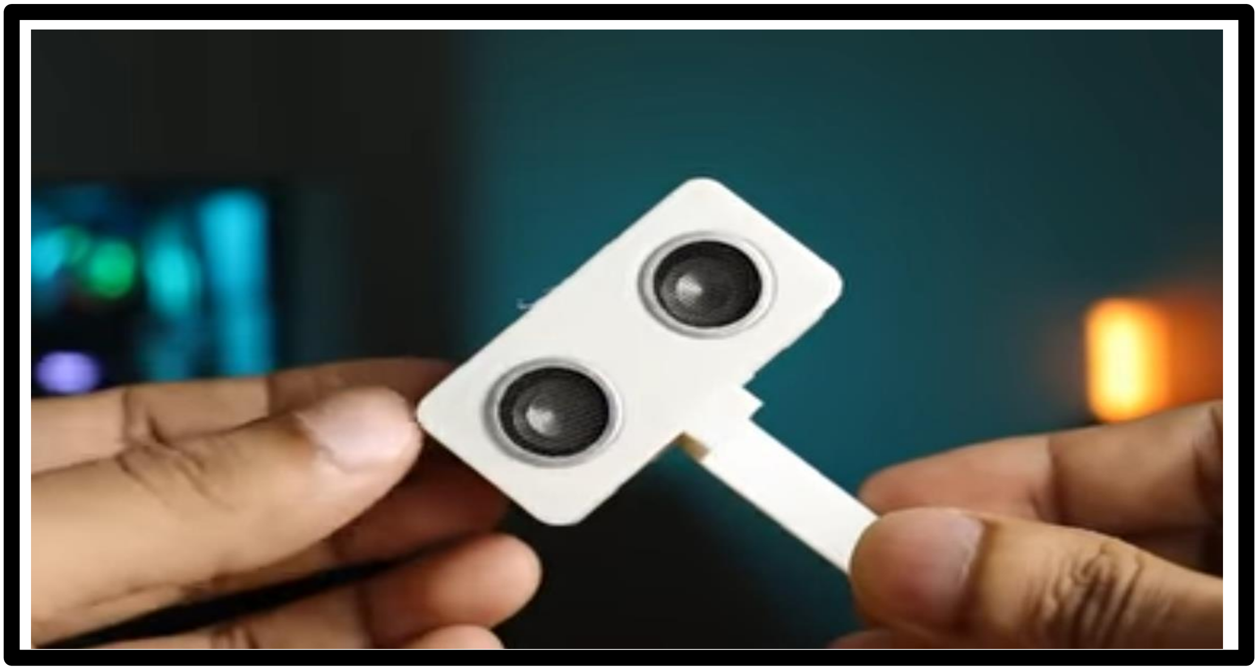


Fig 5.15. Connecting the holder to the sheet

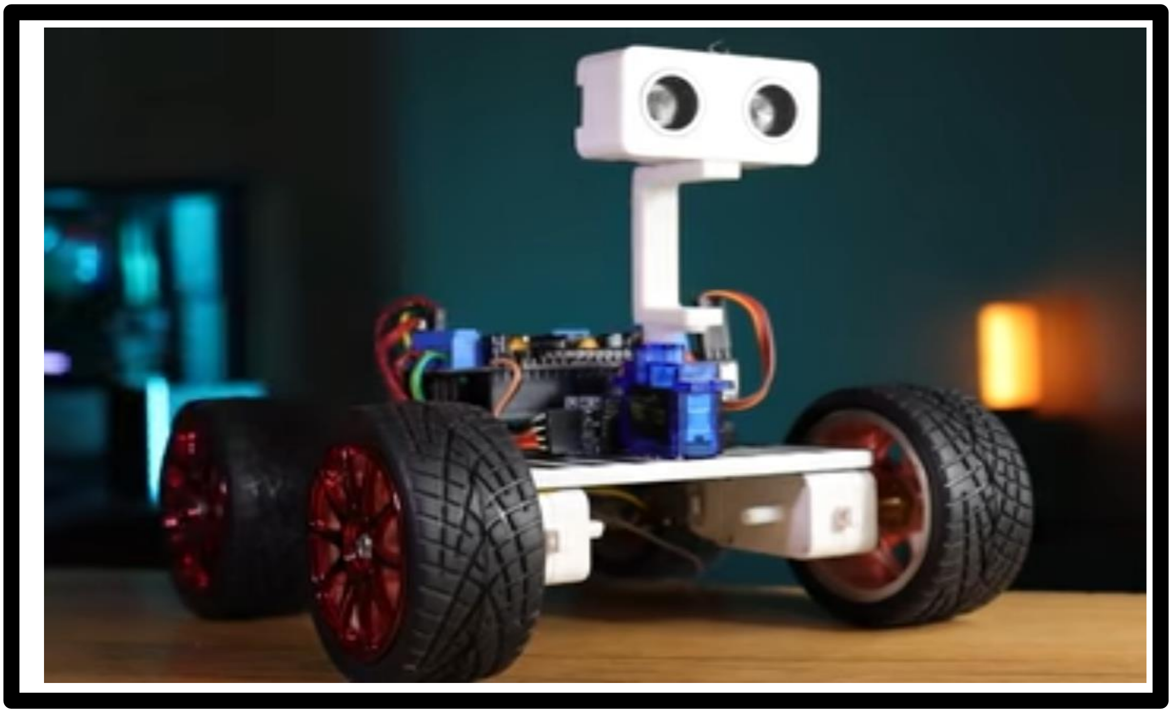


Fig 5.16. Attaching Ultrasonic Sensor to the Holder

Step 8. Uploading the piece od code

First, we have to remove the Bluetooth module and the we have to connect the Usb Arduino cord to the motor driver and then we have to upload the code from the Arduino Ide to the module and then we have to again attach the bluetooth module.

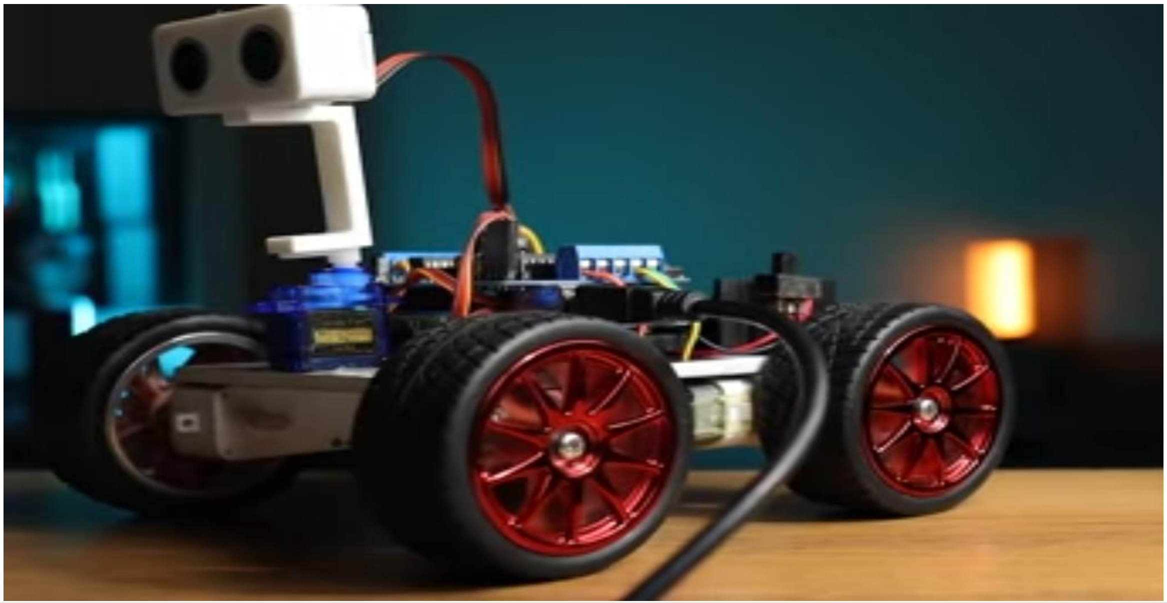
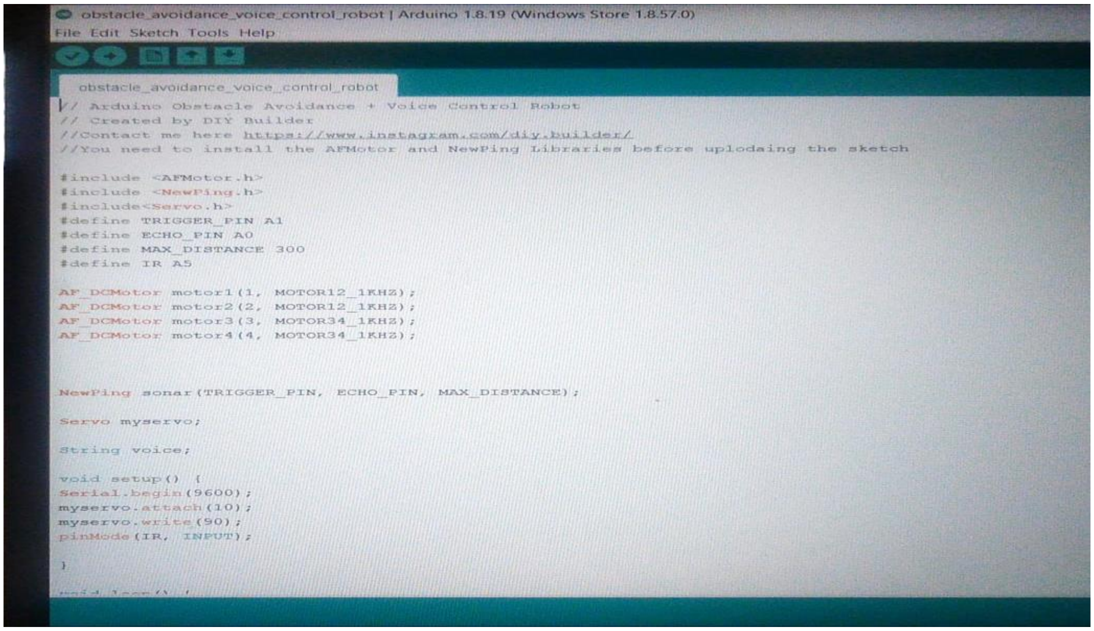


Fig 5.17. Connecting USB Cable



obstacle_avoidance_voice_control_robot | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
obstacle_avoidance_voice_control_robot

```
// Arduino Obstacle Avoidance + Voice Control Robot
// Created by DIY Builder
//Contact me here https://www.instagram.com/diy.builder/
//You need to install the AFMotor and NewPing Libraries before uplodaing the sketch

#include <AFMotor.h>
#include <NewPing.h>
#include<Servo.h>
#define TRIGGER_PIN A1
#define ECHO_PIN A0
#define MAX_DISTANCE 300
#define IR A5

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);


NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

Servo myservo;

String voice;

void setup() {
Serial.begin(9600);
myservo.attach(10);
myservo.write(90);
pinMode(IR, INPUT);

}

void loop() {
```

Fig 5.18. Uploading sketch to Arduino IDE

Step 9. Attaching the Batter and its Holder and to open the Voice control App

We have to attach the battery holder to the sheet with the help of the double side tape and then put the batteries on it and then connecting the holder wires to the shield. And then at last we have to download the voice control app and then we have to connect with the Bluetooth module and we are ready to send the instructions to the car.

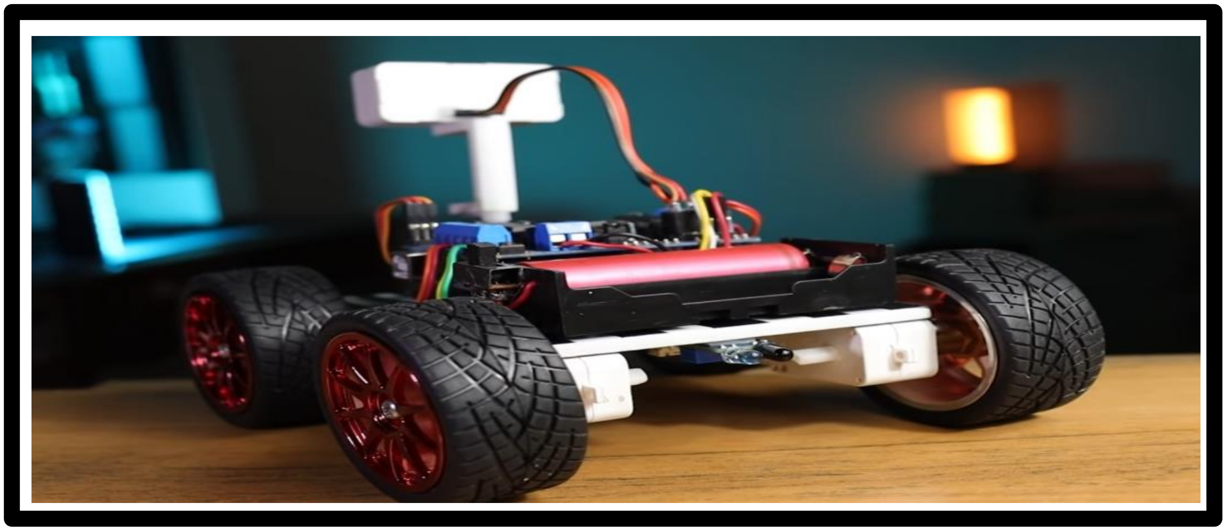


Fig 5.19. Attaching the Battery Holder

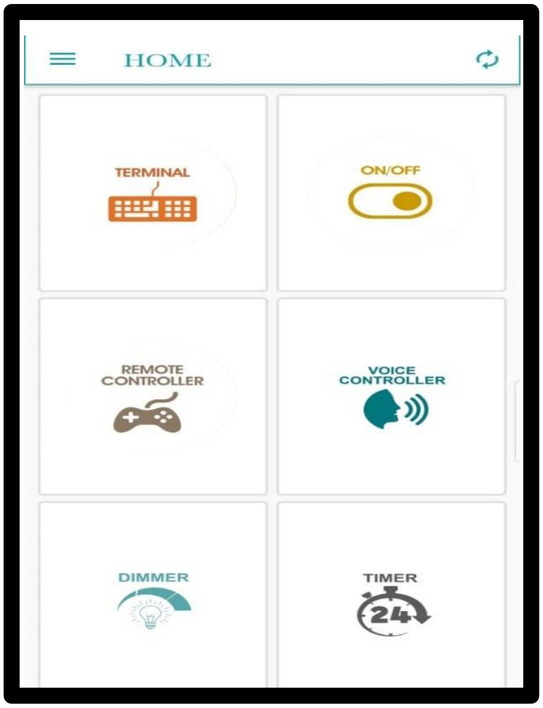
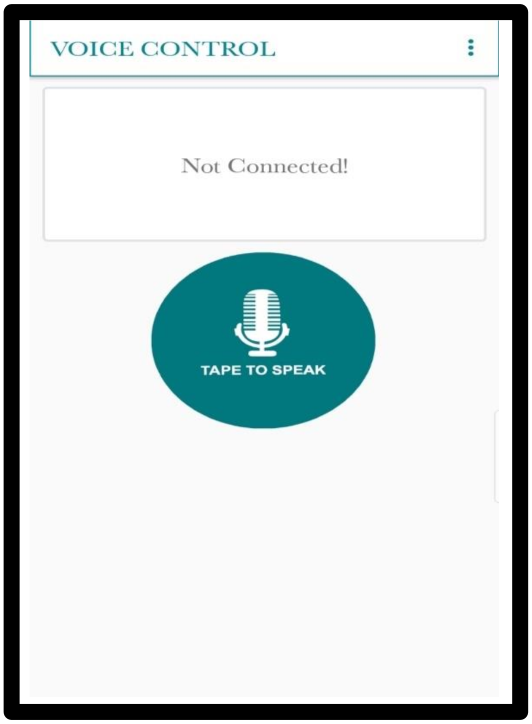


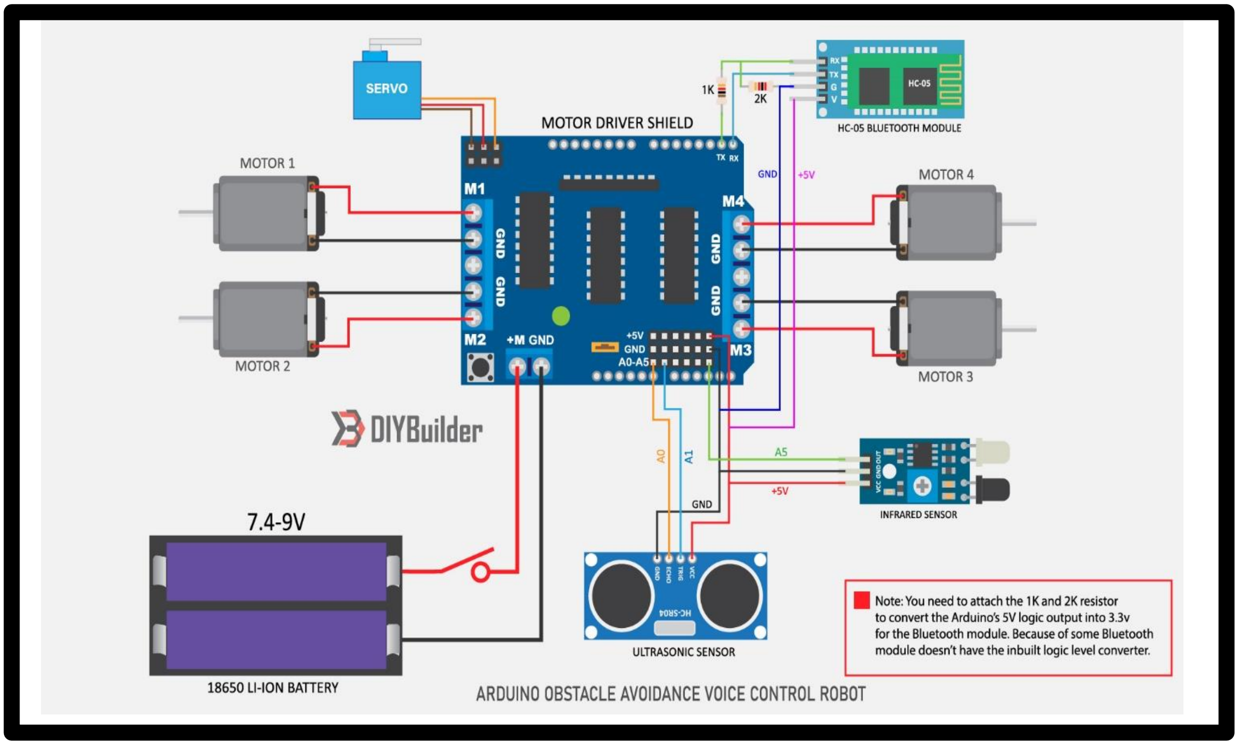
Fig 5.20. Voice Control App



Not Connected!

Fig 5.21. Voice Control App

Block Diagram



Note: You need to attach the 1K and 2K resistor to convert the Arduino's 5V logic output into 3.3v for the Bluetooth module. Because of some Bluetooth module doesn't have the inbuilt logic level converter.

ARDUINO OBSTACLE AVOIDANCE VOICE CONTROL ROBOT

Fig 5.22. Block Diagram

Code

```
// Arduino Obstacle Avoidance + Voice Control Robot

// Created by DIY Builder

//Contact me here https://www.instagram.com/diy.builder/

//You need to install the AFMotor and NewPing Libraries before uplodaing the sketch

#include <AFMotor.h>

#include <NewPing.h>
```

```
#include<Servo.h>

#define TRIGGER_PIN A1

#define ECHO_PIN A0

#define MAX_DISTANCE 300

#define IR A5

AF_DCMotor motor1(1, MOTOR12_1KHZ);

AF_DCMotor motor2(2, MOTOR12_1KHZ);

AF_DCMotor motor3(3, MOTOR34_1KHZ);

AF_DCMotor motor4(4, MOTOR34_1KHZ);

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

Servo myservo;

String voice;

void setup() {

Serial.begin(9600);

myservo.attach(10);

myservo.write(90);
pinMode(IR, INPUT);
}

void loop() {
  int distance = sonar.ping_cm();
  //int IR1 = digitalRead(IR);
  //Serial.println(IR1);
if(Serial.available()>0) {
  voice="";
  delay(2);
  voice = Serial.readString();
  delay(2);
  Serial.println(voice);
if (voice == "turn left") {
  left();
}else if (voice == "left") {
  left();
}else if(voice == "turn right") {
  right();
}else if(voice == "right") {
  right();
}
}
```

```
while(voice == "move forward") {
  forward();
}
while(voice == "move backward") {
  backward();
}
}
void forward() {
  int distance = sonar.ping_cm();
 if(distance < 10){
  Stop();
  voice="";
 }else {
  motor1.setSpeed(255);
  motor1.run(FORWARD);
  motor2.setSpeed(255);
  motor2.run(FORWARD);
  motor3.setSpeed(255);
  motor3.run(FORWARD);
  motor4.setSpeed(255);
  motor4.run(FORWARD);
}
}
void backward() {
  int IR_Sensor = digitalRead(IR);
  if(IR_Sensor == 0) {
    Stop();
    voice="";
  }else {
  motor1.setSpeed(255);
  motor1.run(BACKWARD);
  motor2.setSpeed(255);
  motor2.run(BACKWARD);
  motor3.setSpeed(255);
  motor3.run(BACKWARD);
  motor4.setSpeed(255);
  motor4.run(BACKWARD);
}
```

```
}

void left() {

  myservo.write(180);

  delay(500);

  myservo.write(90);

  delay(500);

  motor1.run(BACKWARD);

  motor1.setSpeed(255);

  motor2.run(BACKWARD);

  motor2.setSpeed(255);

  motor3.run(FORWARD);

  motor3.setSpeed(255);

  motor4.run(FORWARD);

  motor4.setSpeed(255);

  delay(700);

  motor1.run(RELEASE);

  motor2.run(RELEASE);

  motor3.run(RELEASE);

  motor4.run(RELEASE);
```

```
}
void right() {
  myservo.write(0);
  delay(500);
  myservo.write(90);
  delay(500);
  motor1.run(FORWARD);
  motor1.setSpeed(255);
  motor2.run(FORWARD);
  motor2.setSpeed(255);
  motor3.run(BACKWARD);
  motor3.setSpeed(255);
  motor4.run(BACKWARD);
  motor4.setSpeed(255);
  delay(700);
  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
```

```
}

void Stop() {

 motor1.run(RELEASE);

  motor2.run(RELEASE);

  motor3.run(RELEASE);

  motor4.run(RELEASE);

}
```

Chap 6. RESULTS AND APPLICATIONS

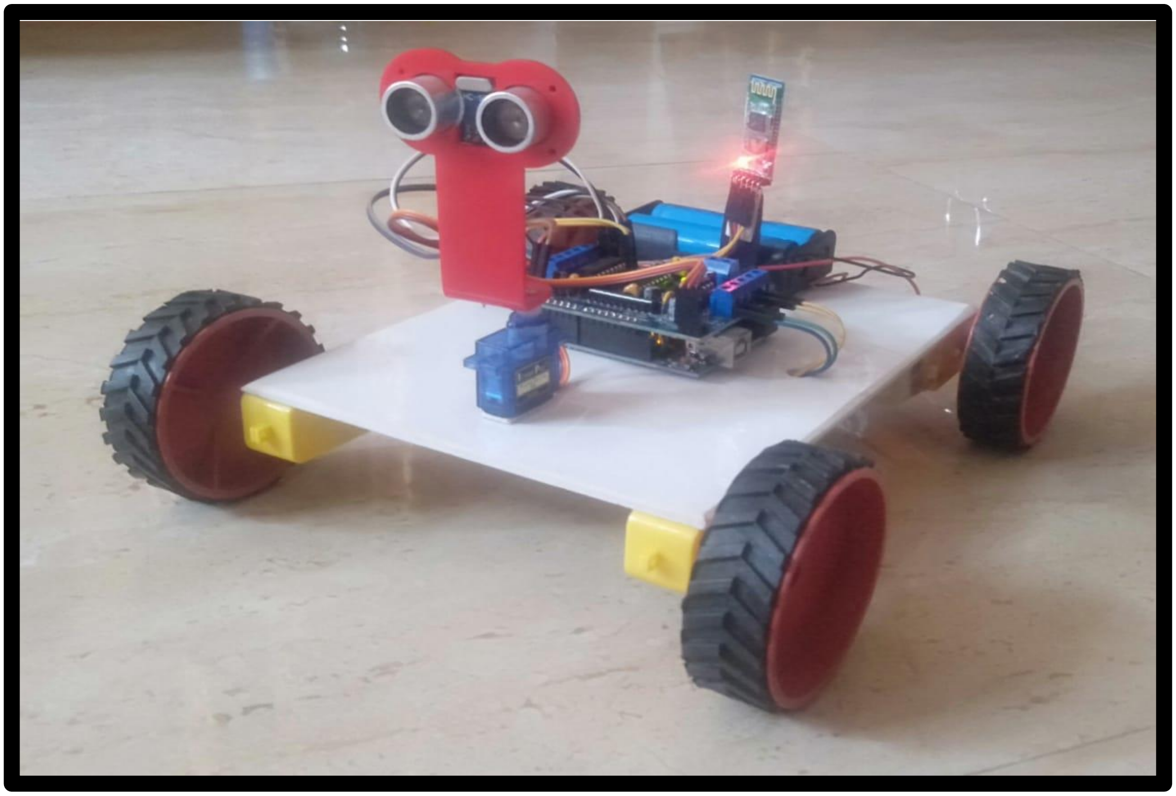


Fig. 6.1 Output

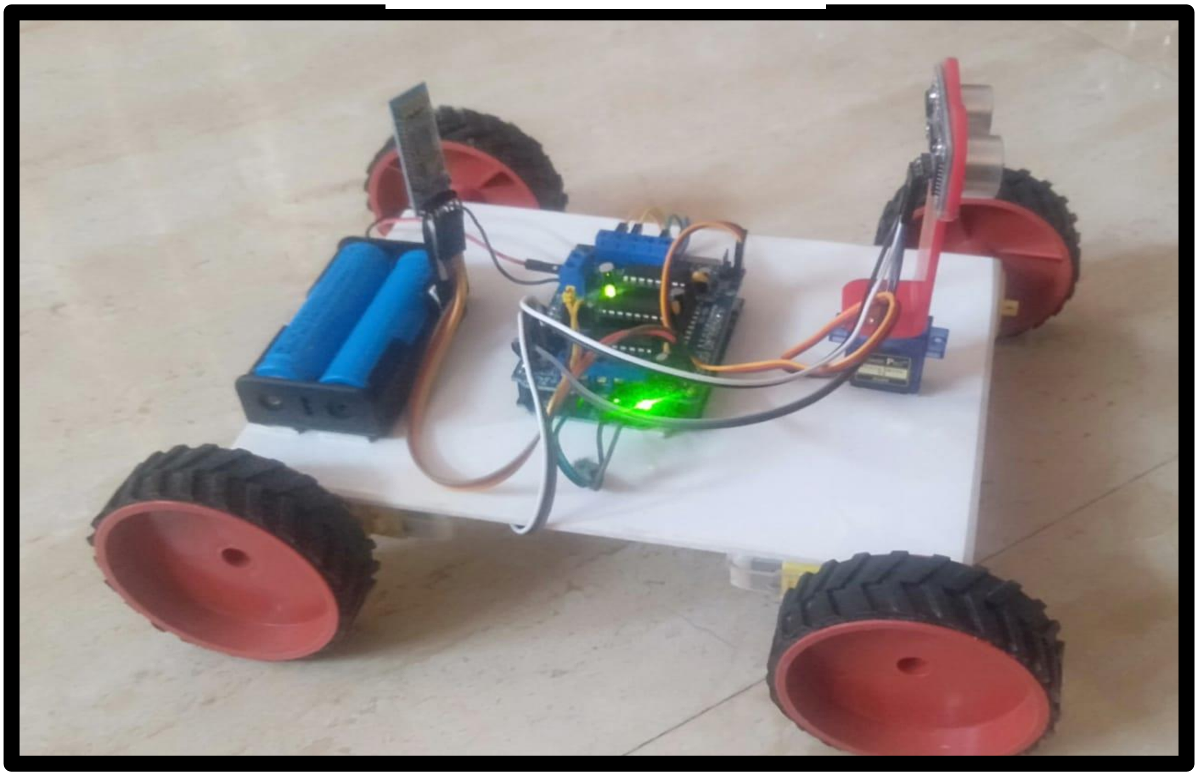


Fig. 6.2 Output

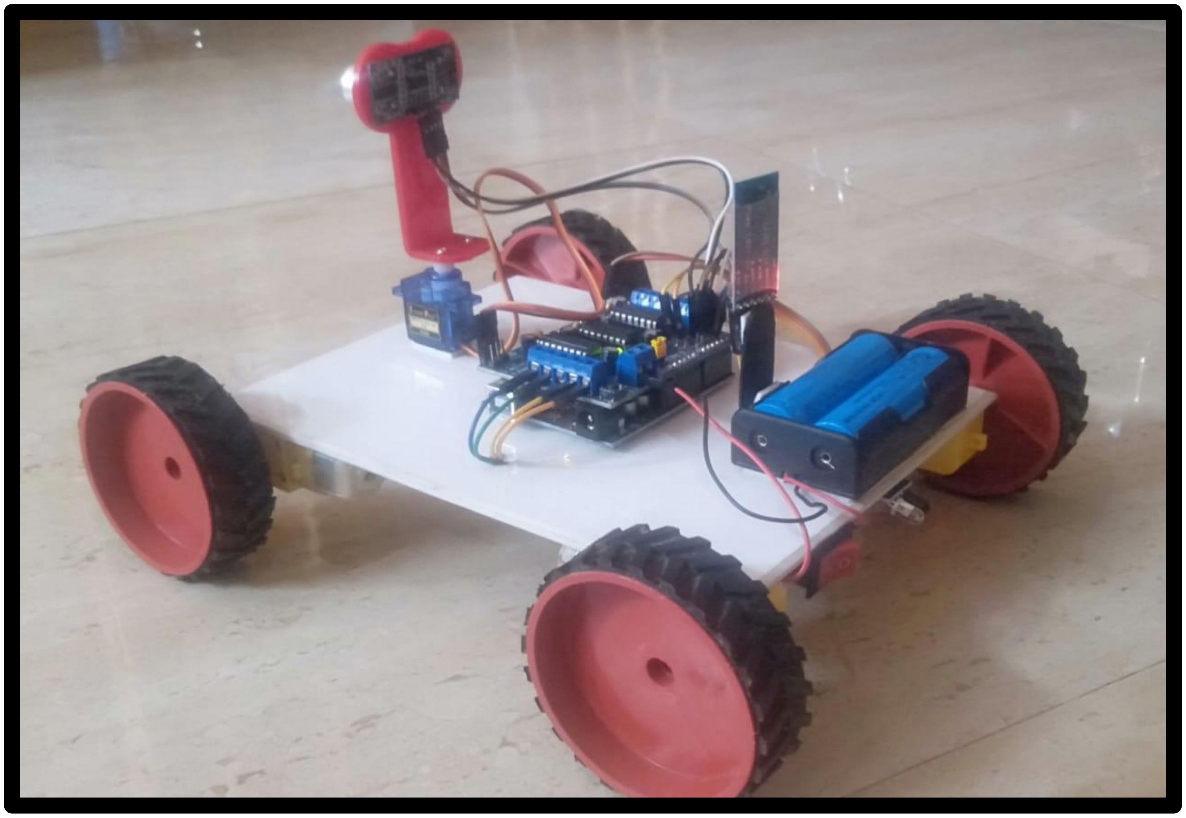


Fig. 6.3 Output

Applications:-

1. The car is useful in places where humans find difficult to reach but human voice reaches. E.g, in a small pipeline, in a fire-situations, in highly toxic areas.

2. The robot can be used as a toy.

3. It can be used to bring and place small objects.

4. It is the one of the important stages of Humanoid robots.

5. Command and control of appliances and equipment.

Chap 7. CONCLUSION AND FUTURE SCOPE

The “Arduino Voice Controlled Car” project has many applications and in present and future. The project can be made more effective by adding features to it in the future. The project has applications in wide variety of areas such as military, home security, rescue missions, industries, medical assistance etc. We were successful in implementing a simple model of voice controlled robotic vehicle using the available resources.

The implementation of this project is easy, so this car is beneficial for human life. The Voice Control Car is useful for disable people and monitoring purpose. It works on simple voice command, so it is easy to use. It is useful for those areas where humans can’t reach. The size of this car is small, so we can use this car for spying purpose. It can be used for surveillance. We can implement web cam in this car for security purpose. The voice recognition software has an accuracy and for identify a voice command and it is also highly sensitive to the surrounding noise.

The order of speech signals is automatically transmitted via a wired network to the server. The car is built primarily on a platform based on a microcontroller. Evaluation of the output of the original tests is carried out with promising implications.

Possible developments to feasible technologies in households, schools, vehicle networks and businesses are also addressed. Several areas that may additionally be discussed are the impact of noise on speech to textual content translation. The accent of the speaker no longer affects the robotic activity because speech commands are interpreted using a cloud-based server that works independently of the speaker accent.

The use of renewable energy sources for robotic operation would not only increase the value of robotic energy, but would also be environmentally friendly. Solar cells can be a suitable power source to use. The design of the robotic assistant is ideal for applications ranging from chemical manufacturing to comfortable home circumstances. Accuracy of detecting a voice command correctly is found to be 75%.

Chap 8. REFERENCES AND BIBLIOGRAPHY

1. https://youtu.be/1lrf50hBsHA
2. https://youtu.be/j7yltf_LtKQ
3. https://www.arduino.cc/en/main/arduinoBoardUno
4. https://wiki.seeedstudio.com/Motor_Shield_V2.0/
5. https://www.sgbotic.com/index.php?dispatch=products.view&product_id=2674#:~:text=These%20inexpensive%20and%20reliable%20TT,with%202%20x%20200mm
6. https://protosupplies.com/product/dc-motor-and-wheel-set/
7. https://circuitdigest.com/article/servo-motor-working-and-basics
8. https://www.maxbotix.com/articles/how-ultrasonic-sensors-work.htm#:~:text=Ultrasonic%20sensors%20work%20by%20sending,and%20to%20receive%20the%20echo
9. https://www.fenixlighting.com/blog/the-ultimate-guide-to-the-18650-battery/
10. https://en.m.wikipedia.org/wiki/Jump_wire
11. https://www.iqsdirectory.com/articles/power-supply/dc-power-supply.html#:~:text=A%20DC%20power%20supply%20is,a%20%22bench%20power%20supply.%22
12. https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-
13. https://www.seeedstudio.com/blog/2019/12/19/all-about-proximity-sensors-which-type-to-use/