An Industry Oriented Mini Project Report On

GESTURE CONTROLLED ROBOTIC CAR

Submitted In Partial Fulfillment of The Requirements
For The Award of The Degree of

Bachelor of Technology
In
Electronics & Communication Engineering
By

Valpadasu Sai Sudhamsh (HT. NO: 18831A04B2)

Under the guidance of
Mr. S. Sivaiah
Asst. Professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

GURU NANAK INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTUH) Campus: Ibrahimpatnam, R.R. District – 501506 (T.S.) Ph: (0/95) 8414-202120/21 Fax: 08414-223344 City Office: B2, 2 nd floor, Above Bata, Vikrampuri Colony, Secunderabad – 500009, Ph: 040 – 66323294, 65176117 Fax: 040 – 27892633)

2021-2022

Guru Nanak Institute of Technology

(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTU Hyderabad)Campus: Ibrahimpatnam, R.R. District – 501506 (T.S.) Ph: (0/95) 8414-202120/21 Fax: 08414-223344 City Office: B2, 2 nd floor, Above Bata, Vikrampuri Colony, Secunderabad – 500009, Ph: 040 – 66323294, 65176117 Fax: 040 – 27892633

CERTIFICATE

This is to certify that the Project report entitled "***GESTURE CONTROLLED ROBOTIC CAR***" is being submitted by ***VALPADASU SAI SUDHAMSH (18831A04B2)*** in partial fulfillment for the award of the Degree of Bachelor of Technology in ***Electronics And Communication Engineering*** of Jawaharlal Nehru Technological University Hyderabad during the year 2021-2022. The Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for Bachelor Degree.

Internal Guide

Mr. S. SIVAIAH
Assistant Professor
Department of ECE

Project Coordinator

Mr. G. MADHU
Assistant Professor
Department of ECE

Head of the Department

Dr. S. P. YADAV
Department of ECE

Principal

Dr. S. SREENATHA REDDY
GNIT

External Examiner

GNIT Guru Nanak Institute of Technology
(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTU Hyderabad)Campus: Ibrahimpatnam, R.R. District – 501506 (T.S.) Ph: (0/95) 8414-202120/21 Fax: 08414-223344 City Office: B2, 2 nd floor, Above Bata, Vikrampuri Colony, Secunderabad – 500009, Ph: 040 – 66323294, 65176117 Fax: 040 – 27892633

DECLARATION

We, hereby certify that the project work which is being presented in the thesis entitled ***"GESTURE CONTROLLED ROBOTIC CAR"*** is being submitted by ***V.SAI SUDHAMSH (18832A04B2)*** in partial fulfillment of requirements for the award of degree of ***B.Tech. (Electronics And Communication Engineering)*** submitted in the ***Department of ECE*** is an authentic record of our own work carried out during a period from ***September 2021 to January 2022.*** The matter presented in this thesis has not been submitted by us in any other University/ Institute for the award of any degree.

Signature of the Students

V.SAI SUDHAMSH

This is certifying that the above statement made by the candidates is correct to best of my knowledge.

Signature of the Internal Guide

Mr. S. SIVAIAH

The B.Tech Viva-Voice Examination of **Mr. VALPADASU SAI SUDHAMSH** has been held on _____ and it is accepted.

Signature of HOD **Signature of External Examiner**

ACKNOWLEDGEMENT

The project entitled ***"IOT BASED GESTURE CONTROLLED ROBOTIC CAR "*** has seen the light of the day because the success of this project is attributed to the intensity, drive and technical competence of many individuals who contributed to make this project a reality despite all odds. So, we express heartfelt pleasure for the people who helped us in this project and gave a never ending support right from the stage the idea was conceived.

We deeply express our sincere thanks to internal guide **Mr. S.SIVAIAH** whose initiation to bring up this project

We express our sincere thanks to **Mr. G. MADHU**, Project Coordinator and my entire faculty, without their constant encouragement and everlasting support this work would not have been possible.

We are very much thankful to **Dr. S. P. YADAV**, Head of ECE Department for his excellent guidance and constant encouragement throughout the project work. His timely advices and endless debates ensured the completion of project report in time.

We cordially thanks to **Dr. SANJEEV SRIVATHSAVA**, Dean Academics, R&D for his valuable guidance.

We would like to express my profound sense of gratitude to **Dr. S. SREENATHA REDDY**, Principal for his constant and valuable guidance.

We are deeply indebted to Honorable **Dr. H. S. SAINI**, Managing Director of Guru Nanak Institutions, for constant Cooperation during the course of work.

Last but not least, we would like to thank all teaching and Non-teaching staff members of ECE department for their support and also, we would like to thank God for all good deeds.

In All Sincerity,

Mr. VALPADASU SAI SUDHAMSH(18831A04B2)

GNIT Guru Nanak Institute of Technology

(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTU Hyderabad)

VISION AND MISSION OF INSTITUTE:

THE VISION OF THE INSTITUTION:
To be a World-class Educational and Research Institution in the service of humanity by promoting high quality Engineering and Management education.

THE MISSION OF INSTITUTION:
M1: Imbibe soft skills and technical skills
M2: Develop the faculty to reach international standards.
M3: Maintain high academic standards and teaching quality that promote analytical thinking and Independent judgement.
M4: Promote research, innovation and product development in collaboration with reputable Foreign Universities.
M5: Offer collaborative industry program in the emerging areas and spirit of enterprise.

QUALITY POLICY:
GNIT is committed to provide quality education through dedicated and talented faculty, world class infrastructure, labs and advanced research center to the students.

THE VISION OF THE DEPARTMENT

The Vision of the Electronics and Communication Engineering department is: To be recognized as a leading Electronics & Communication Engineering department in the region by students and employers and be known for leadership and commitment to foster quality teaching, learning, research and innovation. The Mission of the Electronics and Communication Engineering department is:

THE MISSION OF THE DEPARTMENT

1. Nurture young individuals into knowledgeable, skillful and ethical professionals in their pursuit of Electronics and Communication Engineering.
2. Nurture the faculty to expose them to world-class infrastructure.
3. Sustain high performance by excellence in teaching, research and innovations.
4. Extensive partnerships and collaborations with foreign universities for technology up gradation.
5. Develop Industry-Interaction for innovation and product development

GNIT Guru Nanak Institute of Technology

(Approved by AICTE, New Delhi, NBA Accredited ECE, ME, CSE & Affiliated to JNTU Hyderabad)

PROGRAM EDUCATIONAL OBJECTIVES (PEO's):

PEO-1: Graduates shall have the ability to apply knowledge and technical skills in emerging areas of Electronics and Communication Engineering for higher studies, research, employability, product development and handle realistic problems.

PEO-2: Graduates shall possess managerial skills, maintain ethical conduct, sense of responsibility to serve the society and protect the environment.

PEO-3: Graduates shall possess academic excellence with innovative insight, soft skills, leadership qualities, knowledge of contemporary issues for successful professional career.

PROGRAM OUTCOMES (PO's):

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. Department of ECE,GNIT.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics natural science, and engineering sciences.
3. Design / development of solutions: Design solutions for complex engineering problems and design system components or process that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, society and environmental consideration
4. Conduct Investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethics principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO's) ECE DEPARTMENT:

After completion of the program the graduate will be able to:

1. Apply the fundamental knowledge and skills of Electronics to design circuits and develop the Integrated Circuits for various applications in the field of communication, signal processing and in system automation to address National/Global issues
2. Product development by using cutting edge software and hardware tools of Electronics and communication engineering.

INDEX

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Gesture Controlled Robot is a car which can be controlled by simple human gestures. The user just needs to wear a hand glove which consists of a gesture device in which a accelerometer sensor is included. The sensor will record the movement of hand in a specific direction which will result in the motion of the robot in the respective directions. The sensor is intended to replace the remote control that is generally used to run the car. The robot and the gesture instrument are connected wirelessly through radio waves so the user can interact with the robot in a more friendly way due to the wireless communication. We can control the car using accelerometer sensors connected to a hand glove. It will allow user to control the forward, backward, left and right movements, while using the same accelerometer sensor to control the throttle of the car. Gesture controlled robot works on the principle of accelerometer which records hand movements in X and Y directions and sends that data to the comparator which assigns proper voltage levels to the recorded movements. This recorded information is then passed to an encoder which makes it ready for RF transmission. On the receiving end, the information is received wirelessly via RF receiver, decoded and then passed onto the micro-controller which takes various decisions based on the received information. These decisions trigger the motors in different configurations to make the robot move in a specific direction.

CHAPTER 1
INTRODUCTION

Now-a-days, robotics are becoming one of the most advanced in the field of technology. A Robot is an electro-mechanical system that is operated by a computer program. Robots can be autonomous or semi-autonomous. An autonomous robot is not controlled by human and acts on its own decision by sensing its environment. Majority of the industrial robots are autonomous as they are required to operate at high speed and with great accuracy. But some applications require semi-autonomous or human controlled robots.Some of the most commonly used control systems are voice recognition, tactile or touch controlled and motion controlled.

1.1 Use for the Society:

The traditional wired buttons controlled robot becomes very bulgy and it also limits the distance the robot goes as we use only a limited length of wire. But the Gesture control through wireless communication can provide intuitive and efficient solutions in robotics, with specific applications including unmanned ground vehicles for military surveillance and improved manufacturing safety and efficiency, robotic arm manipulation for prosthetic devices, robot-assisted surgeries, and robotic nursing-care assistants. The goal of this work is to assess the feasibility of a portable, low-cost, easy-to-use, and engaging robot device.

1.2 Existing Systems:

Most of the existing systems have the controlling unit connected to devices by the means of wire or an IR sensor which is causing many limitations. Some of the limitations are discussed below

- The range of communication is being limited.

- Very bulgy to carry.
- Infrared waves at high force can harm eyes.
- Lower information rate transmission contrast.
- The signal can be blocked by any material that is in front of the transmission. This could be things such as objects, walls and people. Also, the transmitters and receivers must be closely aligned to communicate by being directly in sight of each other.

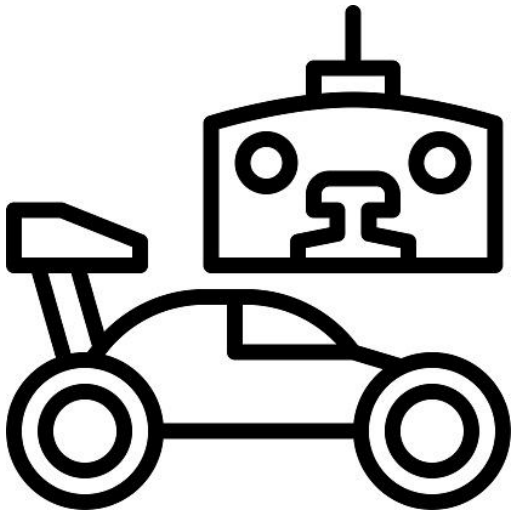


Fig 1.1: Wired Remote Control Car

1.3 Proposed System:

The basic idea of our project is to develop a system (robot) which can recognize the Human Interaction with it to accomplish the certain tasks assigned to it. In our project we will design a wearable Hand Glove which will contain the sensors mounted on it to capture the movement of the hand and convert the raw mechanical data into electrical form. This data will be further processed and converted into an understandable format for the Arduino Nano mounted on the Glove. This Arduino Nano with the transmitter module connected to it will act as a transmitter of the data for wireless communication purpose. Once the transmitted data is received by the receiver module which will be connected to the Micro-controller, it will be processed and further sent to the Micro-controller.

Micro-controller will decode the commands and accordingly it will actuate the motor drivers to control the Motors for various tasks on the robot.

Fig 1.2: Proposed System

CHAPTER 2

BLOCK DIAGRAM & EXPLANATION

2.1 Block Diagram:

The gesture controlled robotic car has two separate circuits which are connected wirelessly hence, the complete system can be explained with two different block diagrams (i.e. transmitter's end & receivers end).

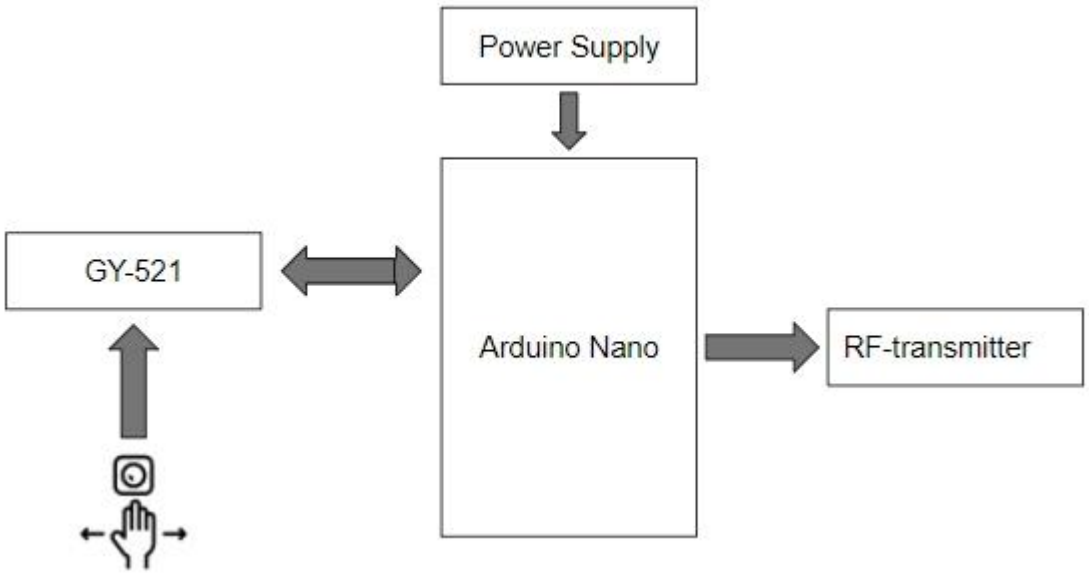


Fig 2.1: Transmitter's End Block Diagram

The above block diagram in figure 2.1 is the transmitter's end of the robot. It consists of a GY-521 accelerometer module, Arduino Nano, RF-transmitter module and a power supply to power-up the setup. This complete setup is built onto a wearable hand glove. The Hand Glove which is containing the sensor mounted on it to capture the movement of the hand will capture the hand movements and convert the raw mechanical data into electrical form. This data will be further processed and converted into an understandable format for the Arduino mounted on the Glove. This data is sent onto to the

encoder present in the RF transmitter. This encoded data is then transmitted into the environment in the form of radio frequency waves. It's then received by the receiver.

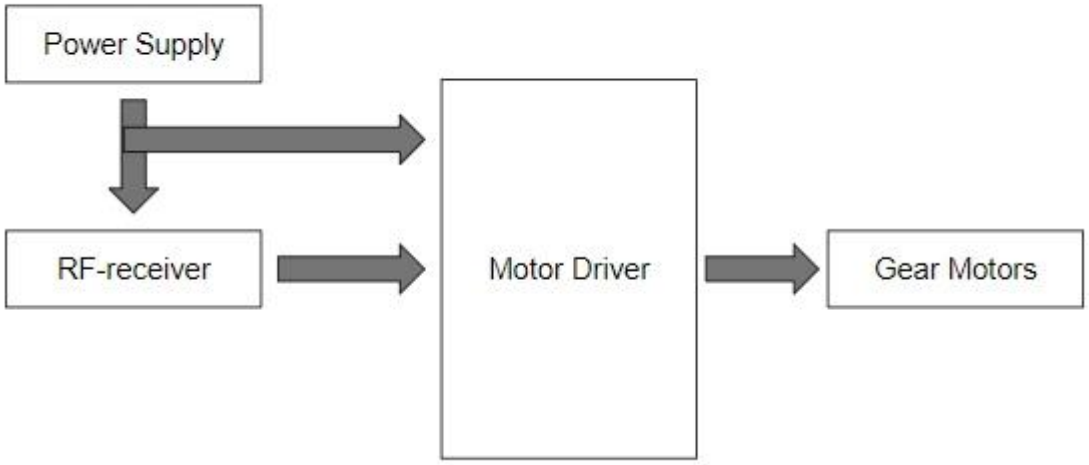


Fig 2.2: Receiver's End Block Diagram

The Receiver's end of the robot consists of a RF-receiver module, motor driver and gear motors. The received data from the transmitter's end is sent to the decoder IC present in RF-receiver module. This data is decoded and sent to the motor driver module. The micro-controller present on the motor driver circuit take various decisions based on the received information which in turn aides in the driving of the DC gear motors.

2.2 Methodology:

Methodology for communication signal

Transmitter Module:

An RF transmitter module is a small PCB ie, printed circuit board sub-assembly capable of transmitting a radio wave and modulating that wave to carry data. Transmitter modules are usually implemented alongside a micro controller which will provide data to the module which is transmitted. RF transmitters are usually subject to regulatory requirements which

dictate the maximum allowable transmitter power output, harmonics and band edge requirement.

Receiver modules:

An RF Receiver module RF433-RX is 433 MHz radio receiver receives the modulated RF signal, and then it demodulates. There are two types of RF receiver module. Super-regenerative modules are usually of low cost and low power designs using a series of amplifiers use to extract modulated data from a carrier wave. Super-regenerative modules are generally imprecise as their frequency of operation varies in a fair amount with temperature and power supply voltage. Super hetero-dyne receivers having a performance advantage over super-regenerative; they offer increased an accuracy and stability over a large voltage and temperature range. This stability comes from a fixed crystal design which in turn leads to a comparatively more expensive product. Radio receiver which receives the transmitted coded from the remote place these codes are converted to digital format and output is available to the pin no 2 of the IC2 master micro-controller; this is the pin of inbuilt art of the micro-controller. We Based on the input codes master will give command to slave micro-controller and robot will behave as follows.

- Moves in forward direction .·
- Moves in reverse direction.
- It can even turn left or right while moving forward or in reverse direction.
- In case of bump, moves reverse turn left or right and wail for the next instruction.
- On the spot left or right turn to pass through the narrow space.

Methodology for Motion Control

This robot is radio operated which is, self powered, and has all the controls like a normal car. The control of the car is based on the hand movements of the user which is collected and by the GY-521 Module and transmitted by the transmitter module. This is the received by the receiver on the car and sent to the motor driver for appropriate movement.

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers as they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors. L293D contains two inbuilt H-bridge driver circuits. In common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state. This project controls a remote robot through RF. The ordinary 433 MHz RF modules are used in this project.

Fig 2.3 Hand Movements

CHAPTER 3

HARDWARE COMPONENTS

3.1 POWER SUPPLY:

The power supply section is the section which provide +5V for the components to work. IC LM7805 is used for providing a constant power of +5V. The ac voltage, typically 220V, is connected to a transformer, which steps down that ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit removes the ripples and also retains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

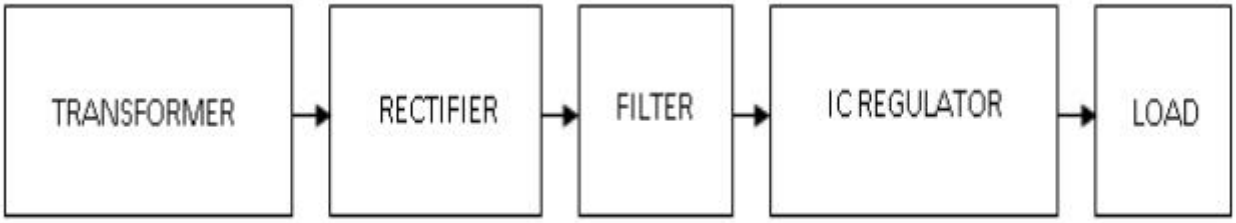


Fig 3.1 Block Diagram of Power Supply

Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC. Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in India) to a safer low voltage. The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the

transformer. Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up. The transformer will step down the power supply voltage (0-230V) to (0- 6V) level. Then the secondary of the potential transformer will be connected to the bridge rectifier, which is constructed with the help of PN junction diodes. The advantages of using bridge rectifier are it will give peak voltage output as DC.

Rectifier:

There are several ways of connecting diodes to make a rectifier to convert AC to DC. The bridge rectifier is the most important and it produces full-wave varying DC. A full-wave rectifier can also be made from just two diodes if a centre-tap transformer is used, but this method is rarely used now that diodes are cheaper. A single diode can be used as a rectifier but it only uses the positive (+) parts of the AC wave to produce half-wave varying DC.

Bridge Rectifier:

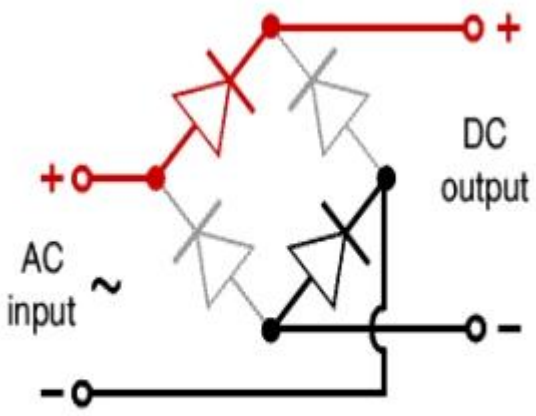


Fig 3.2 Bridge Rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners. Let us assume that the transformer

is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4. The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow. One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit.

i. The main advantage of this bridge circuit is that it does not require a special centre tapped transformer, thereby reducing its size and cost.

ii. The single secondary winding is connected to one side of the diode bridge network and the load to the other side as shown below.

iii. The result is still a pulsating direct current but with double the frequency.

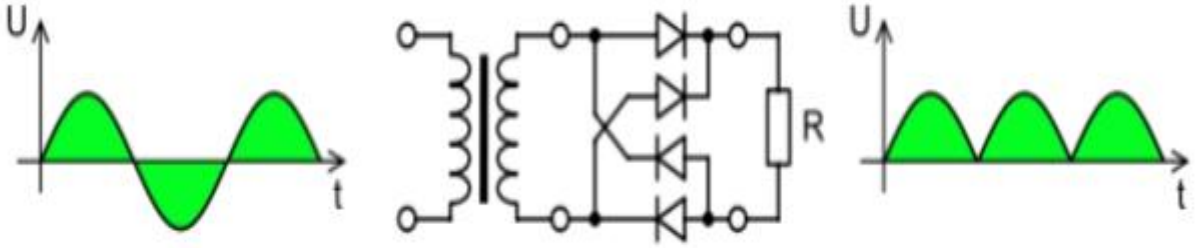


Fig 3.3 Output Waveform of DC

Voltage Regulator:

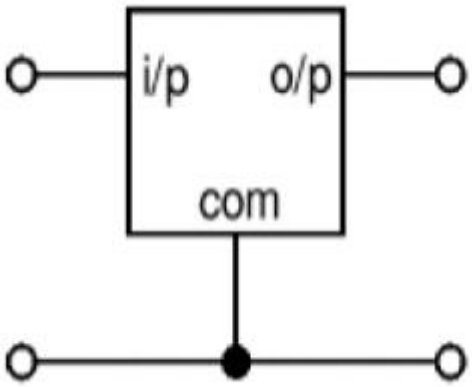


Fig 3.4 Regulator

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustable set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to Tens of watts.

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

Many of the fixed voltage regulator IC's has 3 leads and look like power transistors, such as the 7805 +5V 1Amp regulator. They include a hole for attaching a heat sink if necessary.

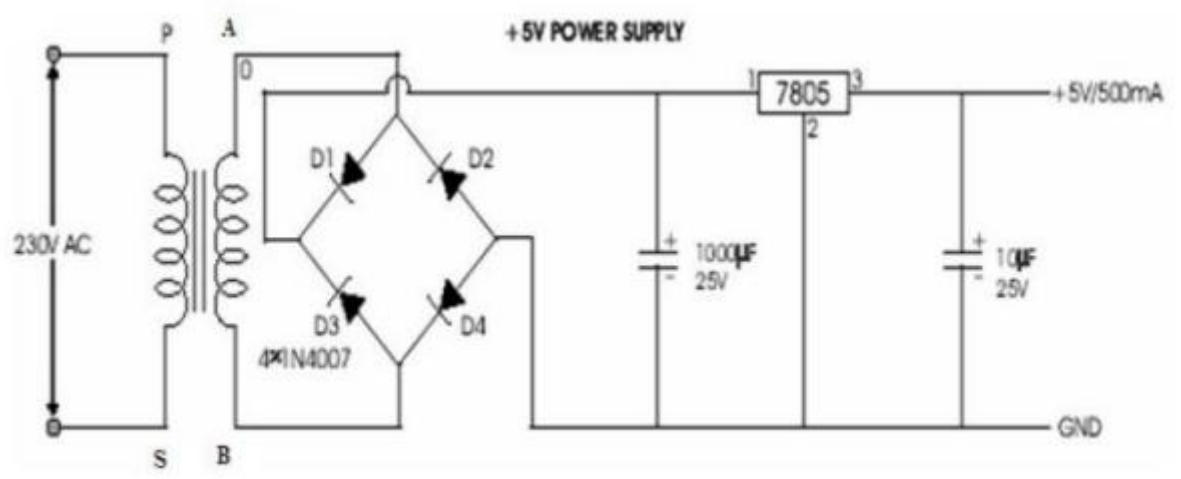


Fig 3.5 Circuit Diagram of Power Supply

3.2 MICRO-CONTROLLER:

A Micro-controller (or MCU) is a computer-on-a-chip used to control electronic devices. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general purpose microprocessor (the kind used in a PC). A typical micro-controller contains all the memory and interfaces needed for a simple application, whereas a general purpose microprocessor requires additional chips to provide these functions.

A micro-controller is a single integrated circuit with the following key features:

- Central Processing Unit - ranging from small and simple 8-bit processors to sophisticated 32- or 64- bit processors
- Input/output interfaces such as serial ports
- RAM for data storage
- ROM, EEPROM or Flash memory for program storage
- Clock generator - often an oscillator for a quartz timing crystal, resonator or RC circuit

Micro-controllers are inside many kinds of electronic equipment (see embedded system). They are the vast majority of all processor chips sold. Over 50% are "simple" controllers, and another 20% are more specialized digital signal processors.

3.2.1 Arduino Nano

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other micro-controllers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LED's on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

Fig 3.6: Arduino Nano Board

Automatic (Software) Reset:

Rather than requiring an actual press of the reset button before a transfer, the Arduino Nano is planned in a way that permits it to be reset by programming running on an associated PC. One of the equipment stream control lines (DTR) of the FT232RL is associated with the reset line of the ATmega328 through a 100 nano farad capacitor. At the point when this line is affirmed (taken low), the reset line drops to the point of resetting the chip.

This arrangement has different implications. At the point when the Nano is connected with a PC running Mac OS X or Linux, it resets each time an association is made to it from programming (through USB). For the accompanying half-second or thereabouts, the boot loader is running on the Nano.

3.2.2 Arduino Nano Technical Specifications:

Micro-controller	ATmega328
Architecture	AVR
Operating Voltage	5V
Flash Memory	32 KB of which 2KB is used by boot loader
SRAM	2 KB
Clock Speed	16 MHz
Analog I/O Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA
Input Voltage	7-12V
Digital I/O Pins	22
PWM Output	6
Power Consumption	19 mA

PCB Size	18 x 45 mm
Weight	7g

Table 3.2.2: Arduino Nano Technical Specifications

3.2.3 Arduino Nano Pin Diagram:

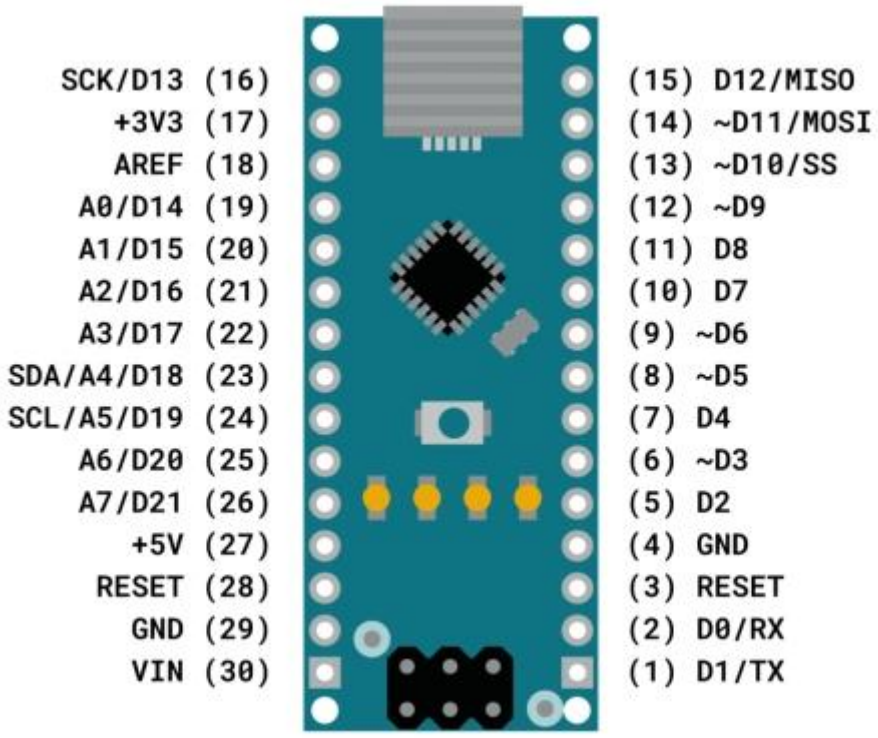


Fig 3.7: Arduino Nano Pin Diagram

3.2.4 Communication:

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other micro-controllers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is

being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus to use the SPI communication.

3.3 GY-521 ACCELEROMETER MODULE:

GY-521 Module is a very important module in this complete project. This is the device that is used to record the hand movements and give instructions to the Arduino Nano board for appropriate movement of the car.

The GY-521 module is a breakout board for the MPU-6050 MEMS (Microelectromechanical systems) that features a 3-axis gyroscope, a 3-axis accelerometer, a digital motion processor (DMP), and a temperature sensor. The digital motion processor can be used to process complex algorithms directly on the board. Usually, the DMP processes algorithms that turn the raw values from the sensors into stable position data. This tutorial gives only a brief introduction to the GY-521/MPU-6050. In particular, it is shown how to retrieve the raw sensor values. The sensor values are retrieved by using the I2C serial data bus, which requires only two wires (SCL and SDA).

Fig 3.8: GY-521 Module

3.3.1 Working:

This device is made by using poly-silicon surface sensor and signal conditioning circuit to measure acceleration. The output of this device is Analog in nature and proportional to the acceleration. This device measures the static acceleration of gravity when we tilt it. And gives an result in form of motion or vibration. According to the data sheet of adxl335 poly-silicon surface-micro machined structure placed on top of silicon wafer. Poly-silicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor which incorporate independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of acceleration.

3.3.2 Pin Diagram:

Pin Description of accelerometer

1. **VCC** 5 volt supply should connect at this pin .
2. **GND** Ground pin.
3. **SCL** - I2C clock pin, connect to your micro-controllers I2C clock line. This pin is level shifted so you can use 3-5V logic, and there's a 10K pull up on this pin.
4. **SDA** - I2C data pin, connect to your micro-controllers I2C data line. This pin is level shifted so you can use 3-5V logic, and there's a 10K pull up on this pin.
5. **XDA** - Auxiliary Serial Data, can be used to interface other I2C modules with MPU6050. It is optional.
6. **XCL** - Auxiliary Serial Clock, can be used to interface other I2C modules with MPU6050. It is optional.
7. **AD0** - If more than one MPU6050 is used a single MCU, then this pin can be used to vary the address.
8. **INT** - Interrupt pin to indicate that data is available for MCU to read.

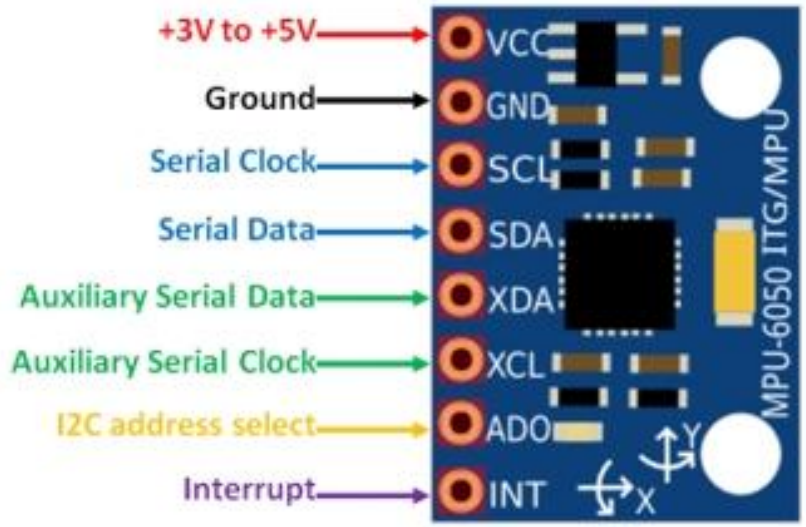


Fig 3.9: GY-521 Module Pin Diagram

3.3.3 Applications:

Some of the major applications of the accelerometer module are discussed below,

- Accelerometers can be used to measure vehicle acceleration. Accelerometers can be used to measure vibration on cars, machines, buildings, process control systems and safety installations.
- Accelerometers are also increasingly used in the biological sciences.
- Accelerometers are also used for machinery health monitoring to report the vibration and its changes in time of shafts at the bearings of rotating equipment such as turbines, pumps, fans, rollers etc.
- Vehicle collisions
- Impact loads – falling debris
- Concussion loads – internal and external explosions
- Collapse of structural elements
- Wind loads and wind gusts
- Air blast pressure
- Earthquakes and aftershocks

3.4 RADIO FREQUENCY MODULE:

As the name suggests, RF module operates at Radio Frequency. This frequency range varies between 30 kHz & 300 GHz. In this RF system, the digital data is represented as variations in the amplitude of carrier wave. This types of modulation is an Amplitude Shift Keying (ASK) .

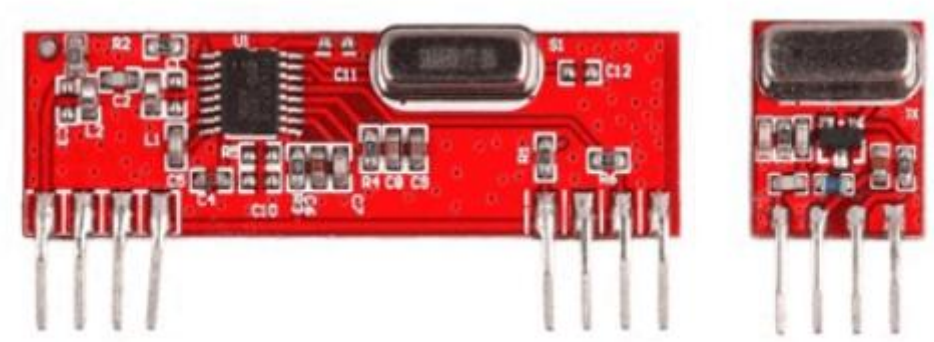


Fig 3.10: RF Module

This RF Module is a combination of RF Transmitter and RF Receiver. The transmitter/receiver (Tx/Rx) pair operates at a frequency of 433 MHz.

3.4.1 Working:

The RF transmitter receives serial data and transmits it wirelessly through through its RF antenna. The transmission occurs at the rate of 1 Kbps – 10 Kbps. RF receiver receives the transmitted data and it is operating at the same frequency as that of the transmitter.

Features of RF Module:

- The Receiver frequency 433MHz
- Receiver typical frequency 105 Dbm
- Receiver supply current 3.5 mA
- Low power consumption
- operating voltage of receiver is 5V
- The transmitter frequency range 433.92MHz

- Supply voltage of transmitter is between 3V to 6V
- Output power of transmitter is between 4Dbm to 12Dbm

3.4.2 Pin Diagram:

The Transmitter module consists of three pins namely Vcc, Din and ground as shown above. The Vcc pin has a wide range input voltage from 3V to 12V. The transmitter consumes a minimum current of 9mA and can go as high as 40mA during transmission. The center pin is the data pin to transmitter the signal. This signal modulated using the ASK and then sent on air at a frequency of 433MHz.

The RF Receiver module has four pins namely Vcc, Dout, Linear out and Ground as shown above. The Vcc pin should be powered with a regulated 5V supply. The operating current of this module is less than 5.5mA. The pins Dout and Linear out is shorted together to receive the 433Mhz signal from air. This signal is then demodulated to get the data and sent out through the data pin.

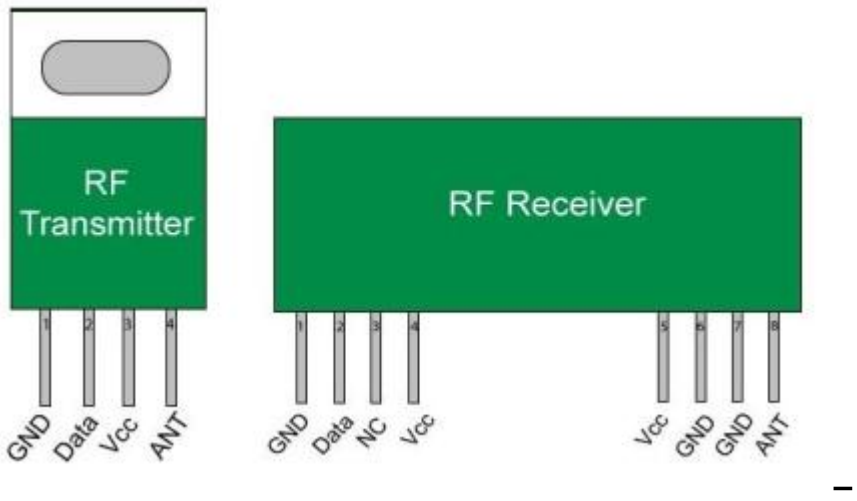


Fig 3.11: RF Module Pin Diagram

3.4.3 Applications:

- RF transceiver module is used in wireless communication. The main application of this transceiver is to make information in the form of data/voice/video apt to be transmitted over the wireless medium.

- The main intention of this device is to alter IF frequency to RF frequency and vice versa.
- RF transceiver module is used for radio transmission, satellite communication, television signal transmission, reception, and in Wimax or WLAN, Zigbee, or ITE networks.

3.5 L293 MOTOR DRIVER MODULE:

L293D is a basic motor driver integrated chip (IC) that enables us to drive a DC motor in either direction and also control the speed of the motor. The L293D is a 16 pin IC, with 8 pins on each side, allowing us to control the motor. It means that we can use a single L293D to run up to two DC motors. L293D consist of two H-bridge circuit. H-bridge is the simplest circuit for changing polarity across the load connected to it.

There are 2 OUTPUT pins, 2 INPUT pins, and 1 ENABLE pin for driving each motor. It is designed to drive inductive loads such as solenoids, relays, DC motors, and bipolar stepper motors, as well as other high-current/high-voltage loads.

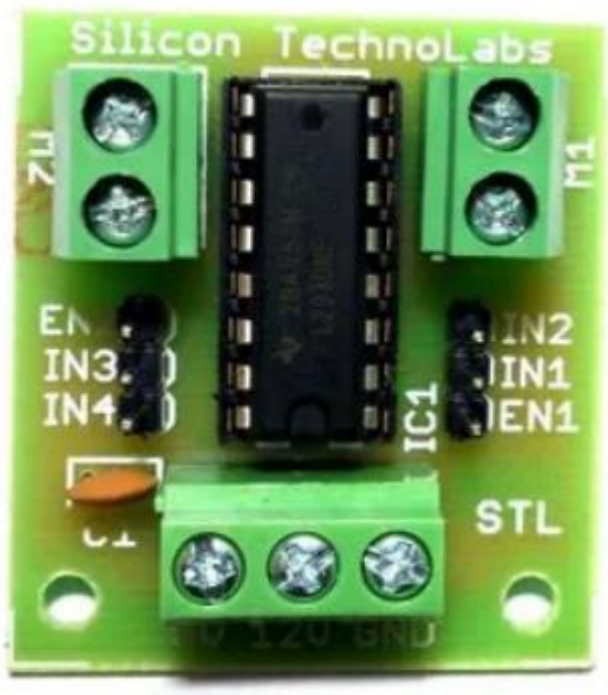


Fig 3.12: Motor Driver Module

3.5.1 Working:

There are 4 input pins for direction control in L293d. Pin 2,7 (1A and 2A) on the left side and pin 15,10 (3A and 4A) on the right of the IC. The left side input pins regulate the rotation of the motor connected across the left end and the right-side input pins regulate the motor on the right-side. The motors are rotated based on the inputs provided across the input pins as HIGH or LOW signals.

Let's take an example, a motor is connected on the left side output pins (pin 3,6). To control this motor, we have to provide an input logic to pin 2,7 (1A,2A).

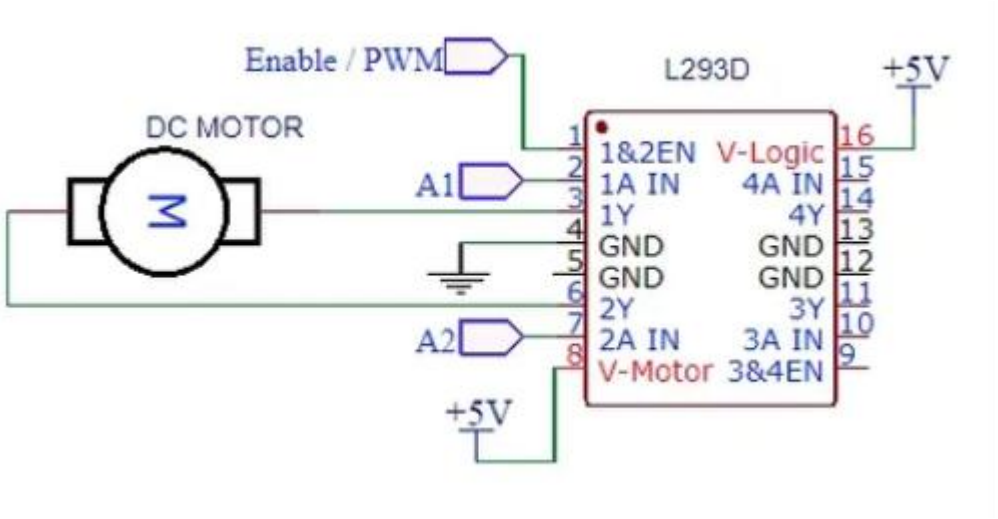


Fig 3.13 Motor Driver Module Example Connection

- Pin 2 = HIGH and Pin 7 = LOW | Clockwise Direction
- Pin 2 = LOW and Pin 7 = HIGH | Counter clockwise Direction
- Pin 2 = LOW and Pin 7 = LOW | Idle (No rotation)
- Pin 2 = HIGH and Pin 7 = HIGH | Idle (No rotation)

In a similar manner, we can control the motor on the right side connected to pin (11,14). For this, we need to provide HIGH and LOW input signal across pin (10,15).

- Pin 10 = HIGH and Pin 15 = LOW | Clockwise Direction
- Pin 10 = LOW and Pin 15 = HIGH | Counter clockwise Direction
- Pin 10 = LOW and Pin 15 = LOW | Idle (No rotation)

- Pin 10 = HIGH and Pin 15 = HIGH | Idle (No rotation)

3.5.2 Pin Diagram:

There are 2 OUTPUT pins, 2 INPUT pins, and 1 ENABLE pin for driving each motor. It is designed to drive inductive loads such as solenoids, relays, DC motors, and bipolar stepper motors, as well as other high-current/high-voltage loads.

Pin Diagram of L293D Motor Driver Module:

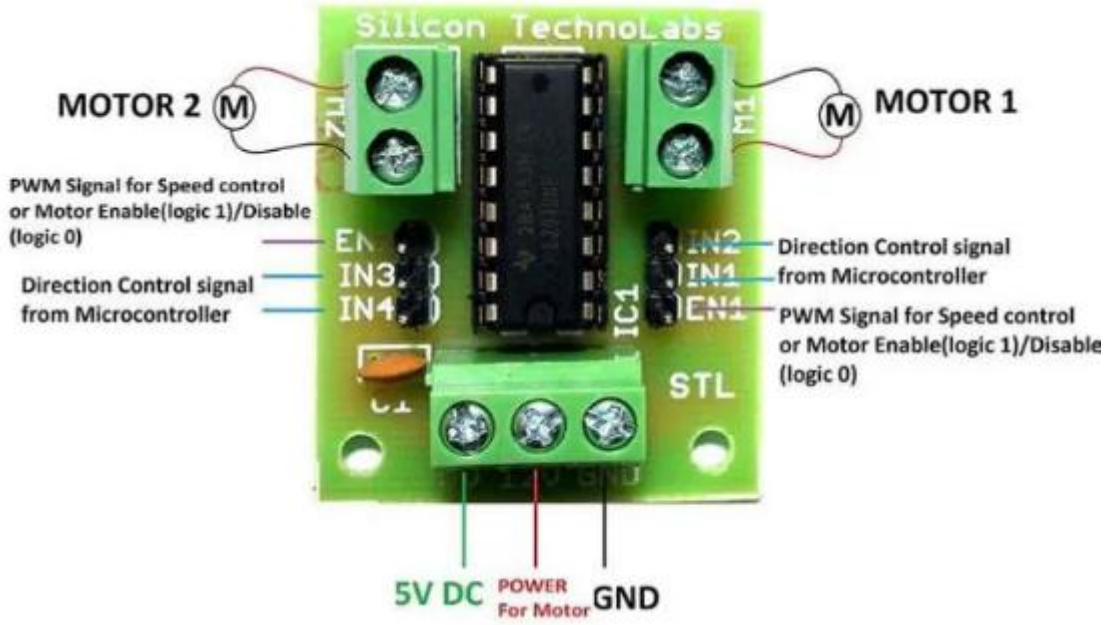


Fig 3.14: Pin Diagram of L293D Motor Driver Module

Pin Description:

The image given above shows the pin-outs of the L293D motor driver module. The function of each pin and port is also mentioned in the picture.

IN1, IN2, and IN3, IN4 are input pins used for providing a control signal from the controller to run the motor in different directions.

- If input logic at IN1, IN2 is (1,0) the motor rotates in one direction.
- If input logic at IN1, IN2 is (0,1) the motor rotates in the other direction.

EN1 and EN2 are enable pins. Connect 5v DC to EN1 and EN2 pin to operate the motor at its normal speed

- If speed control is needed, then give PWM output at pin EN1 and En2 from the micro-controller.

Power for the motor. If 12V DC gear motor is used then apply 12V.

Pin Diagram of L293D Motor Driver IC:

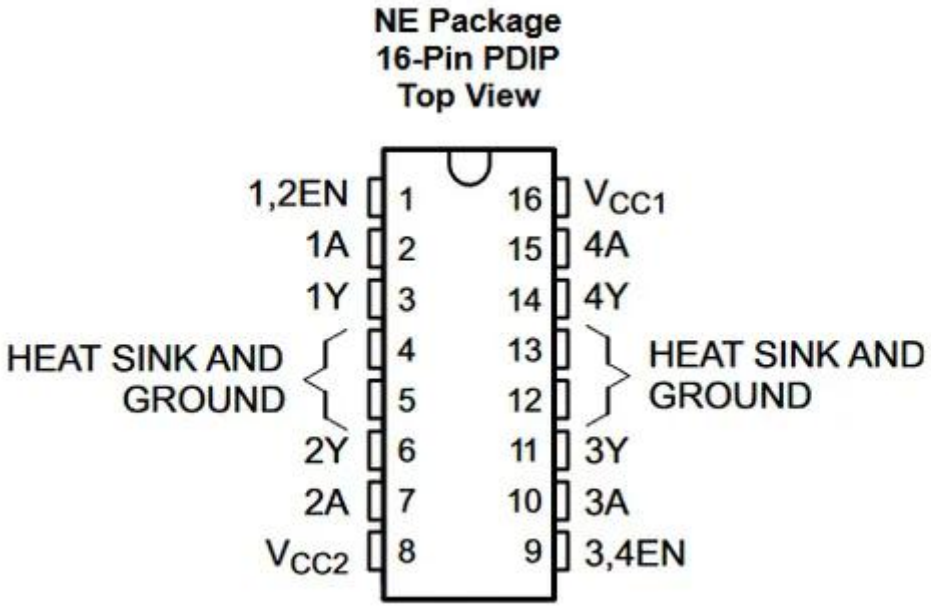


Fig 3.15: Pin Diagram of L293D IC

Pin Description:

Pin No.	Name	Function
1	Enable 1-2	When this pin is given HIGH or Logic 1, the left side of the IC works and when it is low, the left side doesn't work.

2	INPUT 1	When this pin is given HIGH or logic 1, the output 1 becomes HIGH.
3	OUTPUT 1	This pin is connected to one of the terminals of the motor 1.
4,5	GND	Should be connected to the circuit's ground.
6	OUTPUT 2	This pin is connected to one of the terminals of the motor 1.
7	INPUT 2	When this pin is given HIGH or Logic 1, the output 2 becomes HIGH.
8	VCC2	This is the voltage required to run the motor. IT can be greater than the IC voltage(VCC1).
16	VCC1	It provides power to the l293D IC. So, this pin should be supplied with 5 V.
15	INPUT 4	When this pin is given HIGH or logic 1, the output 4 becomes HIGH.
14	OUTPUT 4	This pin is connected to one of the terminals of the motor 2.
13,12	GND	Should be connected to the circuit's ground.

11	OUTPUT 3	This pin is connected to one of the terminals of the motor 2.
10	INPUT 3	When this pin is given HIGH or logic 1, the output 3 becomes HIGH.
9	Enable 3-4	When this pin is given HIGH or Logic 1, the right side of the IC works and when it is low, the right side doesn't work.

Table 3.5.2: L293D IC Pin Description

Specifications of L293D Motor Driver IC:

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Output Current 600 mA Per Channel
- Peak Output Current 1.2 A Per Channel
- Output Clamp Diodes for Inductive Transient Suppression
- Operation Temperature 0°C to 70°C.
- Automatic thermal shutdown is available

3.6 DC MOTOR:

A Direct Current (DC) motor is a rotating electrical device that converts direct current, of electrical energy, into mechanical energy. An Inductor (coil) inside the DC motor produces a magnetic field that creates rotary motion as DC voltage is applied to its terminal. Inside the motor is an iron shaft, wrapped in a coil of wire. This shaft contains two fixed, North and South, magnets on both sides which causes both a repulsive and attractive force, in turn, producing torque. ISL Products designs and manufactures both brushed DC motors and brushless DC motors.

A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gear head to a motor reduces the speed while increasing the torque output. The most important parameters in regards to gear motors are speed (rpm), torque (lb-in) and efficiency (%). In order to select the most suitable gear motor for your application you must first compute the load, speed and torque requirements for your application.

By far the most common DC motor types are the brushed and brushless types, which use internal and external commutation respectively to create an oscillating AC current from the DC source—so they are not purely DC machines in a strict sense.

We in our project are using brushed DC Motor, which will operate in the ratings of 12v DC 0.6A.

The speed of a DC motor can be controlled by changing the voltage applied to the armature or by changing the field current. The introduction of variable resistance in the armature circuit or field circuit allowed speed control. Modern DC motors are often controlled by power electronics systems called DC drives.

3.6.1 Usage:

The DC motor or Direct Current Motor to give it its full title, is the most commonly used actuator for producing continuous movement and whose speed of rotation can easily be controlled, making them ideal for use in applications were speed control, servo type control, and/or positioning is required. A DC motor consists of two parts, a "Stator" which is the stationary part and a "Rotor" which is the rotating part. The result is that there are basically three types of DC Motor available.

Fig 3.16: DC Motor

3.6.2 Applications:

Gear motors are commonly used in commercial applications where a piece of equipment needs to be able to exert a high amount of force in order to move a very heavy object. Examples of these types of equipment would include a crane or lift jack.

- Agitator, Conveyors, Carriage Drives & various material handling equipment.
- Industries– Sugar, Paper, Cement, Pharmaceuticals, Food processing

CHAPTER 4

SOFTWARE SPECIFICATIONS

4.1 Arduino IDE:

The IDE stands for Integrated Development Environment. Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. WRITING SKETCHES Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Note: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde . It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Verify

Checks your code for errors compiling it.

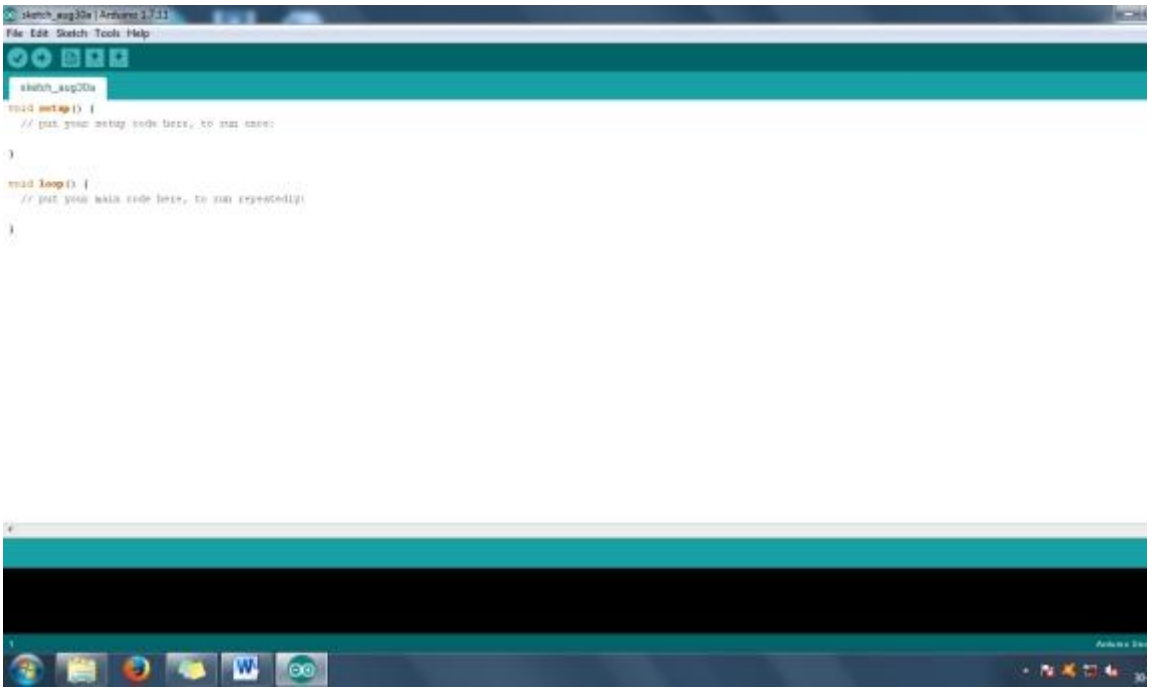
Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.



Open

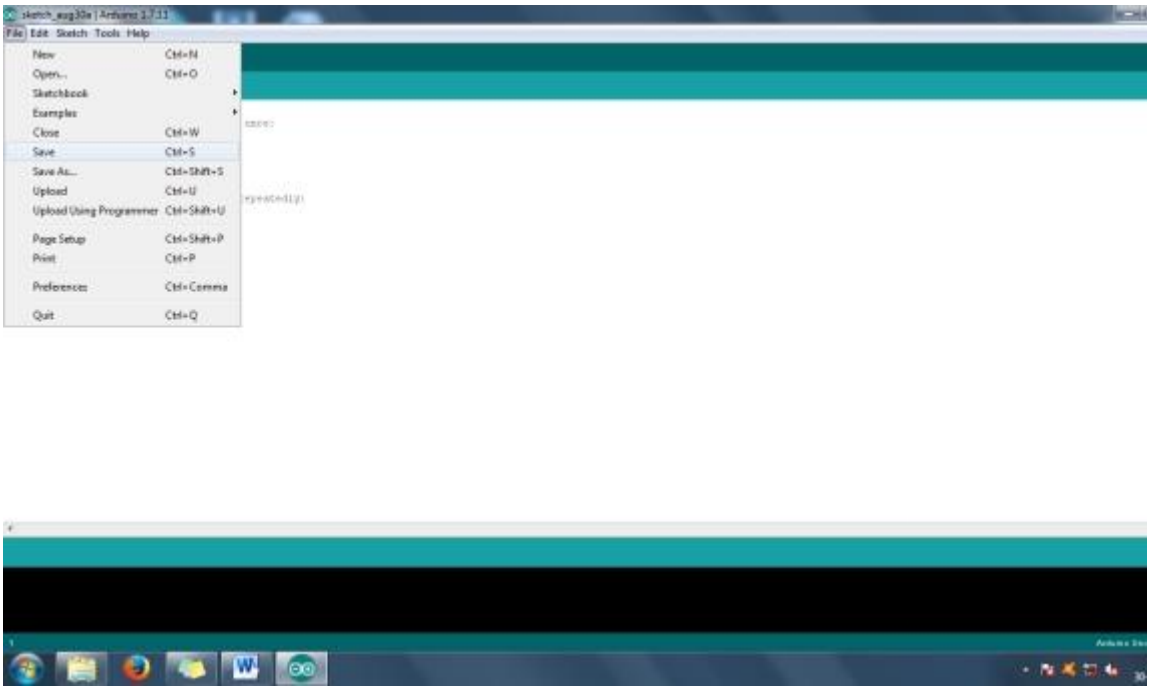
Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch

late in the list, use the File | Sketchbookmenu instead.

Save

Saves your sketch.



Serial Monitor

Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- New

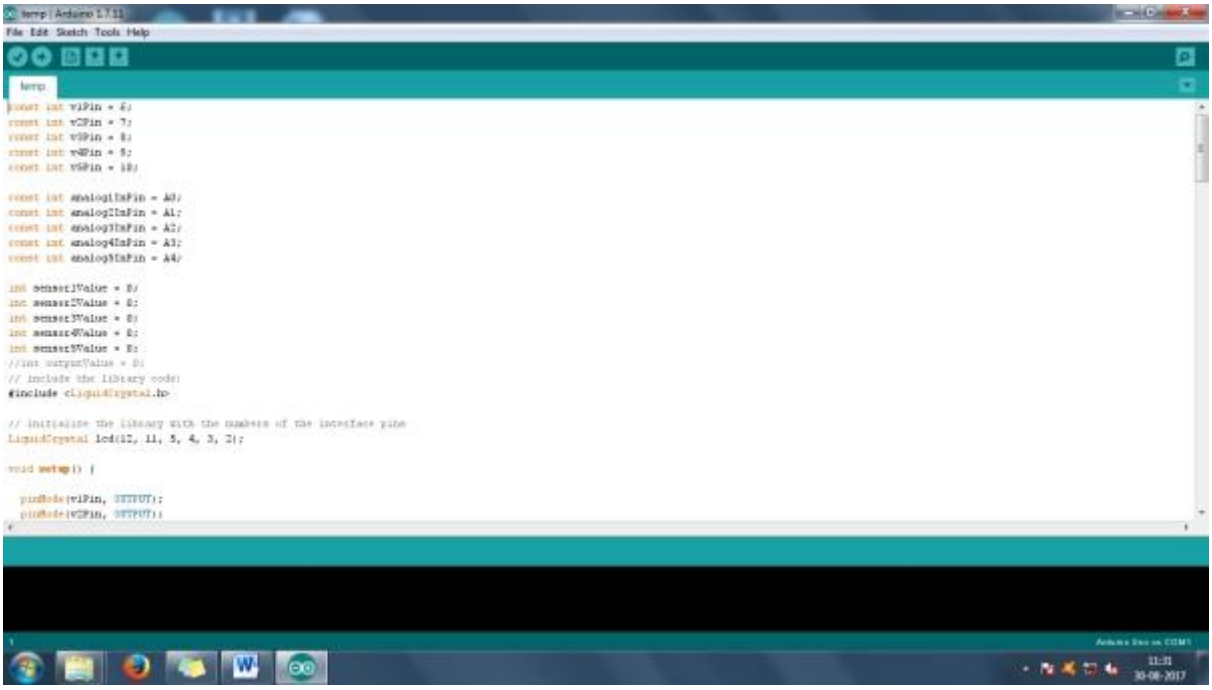
  Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- Open

  Allows to load a sketch file browsing through the computer drives and folders.
- Open Recent

  Provides a short list of the most recent sketches, ready to be opened.

Sketchbook

- Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.



- Close

  closes the instance of the Arduino Software from which it is clicked.
- Save

  saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- Save as...

  allows saving the current sketch with a different name.
- Page Setup

  It shows the Page Setup window for printing.

- Print

  sends the current sketch to the printer according to the settings defined in Page Setup.
- Preferences

  opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- Quit

  closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE

Edit

- Undo/Redo

  goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- Cut

  removes the selected text from the editor and places it into the clipboard.
- Copy

  duplicates the selected text in the editor and places it into the clipboard.
- Copy for Forum

  copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax colouring.
- Copy as HTML

  copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- Paste

  puts the contents of the clipboard at the cursor position, in the editor.
- Select All

  selects and highlights the whole content of the editor.
- Comment/Uncomment

  puts or removes the // comment marker at the beginning of each selected line.

- Increase/Decrease Indent

  adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

- Find

  opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

- Find Next

  highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

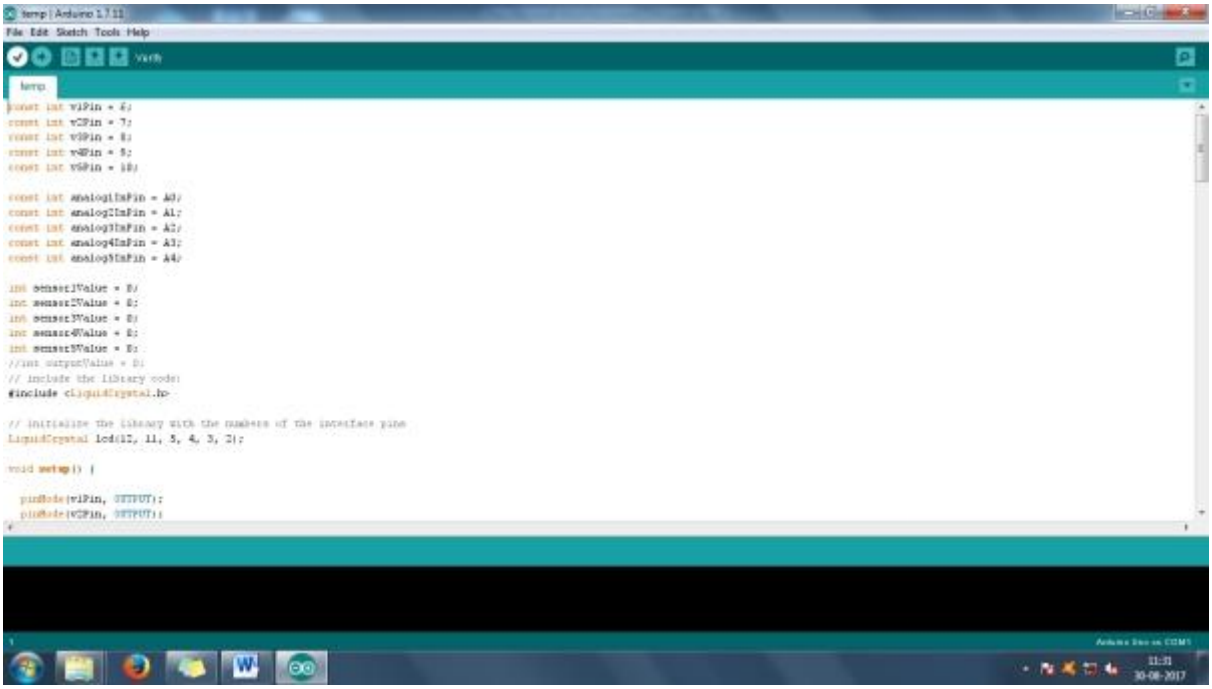
- Find Previous

  highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

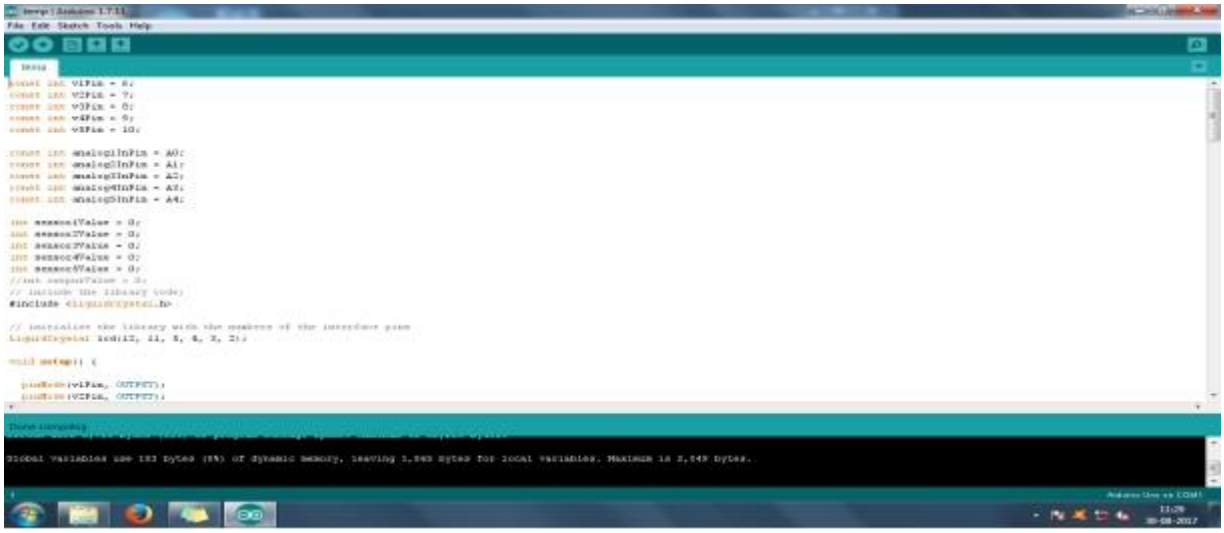
- Verify/Compile

  checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.



- Upload

  compiles and loads the binary file onto the configured board through the configured Port.



Upload Using Programmer

This will overwrite the boot loader on the board; you will need to use Tools > Burn Boot loader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Boot loader command must be executed.

Export Compiled Binary

Saves a .hex file that may be kept as archive or sent to the board using other tools.

Show Sketch Folder

Opens the current sketch folder.

Include Library

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

Add File...

Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

Tools

- Auto Format

  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- Archive Sketch

  Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- Fix Encoding & Reload

  Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- Serial Monitor

  Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- Board

  Select the board that you're using. See below for descriptions of the various boards.
- Port

  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- Programmer

  For selecting a hardware programmer when programming a board or chip and not using the on board USB-serial connection. Normally you won't need this, but if you're burning a boot loader to a new micro-controller, you will use this.
- Burn Boot loader

  The items in this menu allow you to burn a boot loader onto the micro-controller on

an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega micro-controller (which normally come without a boot loader). Ensure that you've selected the correct board from the Boards menu before burning the boot loader on the target board. This command also set the right fuses.

- Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

- Find in Reference

  This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File >Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino. Tabs, Multiple Files, and Compilation. Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino boot loader, a small program that has been loaded on to the micro-controller on your board. It allows you to upload code without using any additional hardware. The boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the micro-controller. The boot loader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch,

they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #includestatements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, boot loaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "Arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

Serial Monitor

Displays serial data being sent from the Arduino or Genuino board (USB or serial board). To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux, the Arduino or Genuino board will reset (rerun your sketch execution to the beginning) when you connect with the serial monitor.

You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

Preferences

Some preferences can be set in the preferences dialog (found under the Arduino menu on the Mac, or File on Windows and Linux). The rest can be found in the preferences file, whose location is shown in the preference dialog.

Language Support

Fig 4.1 : Language Setup for Arduino IDE

Since version 1.0.1 , the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English.

You can return the software to its default setting of selecting its language based on your operating system by selecting System Default from the Editor Language drop-down. This setting will take effect when you restart the Arduino Software (IDE). Similarly, after

changing your operating system's settings, you must restart the Arduino Software (IDE) to update it to the new default language.

Boards

The board selection has two effects: it sets the parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches; and sets and the file and fuse settings used by the burn bootloader command. Some of the board definitions differ only in the latter, so even if you've been uploading successfully with a particular selection you'll want to check it before burning the bootloader. You can find a comparison table between the various boards here.

Arduino Software (IDE) includes the built in support for the boards in the following list, all based on the AVR Core. The Boards Manager included in the standard installation allows adding support for the growing number of new boards based on different cores like Arduino Due, Arduino Zero, Edison, and Galileo and so on.

CHAPTER 5

IMPLEMENTATION

5.1 Flow Chart:

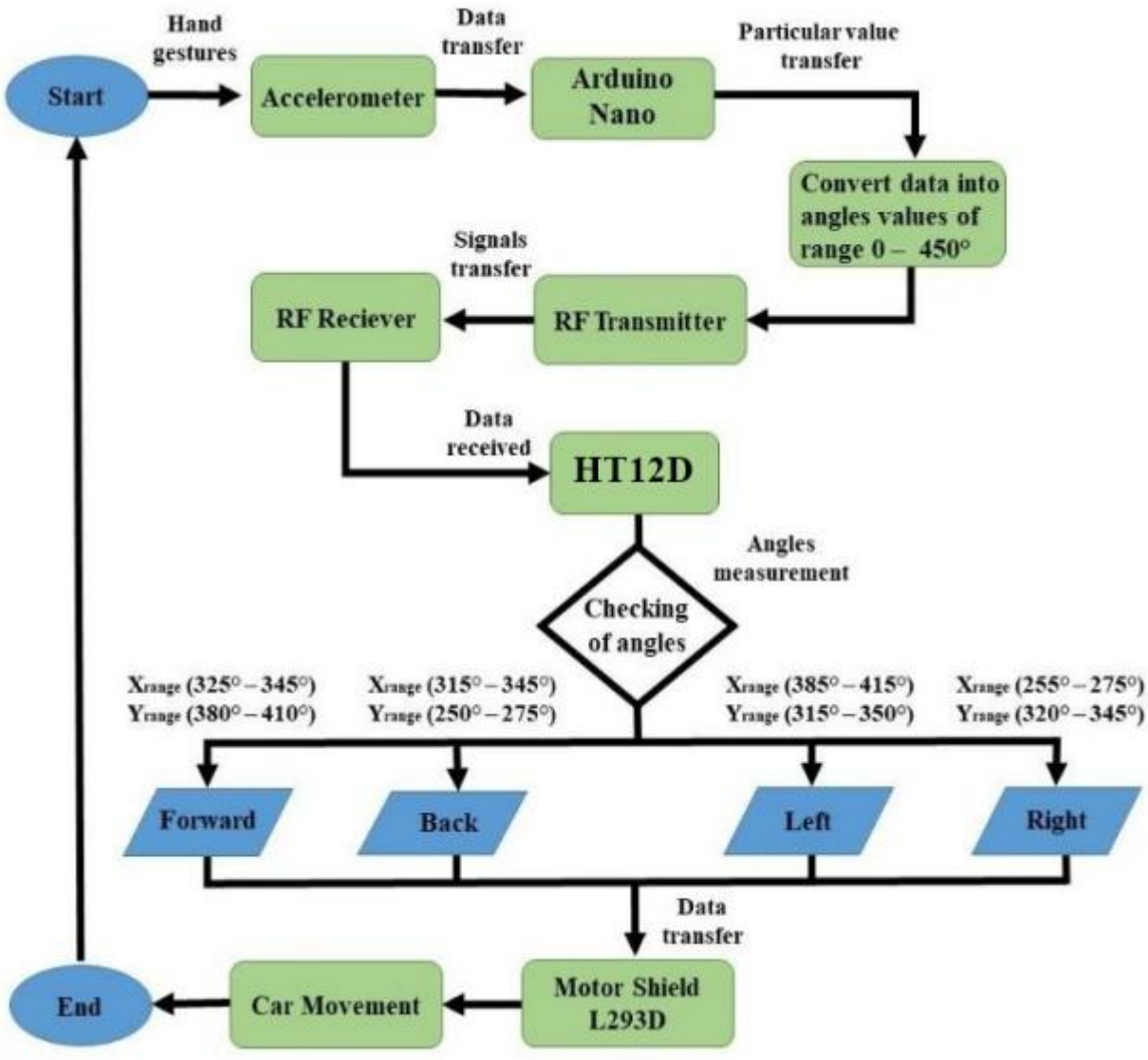


Fig 5.1: Flow Chart

The Gesture Controlled Robot working is divided into two sections:

1. Transmitter's End

2. Receiver's End

1. Transmitter's End:

- In this section of circuit an accelerometer sensor and an RF Transmitter is connected to the Arduino Nano.
- Slightest difference in the motion of the hand is detected and measured by the accelerometer.
- The above data provided by accelerometer is send to HT12E encoder.
- Encoded data is transmitted in to environment using RF transmitter.

2. Receiver's End:

- The transmitted data is received by the RF Receiver module and sent onto the decoder IC.
- The decoder IC decodes the coded waveform and the original data bits are recovered which is in the form of bits
- This data is passed onto the micro-controller which takes various decisions based on the received information.
- In turn it aides in the driving the DC motors.

5.2 Schematic Design:

The Schematic design of the complete system is divided into two sections i.e. the receiver's side and the transmitter's side.

The transmitter's side consists of the following components,

I) Arduino Nano
II) GY-521 Accelerometer

III) Transmitter module

The receiver's side consists of the following components,

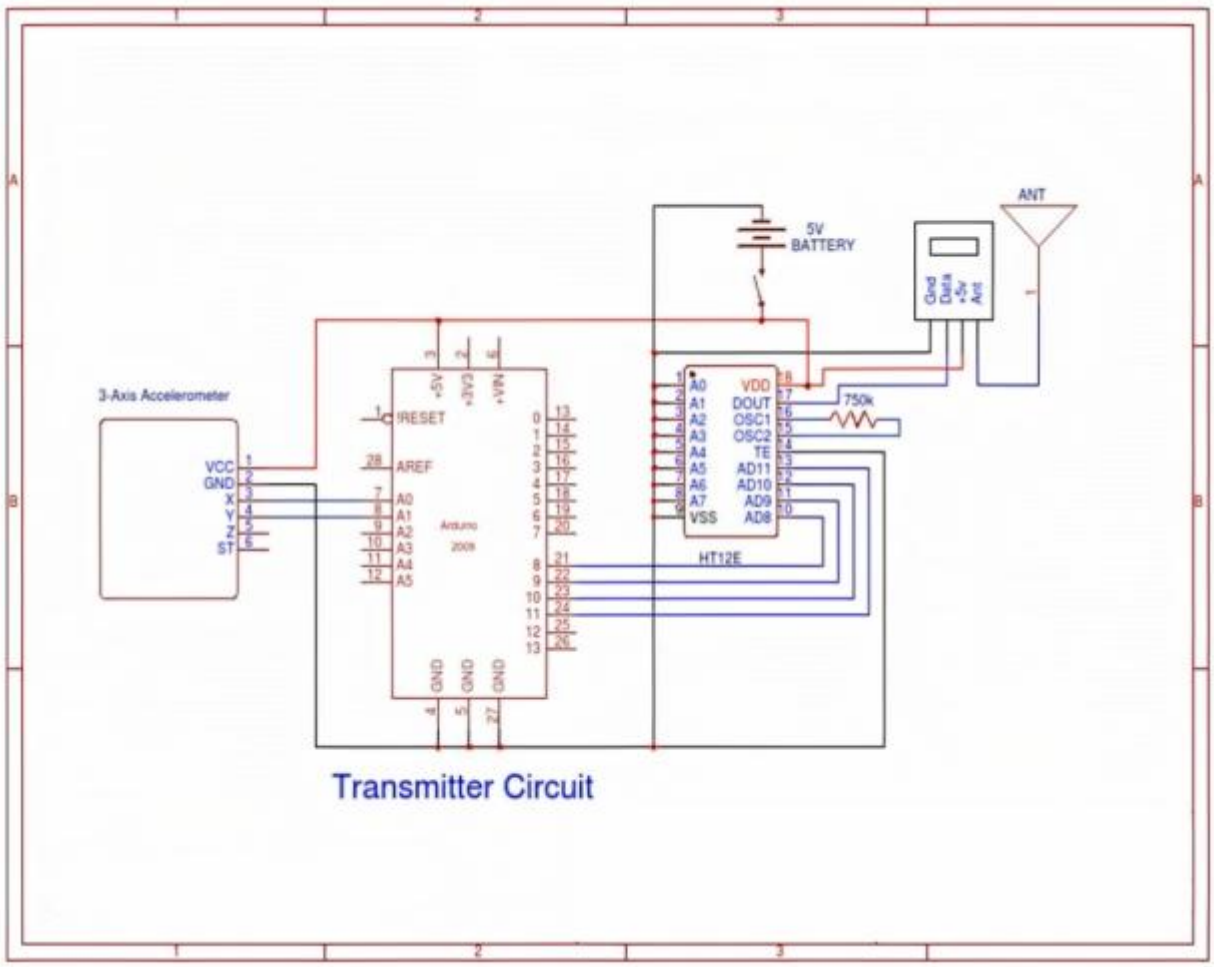
I) Motor Driver Module

II) Receiver Module

III) DC Motors

The schematics of the following to sections are shown below,

Transmitters Side Schematic Design:



Transmitter Circuit

Fig 5.2: Transmitters Side Circuit

Receivers Side Schematic Design:

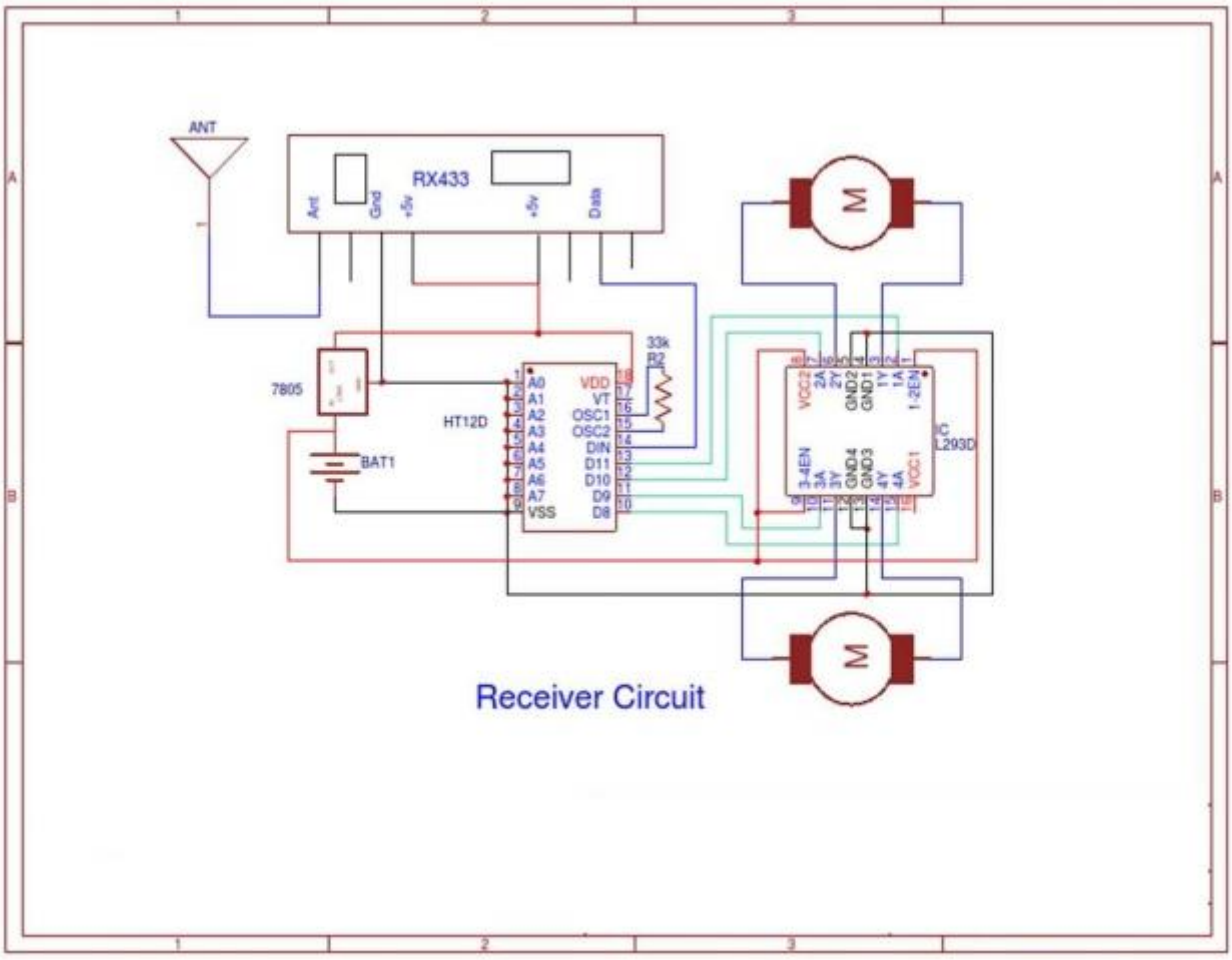


Fig 5.3 Receivers Side Circuit

Hence, as mentioned above we can see how the connections are made in the schematic diagram and also what all components are present in both the sides.

CHAPTER 6

SIMULATION & DESIGN

6.1 Simulation Work:

A gesture controlled robot is controlled by using hand in place of any other method like buttons or joystick. Here one only needs to move hand to control the robot. A transmitting device is used in your hand which contains RF Transmitter and accelerometer. This will transmit command to robot so that it can do the required task like moving forward, reverse, turning left, turning right and stop. All these tasks will be performed by using hand gesture.

Gesture controlled robot moves according to hand movement as we place transmitter in our hand. When we tilt hand in front side, robot start to moving forward and continues moving forward until next command is given. When we tilt hand in backward side, robot change its state and start moving in backwards direction until other command is given. When we tilt it in left side Robot get turn left till next command. When we tilt hand in right side robot turned to right. And for stopping robot we keeps hand in stable.

6.2 Design Screenshots:

As we have discussed above the complete system (robot) is divided into two sections. Hence below we can see the screenshots of the two sections separately,

i) Receiver's Side
ii) Transmitter's Side

Fig 6.1: Receiver's Side

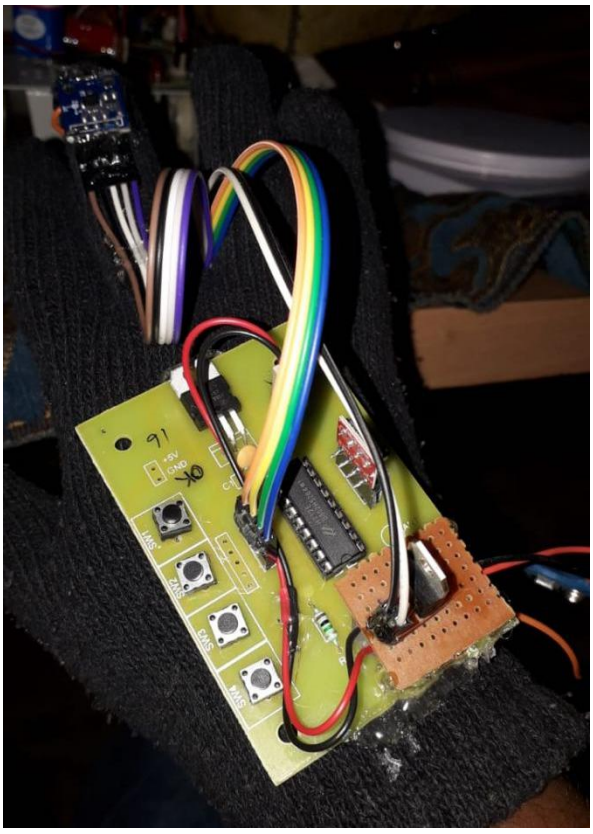


Fig 6.2: Transmitter's Side

6.3 Output Screenshots:

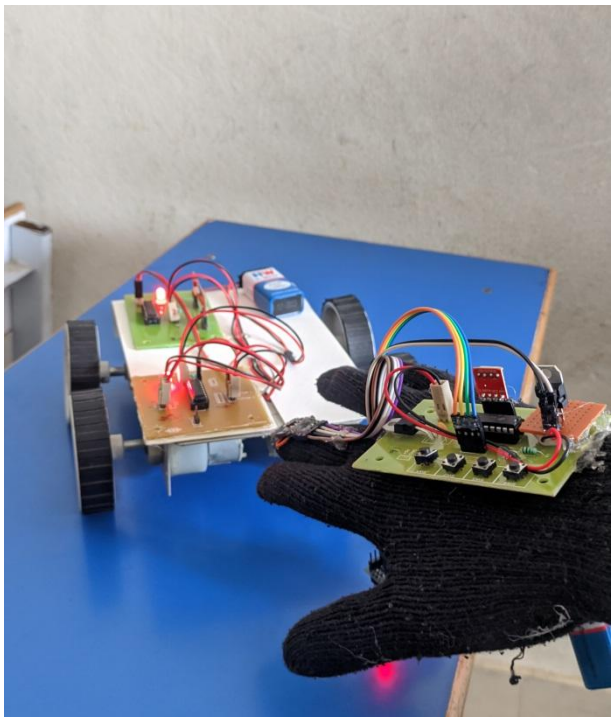


Fig 6.3: Stationary

The above figure 6.3 shows that the robot is at rest when the hand is stationary and exactly parallel to the ground, none of the motors are driven which means that no voltage is passed to the dc motors.

The robot is in a straight forward movement as the hand is tilted forward and bent. Here both the motors have same amount of voltage supplied to them.

The robot is moving to the left as the hand is tilted towards the left side from its stationary position. Here the motors receive voltages in the opposite directions as mentioned below,

- Left motor moves in clockwise direction.
- Right motor moves in a anti-clockwise direction.

The robot is moving to the right as the hand is tilted towards the right side from its stationary position. Here the motors receive voltages in the opposite directions as mentioned below,

- Left motor moves in anti-clockwise direction.
- Right motor moves in clockwise direction.

CHAPTER 7

CONCLUSION

The purpose of project is to control a toy car using accelerometer sensors attached to a hand glove. The sensors are intended to replace the remote control that is generally used to run the car. It will allow us to control the forward and backward, and left and right movements, while using the same accelerometer sensor to control the throttle of the car. based on the hand movements. By using the above mentioned components the hardware was setup, thus resulting in the formation of a robot. In order to implement the experiment a Dell laptop was used, whose web camera acted as the input device for capturing the video. The software part was developed in Java for image processing wherein the hand gestures were analysed to extract the actual direction. Eclipse Ide was used for developing the java code. The direction thus identified was send as characters to the robot with the help of Zigbee. XBee S2 version of Zigbee was used for enabling the communication. The final movement of the robot can be concluded as follows: At the beginning the robot was in a stop mode. As the hand moved from bottom to top, the robot moved in the forward direction. As the hand moved from top to bottom, the robot moved in the backward direction. As the hand was shown as an acute angle towards the left, the robot moved towards the left direction. As the hand was shown as an acute angle towards the right, the robot moved towards the right direction. As the hand is kept stationary with respect to the environment, the robot was in the stop mode. From the experiment, about 80% of the implementation worked according; the remaining was less due to background interference which is a negative marking to the implementation. Hand Gesture Controlled Robot System gives a more natural way of controlling devices. The command for the robot to navigate in specific direction in the environment is based on technique of hand gestures provided by the user. Without using any external hardware support for gesture input unlike specified existing system, user can control a robot from his software station.

FUTURE SCOPE:

1) The on board batteries occupy a lot of space and are also quite heavy. We can either use some alternate power source for the batteries or replace the current DC Motors with ones which require less power.

2) The proposed system is applicable in hazardous environment where a camera can be attached to the robot and can be viewed by the user who is in his station. This system can also be employed in medical field where miniature robot are created that can help doctors for efficient surgery operations For more efficient response, threshold values can be used to detect gesture and advanced features such as finger counts that provide different functional commands can be used.

3) Entertainment applications – Most videogames today are played either on game consoles, arcade units or PCs, and all require a combination of input devices. Gesture recognition can be used to truly immerse a players in the game world like never before.

4) Automation systems – In homes, offices, transport vehicles and more, gesture recognition can be incorporated to greatly increase usability and reduce the resources necessary to create primary or secondary input systems like remote controls, car entertainment systems with buttons or similar.

APPENDICES

APPENDIX – A

CODE:

```
void setup() {
   // initialize the serial communications:
   Serial.begin(9600);
   pinMode(rf1,OUTPUT);
   pinMode(rf2,OUTPUT);
   pinMode(rf3,OUTPUT);
   pinMode(rf4,OUTPUT);


   digitalWrite(rf1, LOW);
   digitalWrite(rf2, LOW);
   digitalWrite(rf3, LOW);
   digitalWrite(rf4, LOW);


}

void loop() {

   // print the sensor values:
   int a,b,c;
   a=analogRead(xpin);
   b=analogRead(ypin);
   Serial.print("\t");
   c=analogRead(zpin);
if(a<=290)
{


   Serial.print("back");
   digitalWrite(rf1, LOW);
   digitalWrite(rf2, HIGH);
   digitalWrite(rf3, LOW);
   digitalWrite(rf4, LOW);
   delay(100);
```

```
}
else if(a>=380)
{


   Serial.print("front");


   digitalWrite(rf1, HIGH);
   digitalWrite(rf2, LOW);
   digitalWrite(rf3, LOW);
   digitalWrite(rf4, LOW);
   delay(100);
}
else if(b<=283)
{


   Serial.print("left");


  digitalWrite(rf1, LOW);
   digitalWrite(rf2, LOW);
   digitalWrite(rf3, LOW);
   digitalWrite(rf4, HIGH);
      delay(100);

}
else if(b>=397)
{


   Serial.print("right");


digitalWrite(rf1, LOW);
   digitalWrite(rf2, LOW);
   digitalWrite(rf3, HIGH);
   digitalWrite(rf4, LOW);
   delay(100);

}
else if(c>=380)
{
Serial.print("stop");
```

```
    digitalWrite(rf1, LOW);
    digitalWrite(rf2, LOW);
    digitalWrite(rf3, LOW);
    digitalWrite(rf4, LOW);
    delay(100);

  delay(10);
}
}
```

APPENDIX – B

BIBLIOGRAPHY

REFERENCES:

[1] Nitin and Naresh, "Gesture Controlled Robot PPT", [http://seminarprojects.com/s/hand-gesturecontrolled-]

[2] Naveet Kumar, Neeraj Purohit, "Gesture ControlledTank Toy User Guide" [http://www.slideshare.net/neeraj18290/wireless-gesturecontrolled-tank-toy-transmitter]Accessed 13 October 2013.

[3] Jochen Triesch and Christoph Von Der Malsburg "Robotic Gesture Recognition (1997)" [http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.5427] Accessed 15 October 2013.

[4] "Real-Time Robotic Hand Control Using Hand Gestures" by Jagdish Lal Raheja, Radhey Shyam, G. Arun Rajsekhar and P. Bhanu Prasad.

[5] Bhosale Prasad S., Bunage Yogesh B. and Shinde Swapnil V. "Hand Gesture Controlled Robot"

[6] [http://www.engineersgarage.com/contribution/accelerometer-based-hand-gesture-controlled-robot] Accessed 3 November, 2013.

[7] [http://www.robotplatform.com/howto/L293/motor_driver_1.html] Accessed 5 November, 2013.

[8] [http://en.wikipedia.org/wiki/Gesture_interface] Accessed 5 November, 2013.

[9] [http://www.wisegeek.com/what-is-a-gear-motor.htm] Accessed 6 November, 2013.

[10] [http://www.scribd.com/doc/98400320/InTech-RealTime-Robotic-Hand-Control-Using-Hand-Gestures] Accessed 6 November, 2013.

APPENDIX - C

BIOGRAPHY

NAME : VALPADASU SAI SUDHAMSH

ROLL NO : 18831A04B2

EMAIL : SAIVALPADASU30@GMAIL.COM

PHONE : 81795986360