# A Project Report On

VOICE CONTROLLED INDUSTRIAL AUTOMATED ROBOT.

Guided by

# Prof. A. P. Gargade Submitted By

1. Subhash Mangale (18132063) 3. Sankalp Hoshing (18131009)
2. Sandip Rokade (18131003) 4. Narayan Yadav (18131026)



# Department of Electronics and Telecommunication Engineering GOVT. COLLEGE OF ENGGINEERING AND RESEARCH AWASARI (KHURD), TAL- AMBEGAON, DIST- PUNE 412405

**Academic Year: 2021-2022**

# GOVT. COLLEGE OF ENGG. AND RESEARCH AWASARI (KHURD), TAL- AMBEGAON, DIST- PUNE 412405



**CERTIFICATE**

*This is to certify that following students of B.E. (Electronics and Telecommunication), have done bonafide work on the project entitled –* ***“VOICE CONTROLLED INDUSTRIAL AUTOMATED ROBOT.”***

*They are allowed to submit this work to the Savitribai Phule Pune University towards partial fulfilment of the requirement for the award of Bachelor of Engineering (Electronics and Telecommunication) during the year 2021-2022.*

1. Subhash Mangale (18132063) 3. Sankalp Hoshing (18131009)
2. Sandip Rokade (18131003) 4. Narayan Yadav (18131026)

|  |  |  |
| --- | --- | --- |
| Prof. A. P. Gargade | Dr. M. S. Nagmode | Dr. D. R. Pangavhane |
| **Project Guide** | **Head of Dept.** | **Principal** |

## ACKNOWLEDGEMENT

We consider it as a privilege to express a few words of gratitude and respect to all those who helped us with the project. We feel ourselves honoured to place warm regards Govt. collage of engineering & research, Awasari (kh). We are also thankful to our Principal Dr. D. R. Pangavhane for his support for doing the project.

We would like to express a sincere gratitude to our mentor and Head of Department (E&TC), Dr. M. S. Nagmode Government college of Engineering and Research, Awasari (kh), Pune for his supervision and timely support. We thank him for his constant encouragement and guidance for achieving the goal.

We are also grateful to our project guide Mr. A. P. Gargade for the skilful guidance, profound interest, constructive suggestions, critical evaluations and perceptual encouragement during the conduct of investigation and preparation of manuscript. We shall remain grateful to him for the same forever.

Finally, we are grateful to all the staff members, non-teaching staff and all our friends for giving us the helping hand.

Subhash Mangale Sandip Rokade Sankalp Hoshing Narayan Yadav

The industrial automation is one of the major areas in current world of globalization. Students from various technical disciplines use their knowledge and invent some creative models in fields like robotics, software and security. Large and medium scale industries are using various types of robots nowadays which are working with high precision and accuracy. Our voice controlled robot project in which we are going to build one robot which will work on any type of travelling operation in the industry. We will be going to use software programming with electronic sensors to build moving robot on voice command as well as manual operating commands.

1. This project is based on idea so as to develop a voice command and manually controlled moving robot which has specific direction and motion for specific path.
2. We have used mechanical Omni directional wheels for our task. This motion is generated by using geometrical vector equations with the help of software path planning algorithms.
3. We have to only give a voice command or character command which will acts like a password for that specific path and with the help of such robot we can assign any task to the robot in industry like pick and place work, shifting objects to any transportation vehicle just by a single command.
4. Our main goal is to design mechatronics robot in which we used only two sensors and one controller. At the starting of the project we decided that we will design simple construct but it can achieve complex tasks and we think we have achieved that.

|  |  |  |
| --- | --- | --- |
| Chapter No. | Contents | Page No. |
| 1. | **INTRODUCTION**   * 1. Introduction   2. Need of Robot   3. Objective of Robot | i-ii |
| 2. | **REVIEW AND LITERATURE SURVEY**   * 1. Historical development   2. Present Robot in the market   3. Paper Study   4. Drawbacks of current system | iii-vi |
| 3. | **SYSTEM DEVELOPMENT**   * 1. **HARDWARE**      1. Block Diagram      2. Block diagram Description      3. Hardware Description      4. Circuit Design   2. **SOFTWARE**      1. Software used      2. Algorithms used   3. **WORKING OF SYSTEM** | vii-xxxvii |
| 4. | **TESTING** | xxxvii-xxxviii |
| 5. | **RESULT AND DISCUSSION**   * 1. Performance Graph of Output   2. Result and Discussion |  |

|  |  |  |
| --- | --- | --- |
|  | * 1. Advantages   2. Drawbacks   3. Applications   4. Project Image   5. Estimation Cost | xxxix-xlv |
| 6. | **CONCLUSION AND FUTURE SCOPE** | xlvi |
| 7. | **REFERENCES** | xlvii |
| 8. | **SOFTWARE PROGRAM** | xlviii-lxi |
| 9. | **DATASHEETS** |  |

# LIST OF FIGURES:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of Figure** | **Page No.** |
| 1. | Block diagram | viii |

**LIST OF TABLES:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of Table** | **Page No.** |
| 1. | Pin description of DC motor driver | xix |
| 2. | Control signal description of DC motor driver | xx |
| 3. | Path planning algorithm-direction description | xxxii |
| 4. | Cost estimation | xlv |

# LIST OF GRAPHS:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of Table** | **Page No.** |
| 1. | Kp vs. Gain (PWM) performance graph | xl |

**LIST OF PICTURES:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of Pictures** | **Page No.** |
| 1. | Four wheel robot | iv |
| 2. | Line follower robot | iv |
| 3. | Arduino MEGA | x |
| 4. | Arduino MEGA pin diagram | xii |

|  |  |  |
| --- | --- | --- |
| 5. | Atmega328P pin diagram | xiii |
| 6. | IMU MPU 6050 | xiv |
| 7. | IMU Angles | xv |
| 8. | Bluetooth Module HC-05 | xvi |
| 9. | L298 DC Motor Driver | xviii |
| 10. | DC Motor | xxi |
| 11. | Servo Motor | xxii |
| 12. | Voice app initial screen | xxiii |
| 13. | Manual app initial and control screen | xxiv |
| 14. | Supply LIPO battery | xxv |
| 15. | Omni wheel | xxvi |
| 16. | Circuit Diagram | xxvii |
| 17. | Arduino IDE | xxviii |
| 18. | PID Algorithm | xxix |
| 19. | Vector Equation of Omni wheels velocity | xxxi |
| 20. | Tri-Wheel Chassis | xxxii |
| 21. | Project image | xliv |

# CHAPTER 1 INTRODUCTION

## Introduction

Presently there are lot of industrial assistant robots available in market but either they are too costly or they application specific. There is still need of less costly robot which can provide a simple construction and complex working with ease of understanding to third party user.

In our project, we develop a path planning robot to reduce the cost and which are available in the market. We set our robot to operate in both voice commands as well as in the manual commands controlling mode. We used two mobile applications for robot handling purpose which are available on Google play store so that any third party used can easily access it.

We use Bluetooth standard for the communication purpose. By installing mobile application on the phone, we enable the Bluetooth of mobile to connect with the Arduino Mega Microcontroller. Once we established the connection with Arduino Mega then by using the mobile application GUI we control the robot as our wish with either by Voice commands or android manual command.

Voice controlled robot is a multipurpose industrial travelling robot which can move in any direction as per the needs of user and setting of programmer. Our robot is of the three wheeled robot.

Our Robot body is made in such a way that it does not have any limitations like normal 4 wheel robot. We have used Omni wheels so that it can achieve a Holonomic motion.

This robot can be used in Industrial Purpose as well as in Commercial Sector for travelling operation. E.g. Home assistance robot, Hobby kit robot and for researchers it can be used as well.

## Need of Our Robot

1. Now a day there is increase in the technology and increases the task in technical field.
2. Automation industry requires fast completion of work with accurate results.
3. Different Path planning robots are required to achieve such tasks because of their holonomic drive as well as they can achieve 360 degree rotation with respect to single centroid point.
4. Medium and small scale industries have less valuation so that they can’t afford costly robots. Human being not able to work in such places.
5. It is our main intention to build such robot which can afford to any start-ups as well as small scale industries which can achieve their complete inter-transportation task with high precision.

## Objective of the system

1. We want to build a robot which can be used as complete manager of inter transportation operation in small scale industries.
2. We aim to build our robot in minimum cost.(almost less than < 10000 INR)
3. Robot must have move in all direction with holonomic motion with 0 degree deviation and avoid material damage.
4. To make Bluetooth controlled app supporting robot which can be used by any third party user so that it can easily accessible.
5. To make voice command as well as manual command operated robot for ease of handling to the user means user is not required to control its all function.

# CHAPTER 2

**LITERATURE SURVEY**

## Historical development

The history of robots has its origin from the ancient world. The modern concept began to be developed with the onset of the Industrial Revolution which allowed for the use of complex mechanisms and the subsequent introduction of electricity. This made it possible to power machines with small compact motors.

In the early 20th century, the notion of a humanoid machine was developed

. Today, it is now possible to made human sized robots with the capacity like human thoughts and movements. The first use of modern robots were in factories as industrial robots - simple fixed machines capable of manufacturing tasks which allowed production without the need for human assistance.

Digitally controlled industrial robots make use of artificial intelligence have been built since the 1960s. First robot vehicle concept introduced by so many Automation industries from 1970’s. Sony launches the first robot in 2006. Along with the Sony the Tapia, Asus etc.

Different Hobby kits robot like line follower and image vision robots are also used in industries as well. It has high cost as well as maintaining is difficult.

The first consumer model was introduced on May 11, 1999. New models were released every year until 2005. Although most models were used in higher venture industries and other inspirations included lion - cubs and space explorer, and only the final ERS - 7 versions was explicitly introduced.

## Present robots in the market

**2.2.1 Four wheel robot:**



Picture 2.2.1 Four wheel robot

Four wheel robots are used in small robotics and in hobby kits. It was the first small robotics vehicle that is introduced in industries for travelling purpose. Initially all operations are driven by that vehicle with its drivers in industry. Students and robotics enthusiast still using these type of robots for some applications.

## 1.2.2 Line follower controlled robot



Picture 2.2.2 Line follower robot

The line follower robot is a mobile machine that can detect and follow the line drawn on the floor. Generally, the path is predefined and can be either visible like a black line on a white surface with a high contrasted colour. It uses IR sensor to detect the while and black colour to follow the line draw on surface.

* 1. **Paper Study**

Industrial Robots are one of the widely used mechatronic machines in today's world because of their less complex structure and low cost. Robotics automation companies like Siemens, ABB robotics always focused with people and industrial needs. They always try to simply the handling of robot and increase the working variability of robots. With the advancement in sensor miniaturizations and exponential increment in the speed and capability of microcontrollers, such robots are enough to maintain the work of group of humans.

According to R. Pahuja and N. Kumar in 2014 [6]. The technology was to be used in wirelessly controlled moving robots working in industrial as well as commercial sector.

Another useful contribution in the field was made Shuai Yuan [4] The paper mentioned roughly about the Intelligent path planning technology. In this invention, the path planning is controlled by an android application and are equipped with IOT.

After researching and finding the lot of information from such prominent papers [1] and [2] we are observed that there is so many information and advanced technology available in holonomic motion of Omni robots and dynamics information available for our project.

* 1. **Drawbacks of Current system**

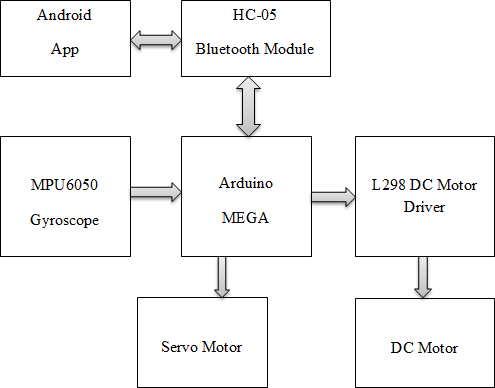
1. All these available robots are complex in construction and they are specific tasking in nature.
2. Sometimes due to use of high quality sensors cost is high.
3. IR sensors are fails due to change in light intensity and this is main drawback of the line follower robot.
4. Setting up of black or white lines on ground surface in an industry is not suitable.

# CHAPTER 3

**SYSTEM DEVELOPMENT**

# HARDWARE

## Block diagram



**Figure 3. 1 Block diagram**

## Blocks:

* + - 1. Android App
      2. HC-05 Bluetooth module
      3. MPU 6050 Gyroscope
      4. Arduino Mega
      5. L298 DC motor Driver
      6. Servo Motor
      7. DC motor

## Block diagram description

1. **Arduino MEGA:**

Arduino is world famous microcontroller development board which is used in so many robotic applications. We have used it as our main controller. We have used it for taking commands from Bluetooth module through the Android Application. We have also interfaced a gyroscope module with the Arduino to convert raw angle values into perfect angles which are in degrees.

## Bluetooth Module HC-05:

Bluetooth module HC-05 is used for communication purpose from android phone to Arduino microcontroller for taking commands from user either in voice format or in manually button commands. We have connected the Bluetooth module by using the UART communication protocol.

## MPU 6050 – Gyroscope Module:

Gyroscope module MPU6050 is used for taking angle values which are required for maintaining the holonomic motion and for application of PID (Proportional, Integral and Derivative) algorithm which is used to maintain the front side of robot in only one direction without changing the direction which is given by user. We have implemented the MPU6050 module by using I2C communication protocol with Arduino which gives faster data so as to perform the task efficiently.

## L298 DC motor driver:

L298 DC motor driver is famous, low cost and less complex driver module available in the market. It is used to give variable PWM signal means variable speed to the DC motor. This is the main unit of our robot because it helps us to maintain our PID algorithms as well as holonomic motion.

## DC Motor:

DC motor is connected to the wheels for locomotion. We have to perform certain direction like forward, backward, left and right. We have used 12V DC motors with speed of 200 RPM and torque of 2 Kg-cm.

## Servo Motor:

Servo motor is used for picking and placing mechanism. First robot picks the material at 0 degree and places the material at 180 degree.

## Android App:

It is used for communication between user and robot. User gives commands which are transferred to the Arduino Microcontroller using Bluetooth module.

## Hardware description

1. **Arduino MEGA:**



## Picture 3.1 1 Arduino MEGA

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 Analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-DC adapter or battery to get started.

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically. The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Each of the 54 digital pins on the Mega 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 a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms.

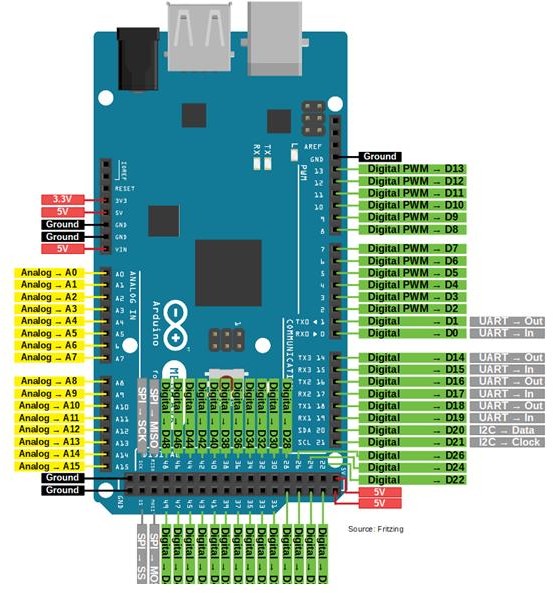
The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication.

The Arduino Mega2560 has a resettable poly fuse 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.

## Specifications:

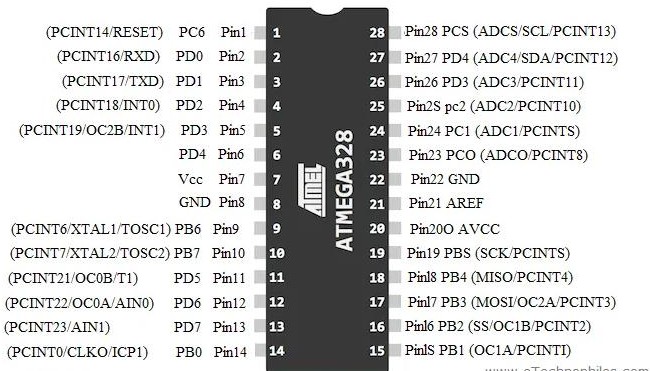
1. Operating voltage : 5V
2. Input voltage (recommended) : 7-12V
3. Input voltage (limits) : 6-20V
4. Digital I/O pins : 54 (of which 14 provide PWM output)
5. Analog input pins 16
6. DC current per I/O pin : 40mA
7. DC current for 3.3V pin : 50mA
8. Flash Memory : 256 KB, 8KB used by boot loader
9. SRAM : 8 KB
10. EEPROM : 4 KB
11. Clock Speed : 16 MHz

## Arduino MEGA Pin diagram:



**Picture 3.1 2 Arduino MEGA Pin diagram**

## ATmega 328 Pin diagram:



**Picture 3.1 3- AT mega 328 P pin diagram**

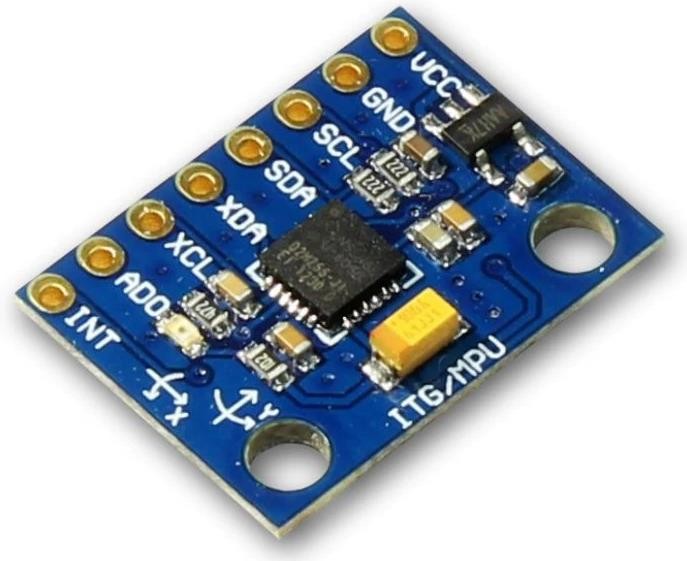
The ATmega328p is a single-chip, high-performance, efficient microcontroller created by Atmel in the megaAVR family. It is an 8-bit AVR RISC-based microcontroller chip.

It consists of **32 KB ISP flash memory** with read-while-write capabilities, **2 KB SRAM**(Static RAM), **1 KB of EEPROM**, **23 general-purpose I/O pins**, a **16MHz clock**, 32 general purpose working registers, three flexible timer/counters with compare modes (two 8 bits and one 16 bit), internal and external interrupts, serial programmable UART, a byte-oriented I2C (inter- integrated circuit) interface pins, SPI serial port, 6-channel 10-bit Analog to Digital converter, programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between a voltage range of 1.8-5.5 volts.

## Specifications:

|  |  |
| --- | --- |
| 1. Program Memory type 2. Program Memory size | : Flash  : 32 KB |
| 3. SRAM | : 2048 |
| 4. Data EEPROM | : 1024 |
| 5. Temperature | : -40 to 85 degree Celsius. |
| 6. Pin Count | : 32 |

1. **IMU MPU 6050 Gyroscope:**



## Picture 3.1 4 IMU MPU 6050

The MPU-60X0 is the world’s first integrated 6-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package. With its dedicated I2C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis Motion Fusion output.

The MPU-60X0 Motion Tracking devices, with its 6-axis integration, on- board Motion Fusion, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers.

Precision tracking of both fast and slow motions, the parts feature a user- programmable gyroscope full-scale range of ±250, ±500, ±1000, and ±2000°/sec (dps) and a user-programmable accelerometer full-scale range of ±2g, ±4g, ±8g, and ±16g. This module has some famous features which are easily accessible, due to its easy availability it can be used with a famous microcontroller like Arduino. Friend if you are looking for a sensor to control a motion of your Drone, Self Balancing Robot, RC Cars and something like this, then MPU6050 will be a good choice for you.

## Angle Calculation:



**Picture 3.1 5 IMU angles**

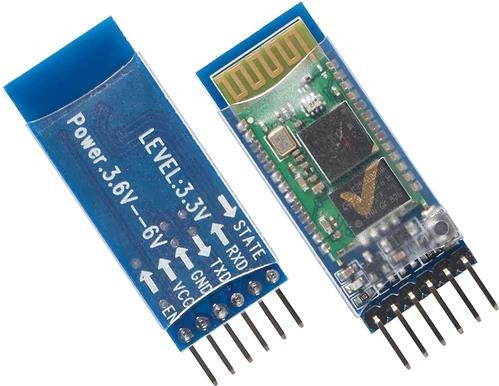
## Types of Angles:

1. Angle in X-direction : Roll
2. Angle in Y-direction : Pitch
3. Angle in Z-direction : Yaw(we used this only for our robot.)

## Specifications:

1. On board 3.3V regulator.
2. I2C interface.
3. Gyroscope operating current 3.6 mA.
4. Crystal frequency 32.768 KHz.
5. Input power supply 5V.
6. Dimensions 3\*2\*1 cms.

## Bluetooth Module HC-05:



**Picture 3.1 6 Bluetooth Module HC 05**

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.

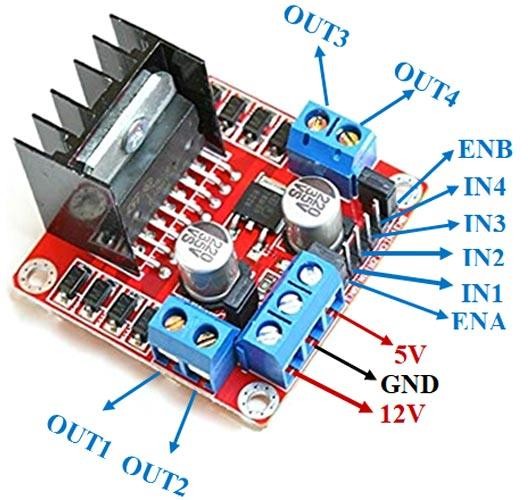
It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network ([PAN](https://en.wikipedia.org/wiki/Personal_area_network)). It uses frequency-hopping spread spectrum ([FHSS](https://en.wikipedia.org/wiki/Frequency-hopping_spread_spectrum)) radio technology to send data over air. It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

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. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulators. As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

## Specification:

1. Frequency : 2.4GHz ISM band.
2. Modulation : GFSK(Gaussian Frequency Shift Keying)
3. Power supply : +3.3VDC 50mA
4. Working temperature : -20 ~ +75Centigrade
5. Dimension : 26.9mm x 13mm x 2.2 mm
6. **L298 DC motor driver :**



**Picture 3.1 7 L298 DC Motor Driver**

This **L298N Motor Driver Module** is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. **L298N Module** can control up to 4 DC motors, or 2 DC motors with directional and speed control.

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.78M05 Voltage regulator will be enabled only when the jumper is placed.

When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller.

The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.

ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

## Specifications:

|  |  |
| --- | --- |
| 1. Driver Model | : L298N 2A |
| 2. Motor Supply Voltage | : 46V |
| 3. Motor Supply Current | : 2A |
| 4. Logic Voltage | : 5V |
| 5. Driver Voltage | : 5-35V |
| 6. Driver Current | : 2A |

**Pin description:**

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Pin Name** | **Description** |
| **1.** | IN1 & IN2 | Motor A input pins. Used to control the spinning direction of Motor A. |
| **2.** | IN3 & IN4 | Motor B input pins. Used to control the spinning direction of Motor B. |
| **3.** | ENA | Enables PWM signal for Motor A. |
| **4.** | ENB | Enables PWM signal for Motor B. |
| **5.** | OUT1 & OUT2 | Output pins of Motor A. |
| **6.** | OUT3 & OUT4 | Output pins of Motor B. |
| **7.** | 12V | 12V input from DC power Source. |
| **8.** | 5V | Supplies power for the switching logic circuitry inside L298N IC. |
| **9.** | GND | Ground pin. |

## Table 3.1 1 - Pin Description of L298 motor driver

**Control Signal motor A:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENA**  **(SPEED)** | **IN1**  **(Dir1)** | **IN2**  **(Dir2)** | **OP1** | **OP2** |
| LOW | LOW | LOW | LOW | LOW |
| LOW | HIGH | HIGH | LOW | LOW |
| HIGH | HIGH | LOW | HIGH | LOW |
| HIGH | LOW | HIGH | LOW | HIGH |

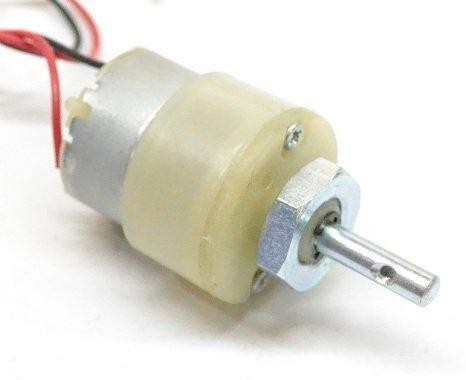
## Table 3.1 2 Control signal of One side of L298 DC Motor Driver

**Control Signal of Motor B:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ENB**  **(SPEED)** | **IN3**  **(Dir3)** | **IN4**  **(Dir4)** | **OP3** | **OP4** |
| LOW | LOW | LOW | LOW | LOW |
| LOW | HIGH | HIGH | LOW | LOW |
| HIGH | HIGH | LOW | HIGH | LOW |
| HIGH | LOW | HIGH | LOW | HIGH |

## Table 3.1 3 Control signal of One side of L298 DC Motor Driver

1. **DC Motor:**



## Picture 3.1 8 DC Motor

These motors are simple DC Motors featuring gears for the shaft for obtaining the optimal performance characteristics. They are known as Center Shaft DC Geared Motors because their shaft extends through the center of their gearbox assembly.

These standard size [DC Motors](https://robu.in/product-category/motor/) are very easy to use. Also, you don’t have to spend a lot of money to control motors with an [Arduino](https://robu.in/product-category/arduino-2/) or compatible board. The [L298N H-bridge module](https://robu.in/product/l298-based-motor-driver-module-2a/) with an onboard voltage regulator [motor driver](https://robu.in/product-category/motor-drivers/brushed-dc-motor-drivers/) can be used with this motor that has a voltage of between 5 and 35V DC.

This 12V DC Motor – 200RPM can be used in all-terrain robots and a variety of robotic applications. These motors have a 3 mm threaded drill hole in the middle of the shaft thus making it simple to connect it to the [wheels](https://robu.in/product-category/wheels/) or any other mechanical assembly.

Specifications:

|  |  |
| --- | --- |
| 1. Operating Voltage(V) | : 12 |
| 2. Rated Speed (RPM) | : 200 |
| 3. Rated Torque(kg-cm) | : 2.0 |
| 4. Stall Torque(kg-cm) | : 5.4 |
| 5. Load Current (A) | : 0.3 |
| 6. No Load Current (A) | : 0.06 |

## Servo Motor:



**Picture 3.1 9 SG90 Servo Motor**

Micro Servo Motor SG90 is a tiny and lightweight server motor with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

## Specifications:

1. Weight : 9 g
2. Dimension : 22.2 x 11.8 x 31 mm approx.
3. Stall torque : 1.8 kg·cm
4. Operating speed : 0.1 s/60 degree
5. Operating voltage : 4.8 V (~5V)
6. Dead band width : 10 µs
7. Temperature range : 0 ºC – 55 ºC

## Voice App Initial Screen:

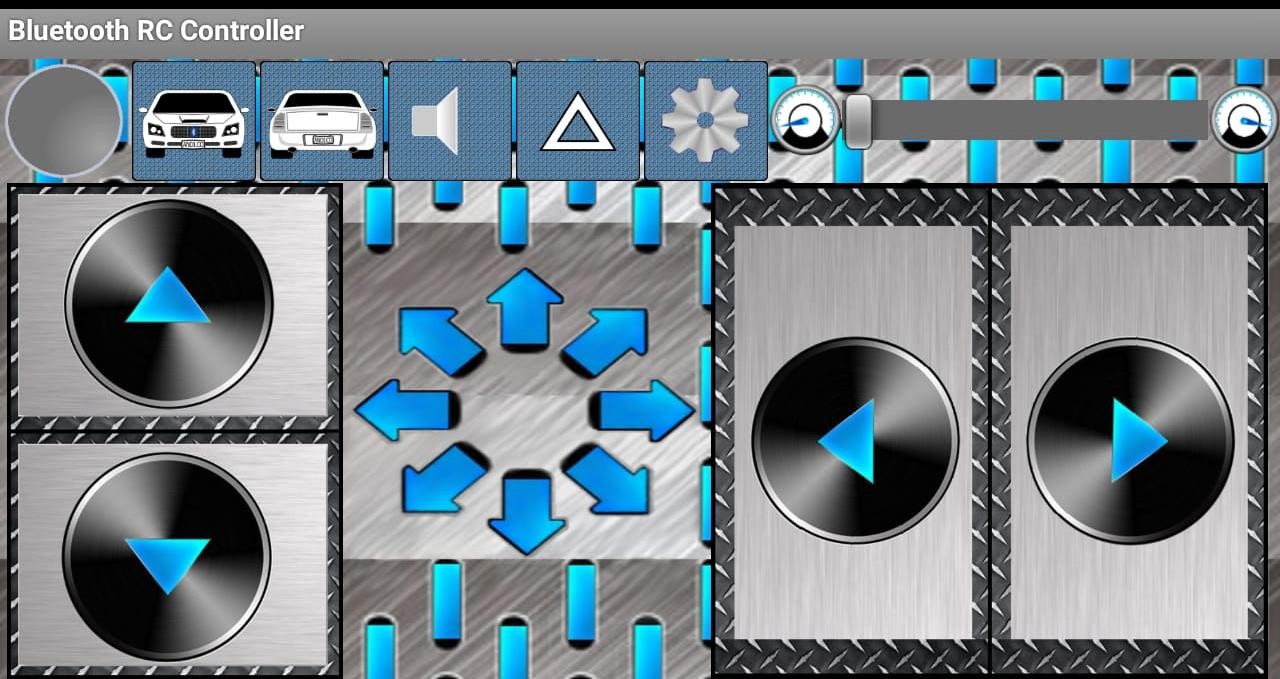


**Picture 3.1 10 Voice App Initial Screen Voice commands and their working:**

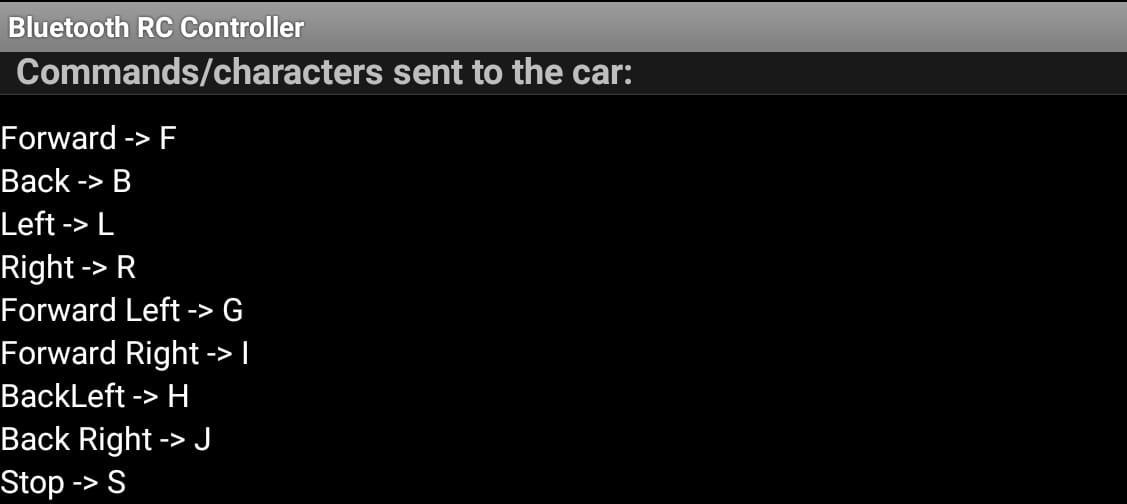
As we seen here we have to give command as follows:

1. F : Forward
2. B : Backward
3. L : Left
4. R : Right
5. W : Rotate robot by 90 degree
6. U : Servo picks and place by 0 and 180 degree respectively.

## Manual App Initial Screen:



**Picture 3.1 11 Manual App Initial Screen**



## Picture 3.1 12 Manual App Control screen and backend commands

**Control buttons and their function:**

As per the picture we seen that the symbols for motion:

1. Upper arrow : Forward
2. Down arrow : Backward
3. Left arrow : Left
4. Right arrow : Right

## Supply LIPO Battery:



**Picture 3.1 13 12V LIPO Battery**

11.1V 2200mAH LIPO battery is Capable of maximum continuous

discharge rates up to 30C, placing this battery among the most powerful Li-Po

battery packs in its class! It offers an excellent blend of weight, power, and

performance.

Specification:

|  |  |
| --- | --- |
| 1. Capacity (mAh) : | 2200 |
| 2. Weight (gm) : | 175 |
| 3. Output Voltage : | 11.1 V |
| 4. Charge Rate : | 1-3 C (Recommended) |
| 5. Discharge Plug : | XT-60 |
| 6. Balance Plug : | JST-XH |
| 7. Length (mm) : | 106 |
| 8. Width (mm) : | 34 |
| 9. Height (mm) : | 23 |
| 10. Max. Charge Rate : | 5 C |

## Omni wheels:



**Picture 3.1 14 Omni Wheels 58mm**

The 58mm Plastic Omni Wheel for Lego is the smallest Omni wheel with the loading capacity of 3kg. These wheels are compatible with Lego Motors and so with the Lego Robots.

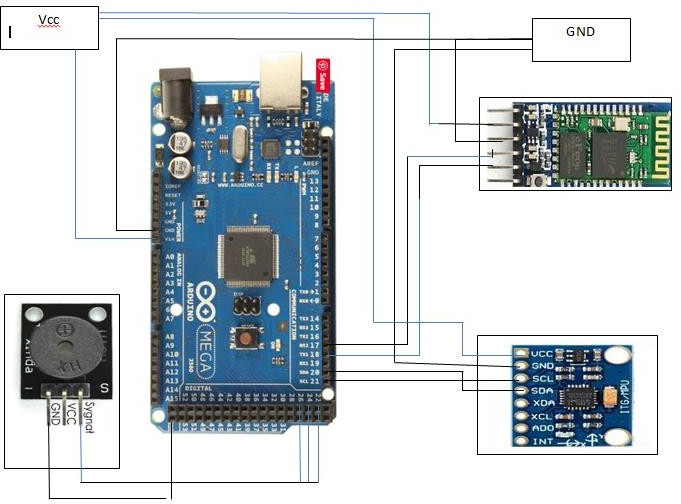
They feature rubber rollers along the circumference of the wheel which avoids slipping while moving sideways and gives minimum friction in movement.

This 58mm Plastic Omni Wheel gives your Lego robot more controllable degrees of freedom as compared to conventional wheels which give only 2 degrees of freedom i.e. moving forward and backward. The Omni Wheels provide easy 360° movement; with rotational and sideways maneuverability.

So, what makes this [wheels](https://robu.in/product-category/wheels/) move in all direction? Those are the [small](https://robu.in/product-category/wheels/wheel-accessories/rollers/) [rollers](https://robu.in/product-category/wheels/wheel-accessories/rollers/) along the wheel circumference. These [Rollers](https://robu.in/product-category/wheels/wheel-accessories/rollers/) are placed in such a way that the rotational axis of these [rollers](https://robu.in/product-category/wheels/wheel-accessories/rollers/) is perpendicular to the rotational axis of the main wheel.

The [Omni wheels](https://robu.in/product-tag/omni-wheel/) can rotate in a forward and a backward direction like ordinary wheels and also rotate freely around itself i.e. 360° rotation because of such two rotational axes within one wheel.

# Circuit Design

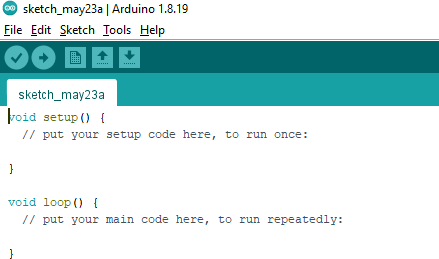


**Picture 3.1 15 Circuit Design**

# SOFTWARE

## Software used

**1) Arduino IDE:**



## Picture 3.2. 1 - Arduino IDE

The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as **Windows, Mac OS X, and Linux**.

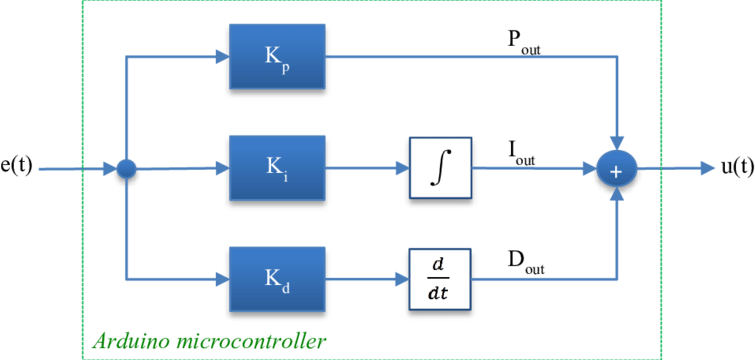
It supports the programming languages C and C++. Here, IDE stands for **Integrated Development Environment**.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuine and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

As seen in the above picture there are two in built functions are used by Arduino ide. Void setup() is used for all declaration and initial clock and serial baud rate declaration. Void loop() block is used for all definition and actual command that will be working continuously with the help of programming in C or C++ or Python.

## Algorithms

1. **PID algorithm:**



## Picture 3.2. 2 PID Algorithm

The main task of our robot is to implement the PID algorithm successfully and we successfully completed it. Here we have to consider some steps for making robot completely stable without any deviation and unbalancing condition.

**Step 1**: Find error generated by robot.

**Step 2**: Multiply that error by kp parameter and assign it to proportional.

**Step 3**: Integral parameter is added to the multiplication of Ki and error.

**Step 4**: Difference error parameter is added to the multiplication of Kd and difference between current error and previous error.

**Step 5**: Add P, I and D.

**Step 6:** Assign current error to previous error.

**Step 7**: **PID** = P + I + D.

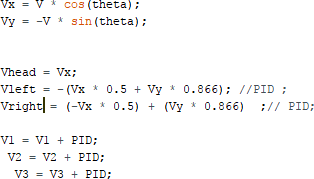
## Path Planning Algorithm:

We have used Omni wheels for holonomic motion. Holonomic means we don’t want to change the head position of robot for that with PID the yaw angle from inertial measurement unit also used. We have set it to 0 degree for holonomic motion.

We used here geometrical equations to decide the motion direction and motion velocity. This makes our robot more efficient than any line follower or any other simple robot.

According to that geometrical Equation the speed and direction of the robot is decided by microcontroller. We don’t need to assign each command for each direction and speed.

## Vector Equations according to Omni wheels:



**Picture 3.2. 3 Vector Equation for Path planning Algorithm.**

# WORKING OF SYSTEM

The Triwheel Omni robot is look like this:



## Picture 3.2. 4 Tri-wheel Omni Robot Chassis

**Working:**

We can move such type of robot without changing its head means we can turn the robot without rotating its front head and this is called as holonomic motion**.** In this robot when we give some commands through Bluetooth module. Arduino check definition of that command in the program and then sends signal to motor driver and motor driver will generate PWM signal in a such a way that it follows holonomic path without deviation.

Path planning algorithm works in a such a way that when velocity of wheel is greater than 0 then it gives command to move the wheel in clockwise direction and if the velocity of wheel from equation is less than 0 then wheel move in anticlockwise direction.(We have mentioned this in our code).

For every direction the algorithm decides speed of each wheel so that the robot will move in holonomic motion without any deviation. Here IMU MPU 6050 helps that algorithm to maintain the 0 degree path by supplying angle values data to the Arduino and then this data is added to the speed of each wheel in the form of velocity of that wheel. In the program we can decide the direction only and speed is decided by the program by using the path planning algorithm.

## Path planning Algorithm – Direction Description

**Note**: Clockwise direction – CLK, Anticlockwise direction- ANTCLK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Direction** | **Front\_wheel** | **Left\_wheel** | **Right\_wheel** |
| 1. | Forward | 0 | CLK | ANTCLK |
| 2. | Backward | 0 | ANTCLK | CLK |
| 3. | Left | ANTCLK | CLK | CLK |
| 4. | Right | CLK | ANTCLK | ANTCLK |
| 5. | Forward-Right | CLK | CLK | ANTCLK |
| 6. | Forward-Left | ANTCLK | CLK | ANTCLK |
| 7. | Backward-Right | CLK | ANTCLK | CLK |
| 8. | Backward-Left | ANTCLK | ANTCLK | CLK |
| 9. | Stop | 0 | 0 | 0 |

## Diagrammatic Representation of Tri-wheel Omni Holonomic motion:

1. **Forward:**

V-front = 0



V-right < 0

V-left > 0

## Backward:

V-front = 0



V-right > 0

V-left < 0

1. **Left:**

V-front <0



V-right > 0

V-left > 0

## Right:

V-front >0



V-right < 0

V-left < 0

1. **Forward-Right:**

V-front >0



V-right < 0

V-left > 0

## Forward-Left:

V-front <0



V-right < 0

V-left > 0

1. **Backward-Right:**

V-front >0



V-right > 0

V-left < 0

## Backward-Left:

V-front < 0



V-right > 0

V-left < 0

1. **Stop:**

V-front =0



V-right =0

V-left =0

# CHAPTER 4 TESTING

## There are two types of testing namely:

1. **Software Testing**

## Hardware testing

1. **Software testing:**

In this method, we implemented whole software testing. When we created any function, we performed debugging operation in order to check the accuracy of that function on Arduino Serial Monitor by simply printing the running and static values of components that we have used in our robot.

## Hardware testing:

All the hardware testing is carried out in the section. Firstly, we carried out the testing of individual components that is whether it is working or not. After the testing, we mounted all the components. Then we write code for Arduino then we tested the overall functionality of the system.

We found the error and corrected it. Hence, we tested the whole system and ensured the reliability of the system. Testing improves the durability, lifetime of the system.

# CHAPTER 5 RESULT AND DISCUSSION

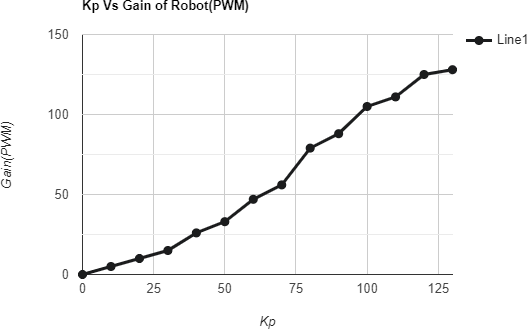
* 1. **Performance graph:**

## KP vs. Gain (PWM) Data:

|  |  |  |
| --- | --- | --- |
| Sr. No. | Kp | Gain (PWM) |
| 1. | 0 | 0 |
| 2. | 10 | 5 |
| 3. | 20 | 10 |
| 4. | 30 | 15 |
| 5. | 40 | 26 |
| 6. | 50 | 33 |
| 7. | 60 | 47 |
| 8. | 70 | 56 |
| 9. | 80 | 79 |
| 10. | 90 | 88 |
| 11. | 100 | 105 |
| 12. | 110 | 111 |
| 13. | 120 | 125 |
| 14. | 130 | 128 |

**Table 5. 1 KP vs. Gain (PWM)**

## Graph: KP vs. Gain (PWM):



**Graph 5. 1 Graph of Kp vs. Gain (PWM)**

# Result and Discussion

## Result:

* 1. Robot can be controlled by the manual mode as well as in voice mode in good way. We can also use the robot in fully automatic mode if we use the DC motors with rotary encoders.
  2. In voice control mode there is some delay in the command of the robot but motion is accurate.
  3. Robot Works perfectly after application of PID algorithm with correct values of parameters of Kp, Kd, and Ki. Robot complete its task in very good manner and all indications like buzzer, LED are working perfectly.
  4. We used two Bluetooth applications which are of size less than 6MB. So the command speed of operation is fast as compared to other applications.
  5. LIPO Battery used in the project has good performance. This battery supply the all required power for the circuit operation.
  6. Bluetooth standard used for the communication purpose so we have some limitation on the range of the operation.
  7. Servo motor positioning is perfectly fine. The servo motor is rotating in the interval of 0 and 180 degree with complete accuracy.
  8. Accuracy of Inertial measurement unit is good. Sometimes it is generating error of 0.75 degrees but we minimized it using PID algorithm by subtracting or adding the error from our set point. Otherwise the IMU MPU6050 works perfectly with I2C interface without any lag or mismatch of values.
  9. Robot performs the desired holonomic motion with almost 0 degree deviation and task to build the holonomic motioned robot is completed successfully.

## Discussion:

In the modern technology, there is increase in use of the robot in different fields. So there also need of the robot which works efficiently in different places in industrial sector as well as the place where automation needs.

According to these demands more research takes place in the robotic field and it is process continuously growing. Now days some prediction is done in robotics field they are related to the artificial intelligent and machine learning.

Machine learning is related to mathematics, geometry and statistics. Our Robot is a combination of Mathematical path planning algorithm and Geometrical trajectory. If we combine our robot with machine learning and AI in future it will be best model that changes all path planning drives and it is affordable also to everyone.

So, in becoming days artificial intelligent take much market in the world, this is implemented everywhere also in robotics field.

## Advantages:

1. The cost of robot is very less as compared to other robots because of path planning algorithm works automatically.
2. Less complex circuit because we have used only 4 electronics components.
3. The development and programming of the robot is done such a way that it remains very user friendly for use to new user.
4. Bluetooth app interface is easy to understand to the buyer or third party user.
5. The robot can be work in any platform. It can be used for inter-transportation, in hotels etc. It can be used everywhere where specific path is required.
6. It is very easy to understand all interface or robot.

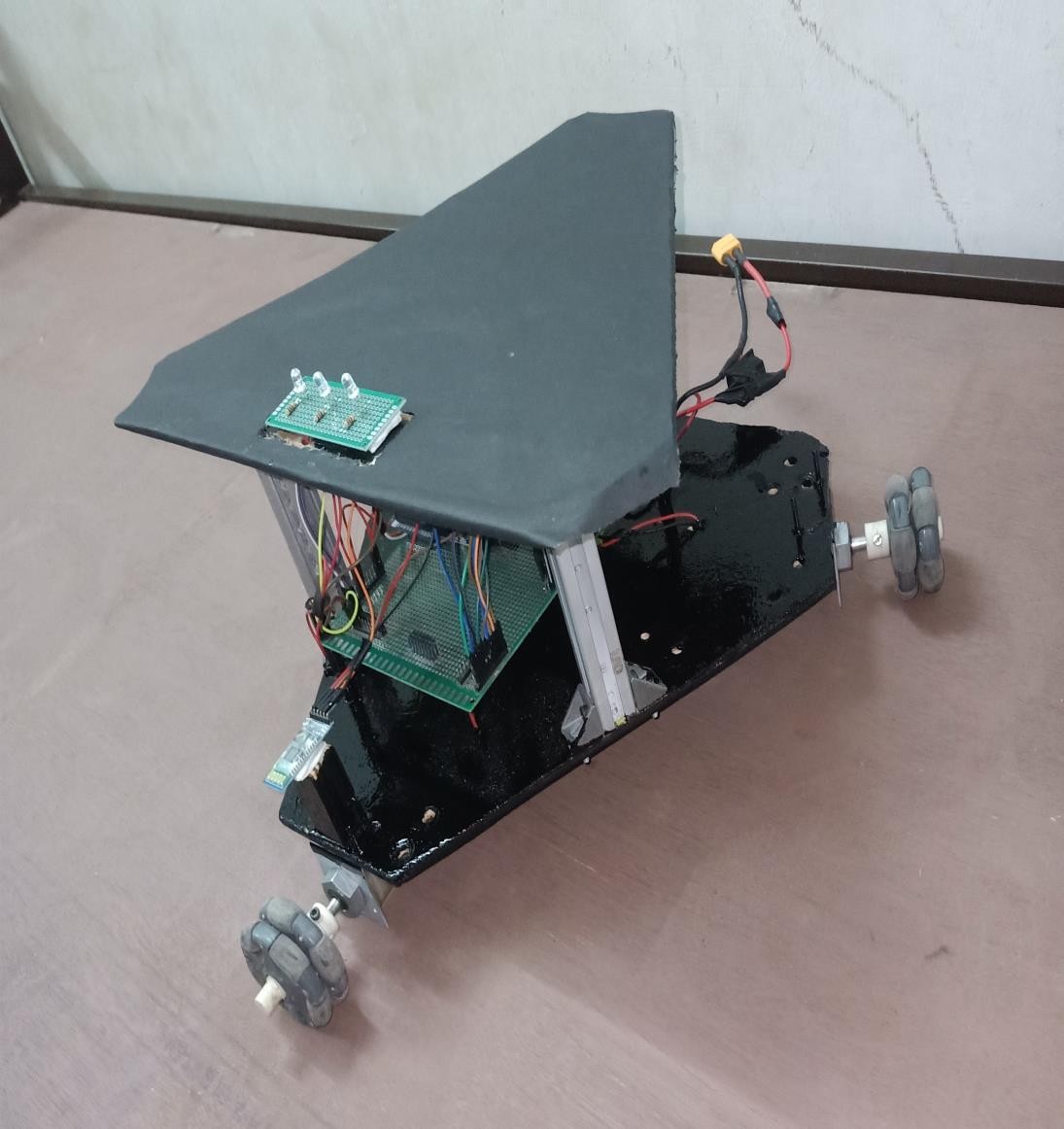
## Drawbacks:

1. Sometimes Bluetooth App takes some time to pass the voice command so motion takes some times to start and stop.
2. Battery Voltage should be in the specified range so that gain of robot is distributed equally and PID algorithm works perfectly.

## Applications:

* + 1. It can be used in transportation in industry and as a packaging and supplying vehicle.
    2. It can be used for research in which path planning algorithm is used.
    3. After implementing image processing and machine learning algorithms it can be used as hybrid model for any type of operation in inter transportation in industry.
    4. It can be used as assistant in IT companies, Hotels and different places where specific path and motion is needed.

## Project Image



**Picture 5. 1 Final Model of Project**

## Estimation Cost:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Components/Material** | **Quantity** | **Cost(INR)** |
| 1. | Arduino MEGA | 1 | 700 |
| 2. | IMU MPU 6050 | 1 | 195 |
| 3. | Bluetooth Module | 1 | 295 |
| 4. | Servo Motor | 1 | 200 |
| 5. | Dc Motor | 3 | 150 |
| 6. | Omni wheels | 3 | 1600 |
| 7. | L298 DC Motor Driver | 2 | 400 |
| 8. | 12V LIPO Battery | 1 | 1400 |
| 9. | Jumper wires | 1 SET | 40 |
| 10. | Double sided PCB | 1 | 75 |
| 11. | Headers | 1 SET | 40 |
| 12. | LED | 1 SET | 20 |
| 13. | Buzzer module | 2 | 120 |
| 14. | Allen Nut and bolts | 10 | 50 |
| 15. | Plywood (700 cm2) | 2 | 100 |
| TOTAL | | 31 | 5385/- ONLY |

**Table 5. 2 - Estimation cost**

## Conclusion:

The goal of this model is to provide an equipment robot that enables individuals to use their voice or android app to control robots or other industrial machinery. The smartphone is nowadays are growing into more and more powerful devices, which have the capacity to interact with other appliances through Bluetooth, Wi-Fi, etc.

Bluetooth being a cheap mode of communication, provide a powerful mode of connection. All our research and projects about controlling devices using voice and manual automation pay off and finally leads us to the conclusion that Yes, it is possible for human beings, small scale industries to control their change automation work through our robot.

## Future scope:

* + 1. As we know nothing in this world is perfect everything is trying to make it better and more effective compared to others.
    2. This technology also requires lots more development. Thus expanding its applications farther where at present we can’t think of.
    3. We can increase the robot working efficiency by introducing the artificial intelligence and machine learning algorithms in it.
    4. India is fastest growing country in fields of technology, banking and automotive, healthcare sectors.
    5. World’s top investors are look forward to invest in Indian sectors so robotics and automotive sectors will be in the huge demand in next year according to one of the magazine.
    6. Our robot will change all the inter transportation work inside the small scale and medium scale industry through our model.
    7. Large scale industries are using this model of Triwheel Omni now a days because its 360 degree rotation and holonomic motion but if we introduce the AI and ML in it they can also use our robot as complete automatic driver robot.

# REFERENCES

1. M. Reza, A. Rastegarpanah and R. Stolkin, "Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments", *Robotics*, vol. 10, no. 48, pp. 1-29, 2021.
2. V. A. Shestakov, I. S. Mamaev and Yu. L. Karavaev, "Controlled motion of a highly maneuverable mobile robot along curvilinear trajectories", *2020 International Conference Nonlinearity Information and Robotics (NIR)*, pp. 1-4, 2020.
3. Hajer Omrane, Mohamed Slim Masmoudi and Mohamed Masmoudi, "Neural controller of autonomous driving mobile robot by an embedded camera", International Conference on Advanced Technologies For Signal and Image Processing - ATSIP, 2018.
4. Shuai Yuan, "Research on Intelligent Car Positioning and Path Planning Technology Based on Ultra-wideband [D]", Xidian University, 2018.
5. A. V. Borisov, A. A. Kilin and I. S. Mamaev, "Dynamics and Control of an Omniwheel Vehicle", *Regular and Chaotic Dynamics*, vol. 20, no. 2, pp. 153-172, 2015.
6. R. Pahuja and N. Kumar, “Android Mobile Phone Controlled Bluetooth Robot,” [www.ijser.in](http://www.ijser.in/) ISSN, vol. 2, no. 7, pp. 2347– 3878, 2014, Using 8051 Microcontroller”.
7. https://howtomechatronics.com/tutorials/arduino/how-to-track-orientation- with-arduino-and-adxl345-accelerometer/

# SOFTWARE PROGRAM

#include<math.h> #include<Wire.h> #include <Servo.h>

/\*\*\*\*\*\*\*\*\*\*IMU declaration\*\*\*\*\*\*\*\*\*\*/ #include <MPU6050.h>

MPU6050 mpu;

Servo myservo;

/\*Motor pins declaration\*/

//head motor

int IN1\_HEAD = 6; // FOR CLK: HIGH

int IN2\_HEAD = 7; // FOR ANTCLK : HIGH int ENA\_HEAD = 5; // SPEED ADJUSTMENT

//left motor

int IN1\_LEFT = 9; int IN2\_LEFT = 10; int ENA\_LEFT = 8;

//right motor

int IN3\_RIGHT = 12; int IN4\_RIGHT = 13; int ENB\_RIGHT = 11;

//

int buzzer1 = 22;

int buzzer2 = 24;

//indication LED int motion = 30; int pick = 32;

int place = 28;

// Timers

unsigned long timer = 0,tp=0,reading = 0; float timestep = 0.01;

// Pitch, Roll and Yaw values float pitch = 0;

float roll = 0; float yaw = 0;

/\*Bluetooth module declaration\*/ char command = 'S' ;

/\*Algorithms function declaration\*/ void Clockwise();

void AntiClockwise(); void V1\_Clockwise(); void V2\_Clockwise(); void V3\_Clockwise(); void V1\_AntiClockwise(); void V2\_AntiClockwise();

void V3\_AntiClockwise();

float theta = 0 \* M\_PI / 180; float V1, V2, V3;

float Vx, Vy;

float V = 225,VL=0,Mpwm=0; float error = 0;

float PID = 0; float P, I, D;

float preverror = 0; float kp = 118; float ki = 0.0001; float kd = 170;

float setpoint = 0.00; int count=0,count1=0; void setup() {

// put your 5setup code here, to run once:s

/\*IMU initialization\*/ Serial.begin(115200); Serial1.begin(9600); Wire.begin();

// Initialize MPU6050

while(!mpu.begin(MPU6050\_SCALE\_2000DPS, MPU6050\_RANGE\_2G))

{

Serial.println("Could not find a valid MPU6050 sensor, check wiring!");

delay(500);

}

// Calibrate gyroscope. The calibration must be at rest.

// If you don't want calibrate, comment this line. mpu.calibrateGyro();

// Set threshold sensivty. Default 3.

// If you don't want use threshold, comment this line or set 0. mpu.setThreshold(3);

// servo myservo.attach(3);

//buzzer

pinMode(buzzer1, OUTPUT); pinMode(buzzer2, OUTPUT);

//led

pinMode(motion, OUTPUT); pinMode(pick, OUTPUT); pinMode(place, OUTPUT);

/\*\*\*\*\*\*\*\*\*\*\*Wheel pinmode \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//HEAD pinMode(IN1\_HEAD,OUTPUT); pinMode(IN2\_HEAD,OUTPUT); pinMode(ENA\_HEAD,OUTPUT);

//RIGHT pinMode(IN1\_LEFT,OUTPUT); pinMode(IN2\_LEFT,OUTPUT); pinMode(ENA\_LEFT,OUTPUT);

//LEFT pinMode(IN3\_RIGHT,OUTPUT); pinMode(IN4\_RIGHT,OUTPUT); pinMode(ENB\_RIGHT,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

/\*IMU definition\*/ timer = millis();

// Read normalized values

Vector norm = mpu.readNormalizeGyro();

// Calculate Yaw

yaw = yaw + norm.ZAxis \* timestep; Serial.print(" Yaw = "); Serial.println(yaw);

// Wait to full timeStep period delay((timestep\*1000) - (millis() - timer));

/\*Bluetooth module commands\*/

if(Serial1.available()>0)

{

command = Serial1.read();

}

Serial.print(" command: "); Serial.print(command);

if (command == 'F')

{

theta = 90 \* M\_PI / 180;

}

if (command == 'B')

{

theta = 270 \* M\_PI / 180;

}

if (command == 'L')

{

theta = 180 \* M\_PI / 180;

}

if (command == 'R')

{

theta = 0 \* M\_PI / 180;

}

if (command == 'G')

{

theta = 135 \* M\_PI / 180;

}

if (command == 'I')

{

theta = 45 \* M\_PI / 180;

}

if (command == 'H')

{

theta = 225 \* M\_PI / 180;

}

if (command == 'J')

{

theta = 315 \* M\_PI / 180;

}

if(command == 'W')

{

count++;

}

if(count ==0)

{

setpoint = 0;

}

if(count==1)

{

setpoint = 60;

}

if(count==3)

{

setpoint = 0;

}

if(count == 2)

{

setpoint = 130;

}

if(command == 'V')

{

tone(buzzer1, 1000); // Send 1KHz sound signal... delay(300); // ...for 0.5 sec

noTone(buzzer1); // Stop sound... delay(300); // ...for 1sec digitalWrite(pick,HIGH); delay(500); digitalWrite(pick,LOW); delay(1000);

servo();

}

if(command == 'v')

{

tone(buzzer2, 1500); // Send 1KHz sound signal... delay(100); // ...for 1 sec

noTone(buzzer2); // Stop sound... delay(100); // ...for 1sec

tone(buzzer2, 1500); // Send 1KHz sound signal... delay(100);

noTone(buzzer2); // Stop sound... delay(100); digitalWrite(place,HIGH); delay(500); digitalWrite(place,LOW); delay(1000);

servo1();

}

if(command == 'U')

{

myservo.write(0); delay(1000); myservo.write(180); delay(3000);

myservo.write(0);

delay(1000);

}

//Yaw PID

error = yaw - setpoint;

P = kp \* error;

I = I + (error \* ki);

D = kd \* (error - preverror); PID = P + I + D;

preverror = error;

Serial.print(" V1: ");

Serial.print(V1);

Serial.print(" V2: ");

Serial.print(V2);

Serial.print(" V3: ");

Serial.print(V3);

Serial.print("V: ");

Serial.print(V);

Vx = V \* cos(theta); Vy = -V \* sin(theta); V1 = Vx;

V2 = -(Vx \* 0.5 + Vy \* 0.866); //PID ;

V3 = (-Vx \* 0.5) + (Vy \* 0.866) ;// PID; V1 = V1 + PID;

V2 = V2 + PID;

V3 = V3 + PID;

if( command == 'S')

{

V1 = 0+PID; V2 = 0+PID; V3 = 0+PID;

}

if (V1 >= 0)

{

V1\_Clockwise();

}

else

{

V1\_AntiClockwise();

}

if (V2 >= 0)

{

V2\_Clockwise();

}

else

{

V2\_AntiClockwise();

}

if (V3 >= 0)

{

V3\_Clockwise();

}

else

{

V3\_AntiClockwise();

}

if(count>3) count = 0; if(count1>2) count1 = 0;

}

void V1\_Clockwise()

{

analogWrite(ENA\_HEAD, abs(V1)); digitalWrite(IN1\_HEAD, HIGH); digitalWrite(IN2\_HEAD, LOW);

}

void V1\_AntiClockwise()

{

analogWrite(ENA\_HEAD, abs(V1));

digitalWrite(IN1\_HEAD, LOW); digitalWrite(IN2\_HEAD, HIGH);

}

void V2\_Clockwise()

{

analogWrite(ENA\_LEFT, abs(V2)); digitalWrite(IN1\_LEFT, HIGH); digitalWrite(IN2\_LEFT, LOW);

}

void V2\_AntiClockwise()

{

analogWrite(ENA\_LEFT, abs(V2)); digitalWrite(IN1\_LEFT, LOW); digitalWrite(IN2\_LEFT, HIGH);

}

void V3\_Clockwise()

{

analogWrite(ENB\_RIGHT, abs(V3)); digitalWrite(IN3\_RIGHT, HIGH); digitalWrite(IN4\_RIGHT, LOW);

}

void V3\_AntiClockwise()

{ analogWrite(ENB\_RIGHT, abs(V3)); digitalWrite(IN3\_RIGHT, LOW); digitalWrite(IN4\_RIGHT, HIGH);

}

void servo()

{ // waits 15 ms for the servo to reach the position

myservo.write(0); // tell servo to go to position in variable 'pos' delay(1000);

}

void servo1()

{

// waits 15 ms for the servo to reach the position myservo.write(0); // tell servo to go to position in variable 'pos' delay(1000);

myservo.write(180); // tell servo to go to position in variable 'pos' delay(3000);

}

## DATASHEETS

