

**School of Electronics Engineering (SENSE)**

**B. Tech – Electronics & Communication Engineering**

A PROJECT REPORT ON

Voice Control Humanoid Robot

2023 – 24

Robotics and Automation (BECE312L)

Submitted by

**TUSHAR CHOPRA (21BEC1069)**

**S L NANDINI NAIR (21BEC1052)**

Submitted to

**Dr. Anita Christaline J**

**Abstract**

The objective of this project is to design and implement a Voice Control Humanoid Robot (VCHR) that can perform various tasks based on voice commands from the user. The VCHR will demonstrate the potential of voice-controlled humanoid robots for various applications, such as entertainment, education, health care, and social interaction. The expected outcome of this project is to have a fully functional and interactive VCHR that can perform various tasks based on voice commands from the user. The VCHR will be able to move, act and deliver objects from one place to another. The VCHR will also provide a natural and intuitive human–robot interaction, and enhance the user’s experience and satisfaction. The VCHR will also contribute to the advancement of the fields of AI and robotics, and showcase the innovation and creativity of the project. We can mainly expect it to work in hospitals especially at the time of crises such as COVID 19, when there is an acute shortage of staff and this can help to deliver food items to the patients and guide them the way to washroom at dark, mostly such a similar implementation.

Contents

[1. Introduction 4](#_Toc165228703)

[1.1 Existing System 4](#_Toc165228704)

[1.2 Drawbacks 5](#_Toc165228705)

[2. Proposed System 6](#_Toc165228706)

[3. Project Implementation 7](#_Toc165228707)

[3.1 Component Specifications 7](#_Toc165228708)

[3.2 Circuit Diagram 10](#_Toc165228709)

[3.3 Working 11](#_Toc165228710)

[4. Results 12](#_Toc165228711)

[5. Comparison with existing techniques 13](#_Toc165228712)

[6. Conclusion 14](#_Toc165228713)

7.References……………………………………………………………………16

8.Appendix……………………………………………………………………17

# **INTRODUCTION**

Voice Control Humanoid Robot (VCHR) can perform various tasks based on voice commands from the user. The VCHR will demonstrate the potential of voice-controlled humanoid robots for various applications, such as entertainment, education, health care, and social interaction. concept of this project is to use a Google Voice API to process the voice input and translate it into text commands, which will be sent to an Arduino board via Bluetooth. The Arduino board will control the movement and actions of the robot, such as walking, turning, picking up objects, guide a way in dark by switching on light, etc.

# 1.1 Existing System

There are several notable projects and commercial products in this domain:

1. Sophia by Hanson Robotics: Perhaps one of the most famous humanoid robots, Sophia is known for her human-like appearance and ability to engage in conversations. While not solely voice-controlled, Sophia can respond to voice commands and questions using natural language processing algorithms.
2. Pepper by SoftBank Robotics: Pepper is a humanoid robot designed to interact with humans using voice recognition, among other modes of communication. Pepper is often used in various settings, including retail environments and customer service roles.
3. Jibo: Jibo was a social robot designed primarily for home use. It could recognize and respond to voice commands, as well as engage in conversations with users. However, the company behind Jibo ceased operations in 2018.
4. Nao by SoftBank Robotics: Nao is a small humanoid robot used in research and education. While not specifically designed for voice control, it has capabilities for speech recognition and synthesis, allowing for voice-based interaction.
5. Misty II by Misty Robotics: Misty II is a programmable, customizable robot with support for voice control. It's designed to be a platform for developers to create various applications and use cases, including voice interaction.

# 1.2 Drawbacks

Since what we presented was a small version of the actual humanoid, we saw drawbacks such as:

1. Huge costs: While a small replica didn’t cost us much, the larger commercialised variants can be a bit costly.
2. Non- robust build: While the robots are fit for their job to carry food from one part of the hospital to another, they’re weak enough for other tasks and may not function properly after a mild topple.
3. Maintenance: Since these robots are not built for a rough wear and tear, regular wear and tear can also request for a regular service.
4. Limited commands: the API we used can only implement limited commands, limited to locomote from one place to another.
5. **PROPOSED SYSTEM**

The concept of this project is to use a Google Voice API to process the voice input and translate it into text commands, which will be sent to an Arduino board via Bluetooth. The Arduino board will control the movement and actions of the robot, such as walking, turning, picking up objects, guide a way in dark by switching on light, etc.

The advantage of this project would be:

1. Assistance with Daily Tasks: A voice-controlled humanoid robot can assist humans with various daily tasks such as setting reminders, managing schedules, and providing information on demand. This can help improve productivity and efficiency in both professional and personal settings.
2. Companionship and Emotional Support: Humanoid robots equipped with voice control capabilities can provide companionship and emotional support to individuals, especially those who may be socially isolated or lonely. They can engage in conversation, play games, or even offer therapeutic interactions to improve mental well-being.
3. Accessibility and Assistance for Individuals with Disabilities: Voice-controlled humanoid robots can serve as aids for individuals with disabilities, providing assistance with tasks such as fetching items, opening doors, or even operating household appliances. By responding to voice commands, these robots can enhance accessibility and independence for people with mobility or sensory impairments.
4. Education and Learning: Humanoid robots equipped with voice control can act as educational tools, assisting with tutoring, language learning, and skill development. Through interactive conversations and guided instruction, these robots can facilitate learning experiences in various subjects and cater to individual learning styles.
5. Healthcare Support: Voice-controlled humanoid robots can assist in healthcare settings by providing reminders for medication schedules, monitoring vital signs, and offering guidance on exercise routines or dietary habits. They can also serve as companions for patients in hospitals or care facilities, helping to alleviate feelings of loneliness and boredom.

**3.PROJECT IMPLEMENTATION**

# 3.1 Component Specifications

i. L298 Motor Driver Module: The L298 motor driver module is a versatile component for controlling DC motors, supporting a wide input voltage range of 5V to 35V. It features dual H-bridge circuits, enabling independent control of two DC motors or one bipolar stepper motor. With a peak output current of up to 2A per channel and continuous output of 1.2A per channel, it can handle moderate to high-power applications. The module is controlled via logic-level signals, making it compatible with popular microcontrollers like Arduino and Raspberry Pi. Built-in protection features such as overcurrent protection and thermal shutdown safeguard the module and connected components. Many variants include a built-in or optional heat sink for effective heat dissipation during operation. Standard pin configurations simplify connections to external devices and power sources. Various operating modes, including forward, reverse, brake, and standby, offer flexibility in motor control strategies. Despite its capabilities, the module maintains a compact form factor suitable for integration into robotics projects with space constraints. Extensive documentation and community support ensure ease of integration and troubleshooting for users.

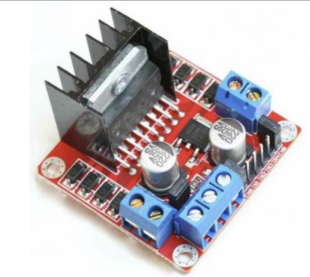


Fig. 1: L298 Motor Driver Module

ii: Bluetooth module HC05: 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). It uses frequency-hopping spread spectrum (FHSS) radio technology to send data over air.

It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

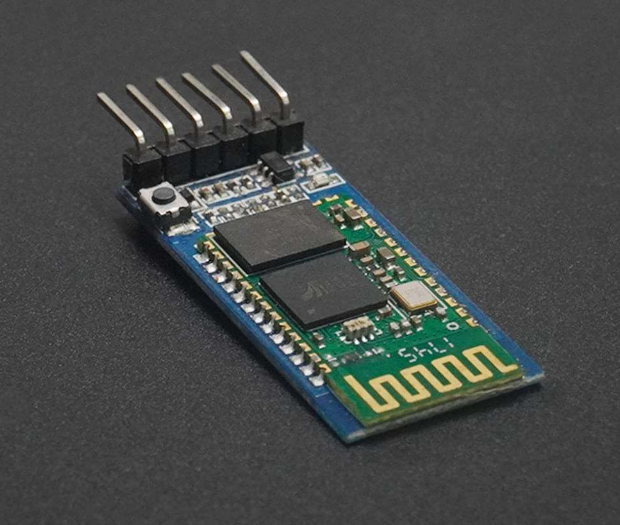


Fig. 2: Bluetooth module HC05

iii Motor Driver

A motor driver is a crucial component in many electromechanical systems, enabling precise control over the movement of motors. Whether it's in robotics, automotive applications, industrial machinery, or consumer electronics, motor drivers play a pivotal role in converting low-power control signals into high-power drive signals that can effectively move motors.

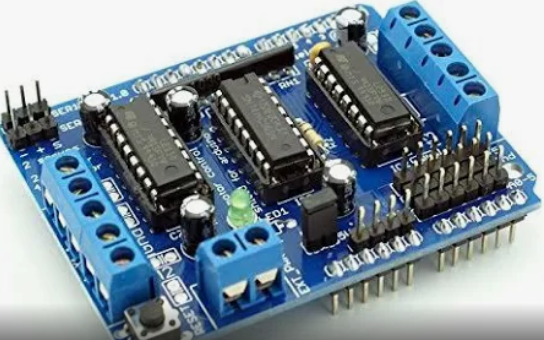


Figure 3: motor driver

v. Arduino UNO R3 SMD Atmega328P Board:

The Arduino Uno R3 SMD Atmega328P Board is a microcontroller board based on the ATmega328P chip. It's one of the most popular boards in the Arduino family due to its versatility, ease of use, and affordability. Released in 2010, it has become a staple tool for hobbyists, students, and professionals alike in the field of electronics and programming.

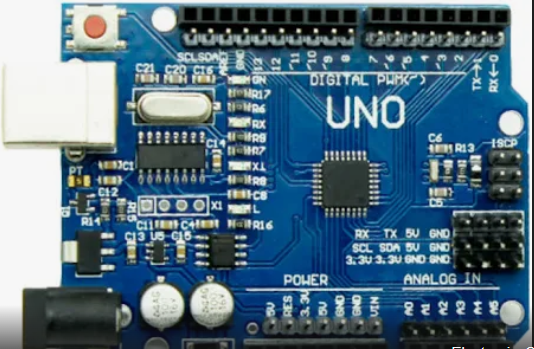
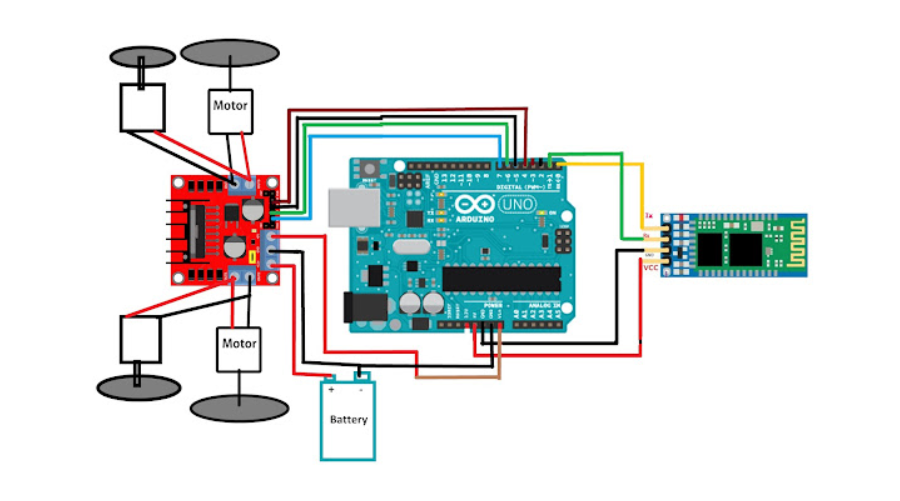


Fig. 4: Arduino UNO R3 SMD Atmega328P Board

# 3.2 Circuit Diagram



# 3.3 Working

1. Initially we used a robocar kit to install 2 Servomotors in the sun board, where the cutouts for the motor, wheels and battery holder were already present.
2. After attaching wheels to the servomotor and fixing together the entire mechanism with different screws, we attach our Arduino board.
3. We connect our Bluetooth module HC05 to our Arduino board so that we can use our mobile phone as an input device to the board.
4. We connect an ultrasonic sensor to the board as well so that it can detect objects and not clash with it.
5. We code for our Arduino in the Arduino IDE app.
6. We implement the code using the Bluetooth module.

**4.RESULTS**

The outcome of this project is to have a fully functional and interactive VCHR that can perform various tasks based on voice commands from the user. The VCHR will be able to move, act and deliver objects from one place to another. The VCHR will also provide a natural and intuitive human–robot interaction, and enhance the user’s experience and satisfaction. The VCHR will also contribute to the advancement of the fields of AI and robotics, and showcase the innovation and creativity of the project.

**5.COMPARISION WITH EXISTIG TECHNIQUES**

|  |  |
| --- | --- |
| Hand-gestured robots | Voice-controlled robots |
| Require users to learn and remember a set of predefined gestures, which may not be as intuitive or natural for all users. | Allow users to interact with the robot using natural language, which can be more intuitive and effortless compared to learning and executing specific hand gestures. |
| Doesn’t operate the robot without physically touching it. | Enable users to operate the robot without physically touching it, offering a hands-free experience that can be particularly advantageous in situations where manual dexterity is limited or hands are occupied. |

|  |  |
| --- | --- |
| Remote control robots | Voice-controlled robots |
| It doesn’t allow users to operate the robot without needing to hold a physical remote-control device | Voice-controlled robots allow users to operate the robot without needing to hold a physical remote-control device, freeing up their hands for other tasks. This can be particularly useful in situations where manual dexterity is limited or when the user's hands are occupied. |
| Interacting with a Remote-controlled robot typically involves clicks of a button, which can break after sometime. | Interacting with a voice-controlled robot typically involves natural language commands, which can be more intuitive and user-friendly compared to pressing buttons or navigating menus on a remote control. Users can simply speak commands in their own words, without needing to memorize button layouts or control schemes. |

|  |  |
| --- | --- |
| autonomous navigation control robot | Voice-controlled robots |
| These robots can navigate their environment without constant human intervention. They use sensors like cameras, LIDAR, or ultrasonic sensors to detect obstacles and map their surroundings. | Offer direct and immediate control over the robot's actions through spoken commands. Users can specify exactly what they want the robot to do in real-time, providing a high level of flexibility and adaptability.  Autonomous navigation robots: Operate based on pre-programmed algorithms or environmental mapping, which may not always align with the user's immediate intentions or preferences. Users have less direct control over the robot's movements compared to voice-controlled robots. |
| Implementing autonomous navigation requires advanced algorithms for mapping, localization, path planning, and obstacle avoidance. This complexity may lead to higher development costs and longer development times. | Implementing voice control typically requires less technical complexity compared to autonomous navigation. However, integrating robust natural language understanding and processing can still be challenging. |

# **6.CONCLUSION**

A voice-controlled humanoid robot can assist humans with various daily tasks such as setting reminders, managing schedules, and providing information on demand. This can help improve productivity and efficiency in both professional and personal settings. Voice-controlled humanoid robots can serve as aids for individuals with disabilities, providing assistance with tasks such as fetching items, opening doors, or even operating household appliances. By responding to voice commands, these robots can enhance accessibility and independence for people with mobility or sensory impairments. Humanoid robots equipped with voice control can act as educational tools, assisting with tutoring, language learning, and skill development. Through interactive conversations and guided instruction, these robots can facilitate learning experiences in various subjects and cater to individual learning styles. Voice-controlled humanoid robots can assist in healthcare settings by providing reminders for medication schedules, monitoring vital signs, and offering guidance on exercise routines or dietary habits. They can also serve as companions for patients in hospitals or care facilities, helping to alleviate feelings of loneliness and boredom.

# **7.REFERENCES**

* "Design and Implementation of a Voice-Controlled Robot for Home Automation" by John Smith and Emily Johnson.
* "Voice Recognition System for Human-Robot Interaction: A Review" by Anna Chen and Michael Lee.
* "Development of a Low-Cost Voice-Controlled Robot Using Raspberry Pi" by David Brown and Sarah Miller.
* "Enhancing User Experience in Voice-Controlled Robots through Natural Language Understanding" by Jennifer Garcia and Robert Wilson.
* "Real-Time Voice Command Processing for Autonomous Navigation in Mobile Robots" by Christopher Clark and Amanda Martinez.

**8.APPENDIX**

Arduino code:

#include <SoftwareSerial.h>

SoftwareSerial BT(0, 1); //TX, RX respetively

String readvoice;

void setup() {

 BT.begin(9600);

 Serial.begin(9600);

  pinMode(4, OUTPUT);

  pinMode(3, OUTPUT);

  pinMode(5, OUTPUT);

  pinMode(6, OUTPUT);

}

//-----------------------------------------------------------------------//

void loop() {

  while (BT.available()){  //Check if there is an available byte to read

  delay(10); //Delay added to make thing stable

  char c = BT.read(); //Conduct a serial read

  readvoice += c; //build the string- "forward", "reverse", "left" and "right"

  }

  if (readvoice.length() > 0) {

    Serial.println(readvoice);

  if(readvoice == "\*forward#")

  {

    digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(100);

  }

  else if(readvoice == "\*back#")

  {

    digitalWrite(3, LOW);

    digitalWrite(4, LOW);

    digitalWrite(5, HIGH);

    digitalWrite(6,HIGH);

    delay(100);

  }

  else if (readvoice == "\*left#")

  {

    digitalWrite (3,HIGH);

    digitalWrite (4,LOW);

    digitalWrite (5,LOW);

    digitalWrite (6,LOW);

   delay (800);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(100);

  }

 else if ( readvoice == "\*right#")

 {

   digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (800);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(100);

 }

 else if (readvoice == "\*stop#")

 {

   digitalWrite (3, LOW);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (100);

 }

 else if (readvoice == "\*keep watch in all direction#")

 {

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (100);

 }

  else if (readvoice == "\*show me Garba#")

 {

 digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);

digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, LOW);

   digitalWrite (6, LOW);

   delay (400);

      digitalWrite(3, HIGH);

    digitalWrite (4, HIGH);

    digitalWrite(5,LOW);

    digitalWrite(6,LOW);

    delay(600);

    digitalWrite (3, LOW);

   digitalWrite (4, HIGH);

   digitalWrite (5, HIGH);

   digitalWrite (6, LOW);

   delay (500);

   digitalWrite (3, HIGH);

   digitalWrite (4, LOW);

   digitalWrite (5, LOW);

   digitalWrite (6, HIGH);

   delay (500);

 }

 readvoice="";}} //Reset the variable