A Project Report On

VOICE CONTROLLED ROBOT

Submitted in partial fulfillment of the

Requirements for the award of the

Degree of

BACHELOR OF TECHNOLOGY

IN

Electronics &communication engineering

By

M.SAI SRI MOUNIKA(170040588)

N.SRAVYA(170040598)

Y.VASANTHI(170040966)

CONTROL SYSTEMS AND INTRODUCTION TO ROBOTICS

(17EC3631)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION

ENGINEERING

KLEF

SEPTEMBER 2019

**CERTIFICATE**

****

This is to certify that the project based laboratory report entitle “VOICE CONTROLLED ROBOT” submitted by M.SAI SRI MOUNIKA(170040588),N.SRAVYA(170040598) and Y.VASANTHI(1700040966) to the “Department of ELECTRONICS AND COMMUNICATION ENGINEERING”, KL University in partial fulfilment of the Requirements for the completion of a project based report in **“CONTROL SYSTEMS AND INTRODUCTION TO ROBOTICS”** course in B Tech 3rd year 1stsemester, is a bonafide record of the work carried out by him under my supervision during the academic year 2019-2020.

PROJECT SUPERVISOR HEAD OF THE DEPARTMENT

**ACKNOWLEDGEMENT**

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana** , for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our principal **Dr. K. SUBBA RAO** for his administration towards our academic growth.

I record it as my privilege to deeply thank our pioneer **Dr. N.V.SUMAN** HOD-ECE for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **Dr.S.ROOBAN AND CH.SANTHOSH** for their novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

M.SAI SRI MOUNIKA(170040588)

N.SRAVYA(170040598)

Y.VASANTHI(170040966)

**ABSTRACT**

Voice Controlled Robot (VCR) is a mobile robot whose motions can be controlled by the user by giving specific voice commands. The speech is received by a microphone and processed by the voice module. When a command for the robot is recognized, then voice module sends a command message to the robot’s microcontroller. The microcontroller analyzes the message and takes appropriate actions. The objective is to design a walking robot which is controlled by servo motors. When any commands are given on the transmitter, the EasyVR module will take the voice commands and convert the voice commands into digital signals. Then these digital signals are transmitted via ZIGBEE module to the robot. On the receiver side the other ZIGBEE module receives the command from the transmitter side and then performs the respective operations. The Hardware Development board used here is ATmega 2560 development board. In ATmega 2560 there are 15 PWM channels which are needed to drive the servo motors. Addition to this there is camera which is mounted in the head of the robot will give live transmission and recording of the area. The speechrecognition circuit functions independently from the robot’s main intelligence [central processing unit (CPU)]. This is a good thing because it doesn’t take any of the robot’s main CPU processing power for word recognition. The CPU must merely poll the speech circuit’s recognition lines occasionally to check if a command has been issued to the robot. The software part is done in Arduino IDE using Embedded C. Hardware is implemented and software porting is done.

**CONTENTS**

S.NO TITLE PAGE.NO

1 Introduction 1-2

2 Circuit diagram 3

3 Components 4-5

4 Working principle 6

5 Construction 7

6 Applications 8

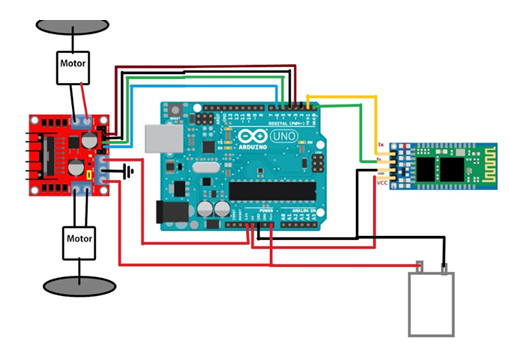
7 Conclusion 9

8 Reference 10

**INTRODUCTION**

When we say voice control, the first term to be considered is Speech Recognition i.e., making the system to understand human voice. Speech Recognition is a technology where the system understands the words (not its meaning) given through speech. Speech is an ideal method for robotic control and communication. The speech recognition circuit we will outline, functions independently form the robot’s main intelligence [central processing unit (CPU)]. This a good thing because it doesn’t takes any of the robots main CPU processing power for word recognition. The CPU must merely poll the speech circuit’s recognition lines occasionally to check if a command has been issued to the robot. We can even improve upon this by connecting the recognition line to one of the robot’s CPU interrupt lines. By doing this, a recognized word would cause an interrupt, letting the CPU know a recognized word had been spoken. The advantage of using an interrupt is that polling the circuit’s recognition line occasionally would no longer be necessary, further reducing any CPU overhead. Another advantage to this stand-alone speechrecognition circuit (SRC) is its programmability. You can program and train the SRC to recognize the unique words you want recognized. The SRC can be easily interfaced to the robot’s CPU. To control and command an appliance (computer, VCR, TV security system, etc.) by speaking to it, will make it easier, while increasing the efficiency and effectiveness of working with that device. At its most basic level speech recognition allows the user to perform parallel tasks, (i.e. hands and eyes are busy elsewhere) while continuing to work with the computer or appliance. Robotics is an evolving technology. There are many approaches to building robots, and no one can be sure which method or technology will be used 100 years from now. Like biological systems, robotics is evolving following the Darwinian model of survival of the fittest. Suppose you want to control a menu driven system. What is the most striking property that you can think of? Well the first thought that came to our mind is that the range of inputs in a menu driven system is limited. By using a menu all we are doing is limiting the input domain space.

**CIRCUIT DIAGRAM**



Components used

1:-arduino uno

2:-bluetooth hc-05

3:-l293d motor driver(even use as l293d ic)

4:-two wheels and caster wheel

5:-two dc motors

6:-jumper wires

7:-mini breadboard

8:-9v battery(power bank)

9:-battery jack

## Working principle:

If you make all connections correctly then download **AMR\_Voice** application from Play Store. Open the application, power up the robot and connect the application to the Bluetooth module. When we speak the specified commands the application sends the instruction to the Arduino through Bluetooth and then Arduino performs the defined operation.

Always disconnect the Rx and Tx pins of Bluetooth module when uploading code to Arduino otherwise it shows some errors and code does not upload. Try “1234” or “0000” password when you connect Bluetooth module with your mobile for the first time. If the wheels are rotating in opposite direction just swaps the connection of the motor from the Motor Driver Shield terminal. Controlling robot through voice is not an efficient way. It is fun to control but I recommend the Gesture control application.

at 1st

in our project we use as l293d ic but you can use motor driver module thats can easily to make...

so in l293d ic 2,7,10 and 15 are pins as input thats are connected to arduino pin...

in our programming we use this pins

2pin of l293d connect to 8pin of arduino(for left motor 3and 6pin of l293d).

7pin of l293d connect to 4pin of arduino(for left motor 3 and 6pin of l293d).

10pin of l293d connect to 7pin of arduino(for right motor 11 and 14 of l293d).

15pin of l293d connect to 3pin of arduino(for right motor 11 nad 14 of l293d).

+5v of l293d are connet to 9v battery to mini breadboard..

bluetooth module:-

vcc-----------+5v

gnd-----------gnd

tx---------------10pin of arduino

rx---------------11 pin of arduino..

## ADVANTAGE S:

* The robot is small in size, so space required for it is small.
* Cost of system is low as we are using smart phone which is nearby available to everyone.
* Speech Recognition works best if the microphone is close to the user (e.g. in a phone, or if the user is wearing a microphone). More distant microphones (e.g. on a table or wall) will tend to increase the number of errors.

## DISADVANTAGES:

* Speech Recognition works best if the microphone is close to the user will tend to increase the number of errors.
* Even the best speech recognition systems sometimes make errors. If there is noise or some other sound in the room the number of errors will increase.

## APPLICATIONS:

* The robot is useful in places where humans find difficult to reach but human voice reaches. E.g. in fire situations, in highly toxic areas.
* It is the one of the important stage of Humanoid robots.
* Command and control of appliances and equipment
* Telephone assistance systems
* Speech and voice recognition security systems
* The robot can be used for surveillance or reconnaissance.

**CONCLUSION**

* The proposed system shows how the android smartphone can be used as remote controller for robot and various embedded technologies with the help of the Bluetooth technology. At the same time, this program uses blue-tooth connection to communicate with robot. The proposed system also shows that how a robot can be used for travelling purpose. The operating system of smartphone is Android, and it can develop effective remote control program and by using WiFi wireless network, the communication between smartphone and robot can be realized, which makes it simple and convenient to control robot.

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

**References:**

[1] http://www.bluetooth.com/Bluetooth/technology/work

[2] <http://www.robokits.co.in>

[3] <http://www.robokitsworld.com>

[4] <http://www.arduino.cc/en/Main/arduinoBoardUno>

[5] http://atmel.in