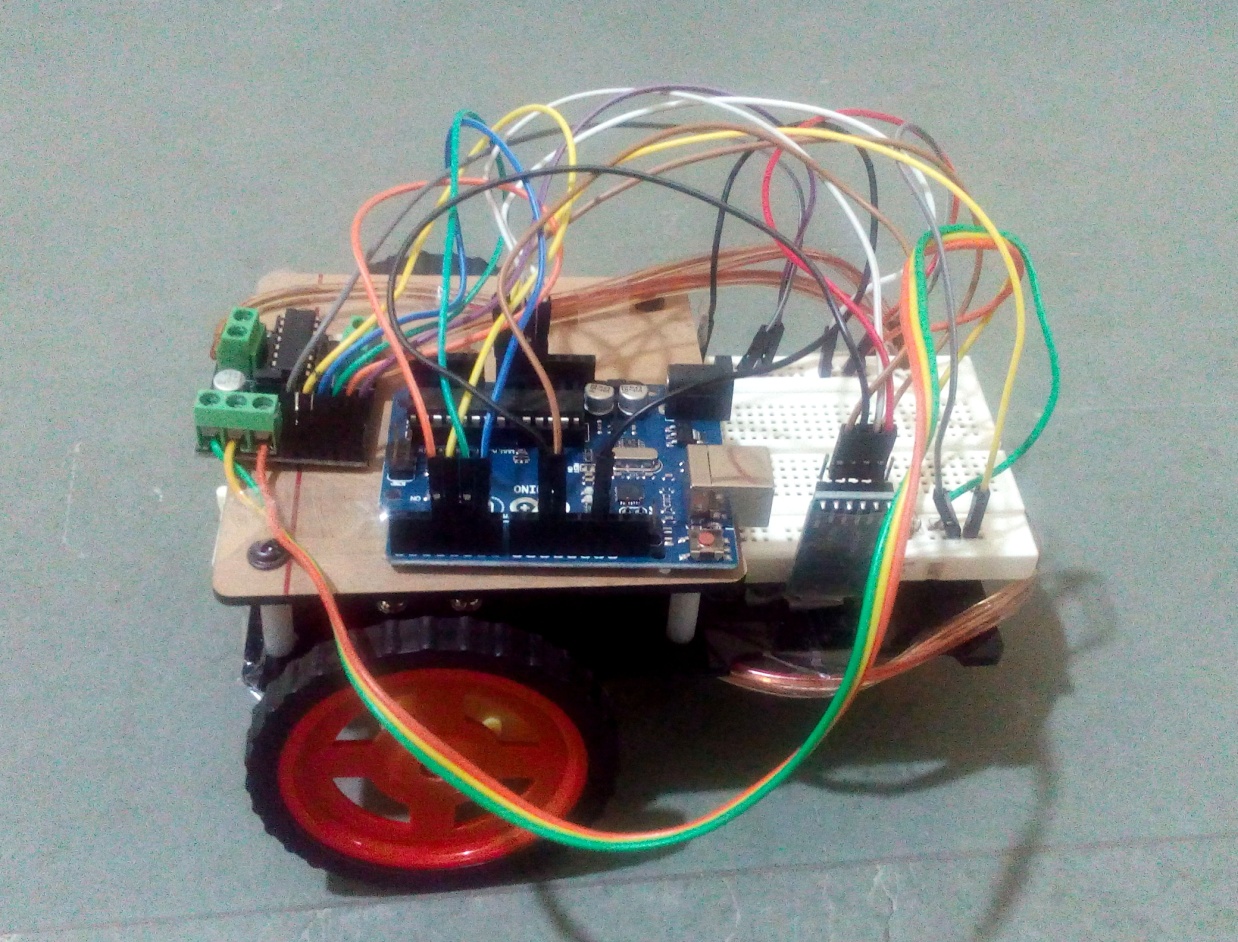
**VOICE CONTROLLED ROBOT**

[DIGITAL LOGIC CIRCUITS PROJECT]

Project Documentation

By: Antra Tripathi

B.tech 1st yr

**Introduction:**

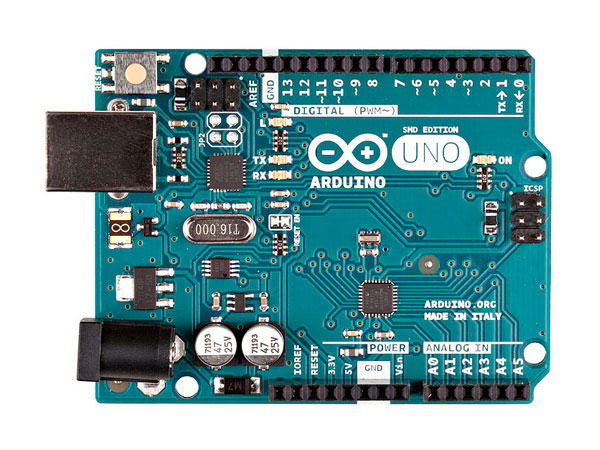
This project ‘Voice Controlled Robot’ is based on digital logic and circuit analysis.It mainly works on voice commands. The voice input message is converted into mechanical movements (forward, backward, left, right etc.) using various components listed below.

**Components Required:**

* Arduino Uno
* Bluetooth module – HC-05
* Motor Driver IC – L293D
* DC Motor X 2
* Wheels X 2
* Chassis X 1
* Jumper Wires
* Bread Board
* Battery 9V

1. **Arduino UNO**

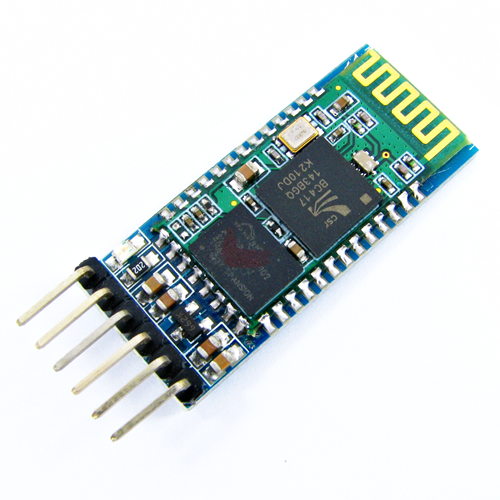
It is an open source Atmel based microcontroller board for development and programming purposes. It has 14 digital input/output pins, of which 6 are PWM based outputs, 6 analog based. It has 16 MHz quartz crystal, a USB, a led, a power jack, an ICSP header and a reset button. A switch and a sensor could be a digital and an analog imput respectively. Any object we want to turn on and off and control could be an input. I t could be a motor or even a computer. Inputs can be processed for the desired output. For programming of the microcontroller, Arduino provides the IDE based on language ‘Processing’ similar to C/C++.



1. **Bluetooth module HC-05**

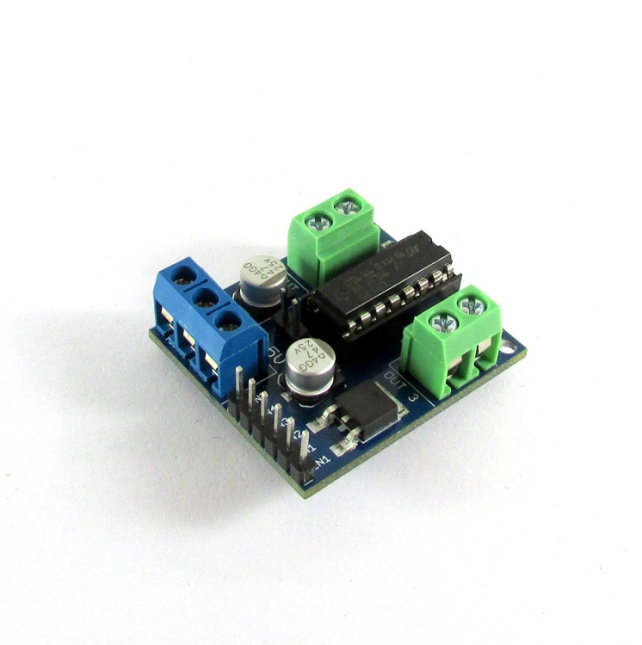
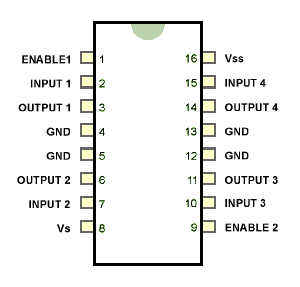
It is a Serial Port Protocol module, required for wireless transmission of information between the android app and the robot. This module, in our application, is set to communicate at 9600 baud (i.e. information transmission at the rate of 9600 bits per second). Default factory setting is Slave. Slave modules cannot initiate a connection to another Bluetooth devices, where as Master modules can do that either way. Module configuration (as Master or Slave) can be done via AT commands. Ir is a six pin device:

* Enable: Controls device. Off when low, high if left open or connected to 3.3V.
* Vcc: Supply Voltage of 3.3V to 5V.
* Gnd : Ground
* TXD and RXD: Transmission and Reception pins respectively.
* State: It acts as a status indicator. Indicates if the Bluetooth is connected or disconnected state.



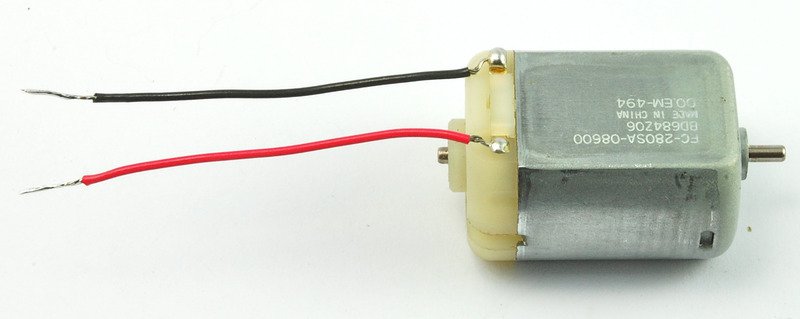
1. **Motor Driver IC – L293D**

A 16-pin IC used to drive the gear motors of the robot. It can be configured to control the set of two motors simultaneously in any direction. There are two H-Bridge circuits in this ID, which forms the working principle. There are four inputs, two on either side. Left pins rotate the motor in left direction and right pins to the right. Based on the input logic as high (1) or low (0), this IC rotates the motor accordingly.

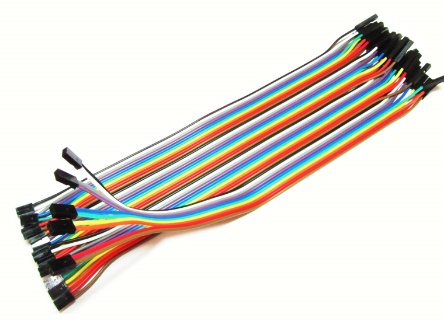
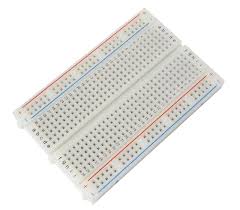


1. **DC Motors**

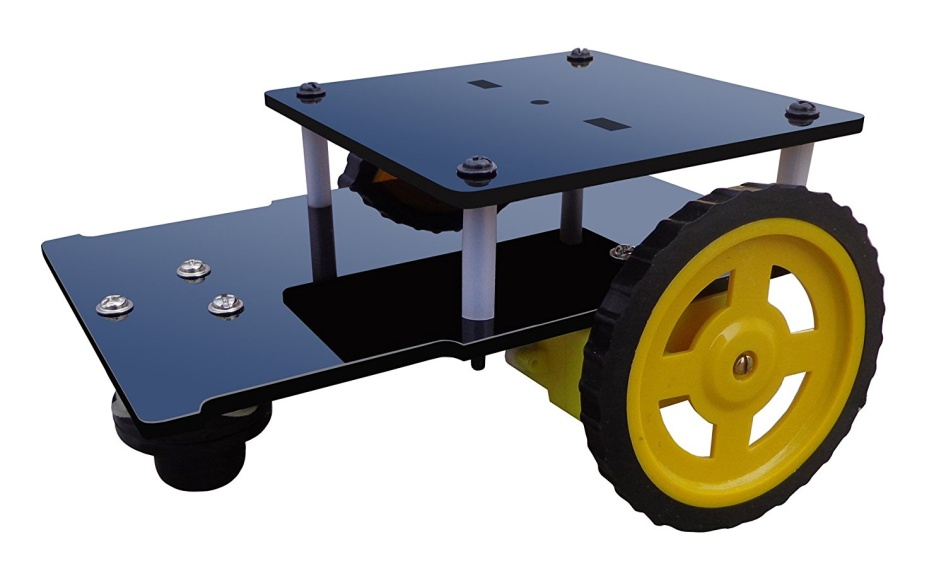
Required to drive the wheels



1. **Jumper wires and Breadboard**



1. **Robotic Chassis**

****

**Features of this Voice Controlled Robot:**

This robot has nine functionality to execute according to the voice commands:

1. Forward

Moves in Forward direction until you stop.

1. Reverse

Moves in Backward direction until you stop.

1. Left

Turns left at 90 degree and stops.

1. Right

Turns right at 90 degree and stops.

1. Right Forward

Turns Right and moves in forward direction until you stop.

1. Left Forward

Turns Left and moves in forward direction until you stop.

1. Right Rotation

Rotates in right direction i.e. continuously changes its direction to right from current position, hence rotates.

1. Left Rotation

Rotates in left direction i.e. continuously changes its direction to left from current position, hence rotates.

1. Stop

Comes to rest.

**Explanation of the connections and working of the Bot**

We have made a voice controlling robot using Arduino and Bluetooth module HC-05. We have made an app called voice bot, through this app we connect our phone Bluetooth to the Bluetooth module HC-05 and various command are given through this app. The Bluetooth module HC-05 consist of 6 pins –

State – describes the state of the module whether it is connected to any device or not.

Receiver- this pin is connected to the 11th i/o pin of the Arduino

Transmitter – this pin of the module is connected to the 10th pin of the Arduino.

Ground- connected to ground(0v)

VCC – connected to 5v

We have chosen the 3,4,5,6 pin as output pin of the Arduino Uno which is connected as an input to the motor driver L293D to drive the dc – motors, the connections are as follow –

ARDUINO UNO CONNECTION WITH DC MOTOR DRIVER (L293D)

1. PIN 3- CONNECTED TO INPUT 1
2. PIN 4- CONNECTED TO INPUT 4
3. PIN 5- CONNECTED TO INPUT 2
4. PIN 6- CONNECTED TO INPUT 3

First VCC of the dc- motor driver is connected to the 5v of the Arduino and second VCC is given the power supply of 12v by the power supply that is the battery. Both the output 1 and output 2 of the dc- motor driver is connected to 1st dc motor and output 3 and output 4 of the dc motor is connected to the 2nd dc motor to drive it in forward or backward directions.

Explanations how and what are the commands given to drive the voice controlling robot. Given below are the command and inputs on which the voice controlling robot works.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INPUT 1** | **INPUT 2** | **INTPUT 3** | **INPUT 4** | **DC MOTOR 1** | **DC MOTOR 2** |
| LOW | HIGH | LOW | HIGH | FWD | FWD |
| HIGH | LOW | HIGH | LOW | REV | REV |
| LOW | HIGH | LOW | LOW | FWD | STOP |
| LOW | LOW | LOW | HIGH | STOP | FWD |
| LOW | LOW | LOW | LOW | STOP | STOP |

When ‘forward’ command is given to the Arduino through Bluetooth module using the app made, the output 4 and output 5 pins of the Arduino is high this make the input 2 and input 4 of the motor driver active and this makes the dc motor move into forward direction

Similarly, when ‘reverse’ command is given to the Arduino through Bluetooth, the output 3 and output 6 pins of the Arduino is high this make the input 1 and input 3 of the motor driver active and this makes the dc motor move into reverse direction.

When ‘left’ command is given to the Arduino through Bluetooth, the output 4 pin of the Arduino is high this make the input 4 of the motor driver active and this makes the dc motor move into left direction.

When ‘right’ command is given to the Arduino through Bluetooth module using the app made, the output 5 pin of the Arduino is high this make the input 2 of the motor driver active and this makes the dc motor move into right direction.

When ‘stop’ command is given to the Arduino through Bluetooth module using the app made, All the pins of the Arduino are inactive this make all the input pin of the motor driver low and none of the output of the motor driver high, this makes the dc motor to stop.

**Code Implementation in Arduino :**

The code is implemented using android programming language, similar to C/C++. Following code is responsible for the functional output.

*#include <SoftwareSerial.h>*

*SoftwareSerial BT(10, 11); //TX, RX respectively*

*String readvoice;*

*void setup() // One time setup*

*{*

*BT.begin(9600); // Baud rate of Bluetooth module*

*Serial.begin(9600);*

*pinMode(3, OUTPUT);*

*pinMode(4, 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, LOW);*

*digitalWrite (4, HIGH);*

*digitalWrite(5, HIGH);*

*digitalWrite(6,LOW);*

*delay(100);*

*}*

*else if(readvoice == "reverse")*

*{*

*digitalWrite(3, HIGH);*

*digitalWrite(4, LOW);*

*digitalWrite(5, LOW);*

*digitalWrite(6,HIGH);*

*delay(100);*

*}*

*else if (readvoice == "right forward")*

*{*

*digitalWrite (3, LOW);*

*digitalWrite (4,LOW);*

*digitalWrite (5, HIGH);*

*digitalWrite (6,LOW);*

*delay (1500);*

*digitalWrite (3, LOW);*

*digitalWrite (4, LOW);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay(100);*

*digitalWrite(3, LOW); // will make it move forward after turning right*

*digitalWrite (4, HIGH);*

*digitalWrite(5, HIGH);*

*digitalWrite(6,LOW);*

*delay(100);*

*}*

*else if ( readvoice == "left forward")*

*{*

*digitalWrite (3, LOW);*

*digitalWrite (4, HIGH);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay (1500);*

*digitalWrite (3, LOW);*

*digitalWrite (4, LOW);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay(100);*

*digitalWrite(3, LOW); // will make it move forward after turning left*

*digitalWrite (4, HIGH);*

*digitalWrite(5, HIGH);*

*digitalWrite(6,LOW);*

*delay(100);*

*}*

*else if ( readvoice == "left")*

*{*

*digitalWrite (3, LOW);*

*digitalWrite (4, HIGH);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay (1500);*

*digitalWrite (3, LOW);*

*digitalWrite (4, LOW);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay(100);*

*}*

*else if (readvoice == "right")*

*{*

*digitalWrite (3, LOW);*

*digitalWrite (4,LOW);*

*digitalWrite (5, HIGH);*

*digitalWrite (6,LOW);*

*delay (1500);*

*digitalWrite (3, LOW);*

*digitalWrite (4, LOW);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay(100);*

*}*

*else if ( readvoice== "left rotation")*

*{*

*digitalWrite (3, LOW);*

*digitalWrite (4, HIGH);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay (100);*

*}*

*else if ( readvoice == "right rotation")*

*{*

*digitalWrite (3, LOW);*

*digitalWrite (4,LOW);*

*digitalWrite (5, HIGH);*

*digitalWrite (6,LOW);*

*delay (100);*

*}*

*else if (readvoice == "stop")*

*{*

*digitalWrite (3, LOW);*

*digitalWrite (4, LOW);*

*digitalWrite (5, LOW);*

*digitalWrite (6, LOW);*

*delay (25);*

*}*

*readvoice="";}*

*} //Reset the variable*