CONTROL SYSTEMS

ECE2010

PROJECT REPORT

BLUETOOTH CONTROLLED CAR

SUBMITTED FOR THE COURSE: CONTROL SYSTEMS (ECE2010)

BY

Name	Registration Number
Manjari Jain	16BEC0077
Bokka Sampreeti	16BEC0875
Gaduputi Sai Yuktha	17BEC0445

SLOT: A1+TA1

SUBMITTED TO

PROF. BAGUBALI SIR

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
Vellore-632 014, Tamil Nadu, India.
www.vit.ac.in

CERTIFICATE

This is to certify that the project work entitled “Bluetooth controlled Robot Car” that is being submitted by “Manjari Jain, Bokka Sampreeti and Gaduputi Sai Yuktha” for “Control Systems (ECE2010)” is a record of bonafide work done under my supervision. The contents of this Project work have neither been taken from any other source nor have been submitted for any other CAL course.

Place: Vellore
Date:
Signature:

ACKNOWLEDGEMENTS

We are extremely grateful to Prof. Bagubali Sir for guiding us throughout the whole duration of the project, We would like to thank the University for giving us the opportunity to take up this project. We would like to thank fellow friends and classmates who directly or indirectly help us to complete the project successfully.

Place: Vellore
Date:
Signature:

ABSTRACT

Android smartphones are undoubtedly the most popular gadgets these days. Apps on the Internet are designed such that they can exploit inbuilt hardware in these mobile phones, such as Bluetooth and Wi-Fi, to control other devices.

Problem Statement

This project, Bluetooth controlled robot has being designed using the same principle that the robot car can be controlled via an app on mobile. The control commands are sent via Bluetooth.

Features

1. It can be controlled from Android smartphones by touch commands.
2. The speed of the robot can also be controlled.
3. The robot will sense and inform to the phone its distance from the nearest obstacle
4. It will also send information about the direction in which it is moving.
5. The design of the system is kept as simple as possible. Few things like cost-effectiveness and simplicity in design, low-profile structure, etc. have been kept in mind before designing the project.
6. Mobile, robot and Bluetooth are the on-going technologies which can be used for the benefit of mankind.

INTRODUCTION

This project has being designed to use a robot that can be operated using Android mobile phone. The controlling of the Robot is done wirelessly through Android smart phone using the Bluetooth feature present in it. Here in the project the Android smart phone is used as a remote control for operating the Robot.

- **Technological advancement**
  A recent newspaper article read "Robots with WiFi and Bluetooth will now clean Kerala's manholes". The Bluetooth Controlled Robot Car being one of the advanced technologies has being integrated to implement the task to clean manholes**.** The robot, equipped with Wi-Fi, Blue Tooth and control panels has four limbs and a bucket system to scoop out the waste from sewers. The project is supported by KWA, which has joined hands with Kerala start up mission to transform new ideas into practical technologies for addressing issues relating to pipe leakage and sanitation.

- **Detailed literature Survey**
  Robot cars are trending and the future of tomorrow's world. From a toy robot car to the autonomous cars, robots are considered to be used for any application. Considering the Bluetooth controlled car, the application of ultrasonic sensor to detect obstacles is a technological advancement. The robot car can also function using speech recognition.

- **List of intelligence used**
  1. Arduino UNO
     In this system Arduino UNO functions as the microcontroller which contains ATMEGA 328P microcontroller chip. The microcontroller is programmed with the help of the Embedded C programming. Arduino has it own programming burnt in its Read Only Memory (ROM).

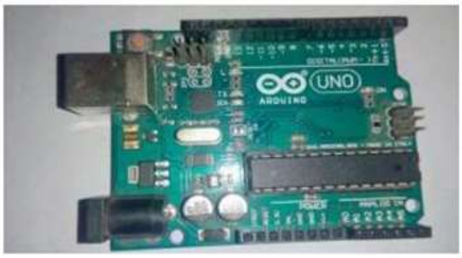


FIGURE 1.Arduino

2. Bluetooth Module (HC-05)
   The Bluetooth module will act as an interface between Smartphone and microcontroller. HC-05 Bluetooth module, can be used as either receiver or transmitter. Generally the transmitter will be smart-phone and receiver will be Bluetooth module.

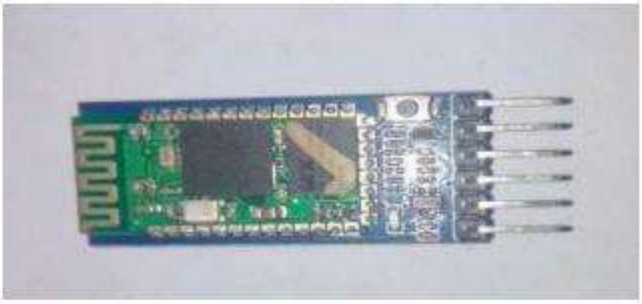


FIGURE 2. Bluetooth Module (HC-05)

3. Smart Phone
   The smart phone is the transmitter of this circuit. It sends the data to microcontroller through Bluetooth module. It also helps to send the instruction of forward, backward, left, right to the microcontroller. Actually, the smart phone is used as a remote of this system.

4. Motor Driver (L293D)
   Motor driver IC is used to control the dc motors. It is also interfaced with the microcontroller and with circuit connections.

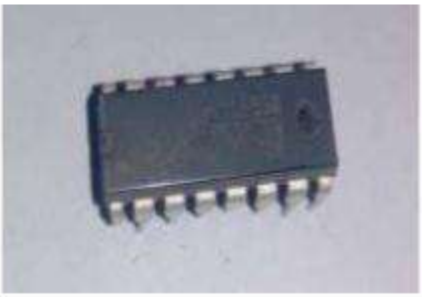


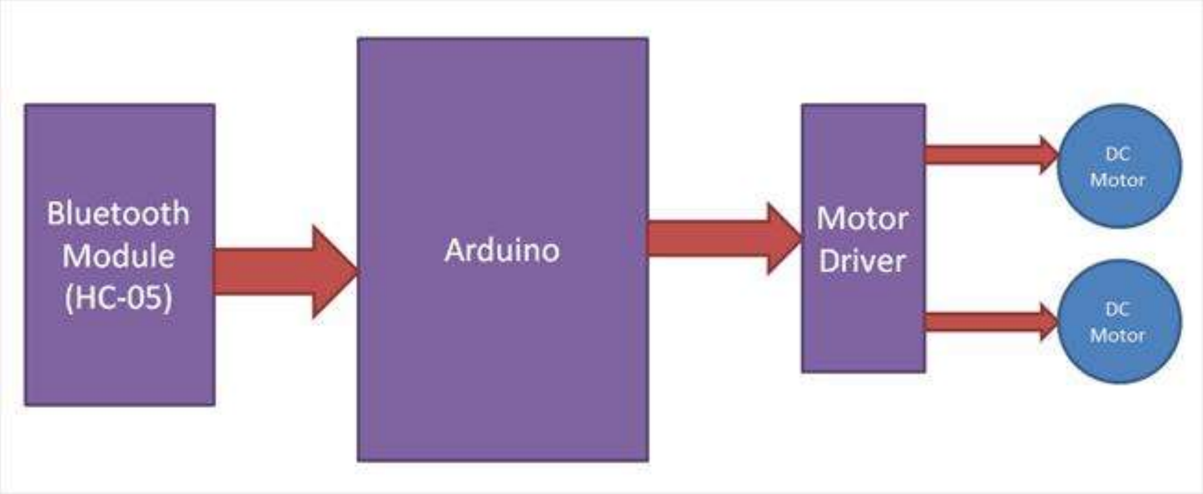
FIGURE 3. Motor Driver (L293D)

5. Battery, 9V
   The battery is connected to the motor driver and also is connected to the Arduino and hence is the power supply to the circuit.

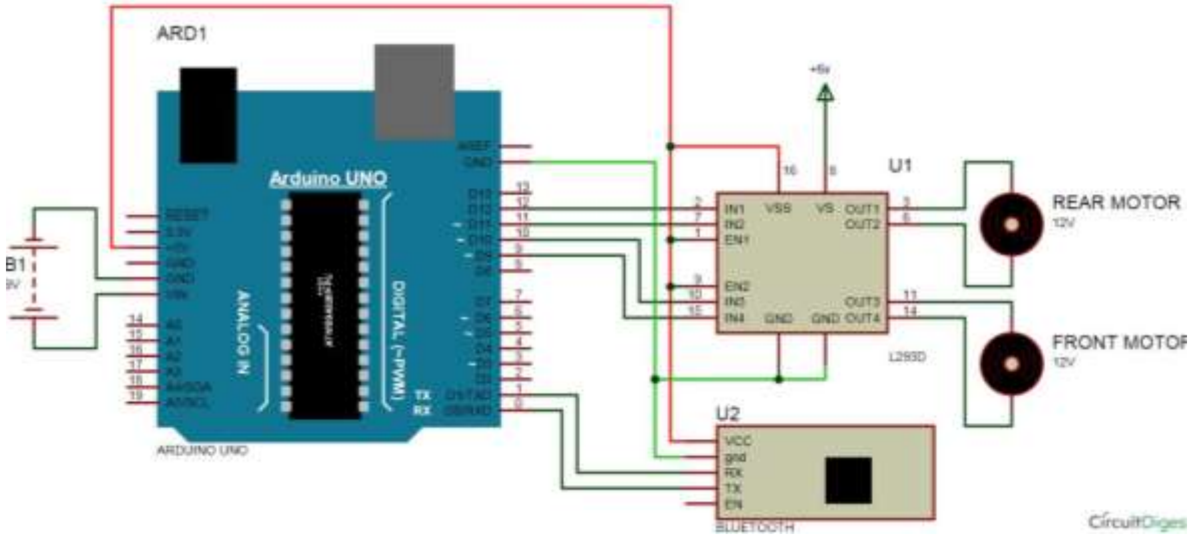
FIGURE 4. 9 V Battery

BLUETOOTH CONTROLLED CAR METHODOLOGY

Block Diagram:



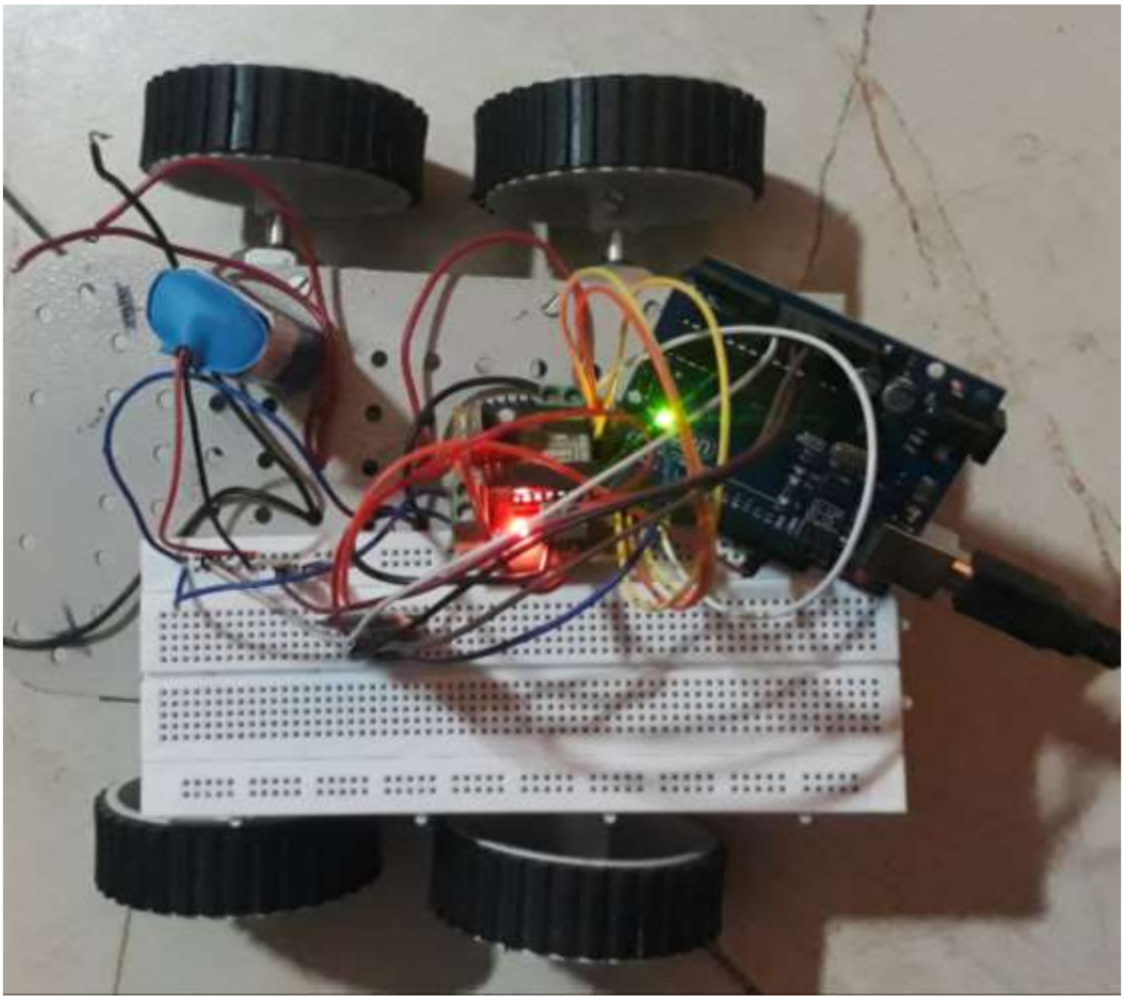
Circuit Diagram:



Working Principle

1. A DC power supply is required to run the system.
2. The DC power supply feeds the Microcontroller and the Bluetooth module. The Bluetooth module receives the signal sent from an android smart-phone, where the application software coded in C language is installed.
3. The microcontroller, thereby, sends instructions, which when executed, helps in functioning of the motor driver.
4. The movement and functioning of the motor can be controlled by using the android based application software.
5. Hardware of this project consists of Arduino UNO, Bluetooth module and a motor driver IC.
6. The project consists of the following three sections:
   a) Input section
   b) Intermediate section
   c) Output section
7. The user (android application) is the input section. This device is connected with the Arduino board (Intermediate section) by the means wirelessly i.e. Bluetooth module. The system can now be connected with the motors (output section) to be controlled via wireless connectivity.

RESULTS AND DISCUSSION



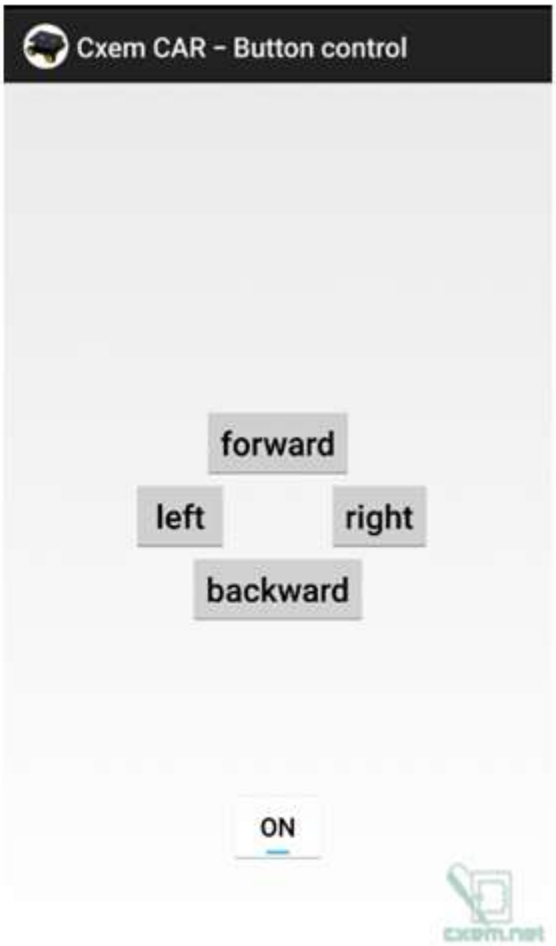


FIGURE. Snapshot of the App designed for Bluetooth Controlled Robot Car

Arduino Code:

```
#include "EEPROM.h"

#define D1 2          // direction of motor rotation 1
#define M1 3          // PWM left motor
#define D2 4          // direction of motor rotation 2
#define M2 5          // PWM right motor
#define HORN 13       // additional channel 1
//#define autoOFF 2500  // milliseconds after which the robot stops when the
connection

#define cmdL 'L'      // UART-command for left motor
#define cmdR 'R'      // UART-command for right motor
#define cmdH 'H'      // UART-command for additional channel (for example Horn)
#define cmdF 'F'      // UART-command for EEPROM operation
#define cmdr 'r'      // UART-command for EEPROM operation (read)
#define cmdw 'w'      // UART-command for EEPROM operation (write)

char incomingByte;    // incoming data

char L_Data[4];       // array data for left motor
byte L_index = 0;     // index of array L
char R_Data[4];       // array data for right motor
byte R_index = 0;     // index of array R
char H_Data[1];       // array data for additional channel
byte H_index = 0;     // index of array H
char F_Data[8];       // array data for  EEPROM
byte F_index = 0;     // index of array F
char command;         // command

unsigned long currentTime, lastTimeCommand, autoOFF;

void setup() {
  Serial.begin(9600);       // initialization UART
  pinMode(HORN, OUTPUT);    // additional channel
  pinMode(D1, OUTPUT);      // output for motor rotation
  pinMode(D2, OUTPUT);      // output for motor rotation
  /*EEPROM.write(0,255);
  EEPROM.write(1,255);
  EEPROM.write(2,255);
  EEPROM.write(3,255);*/
  timer_init();             // initialization software timer
}

void timer_init() {
  uint8_t sw_autoOFF = EEPROM.read(0);   // read EEPROM "is activated or not
stopping the car when losing connection"
  if(sw_autoOFF == '1'){                // if activated
    char var_Data[3];
    var_Data[0] = EEPROM.read(1);
    var_Data[1] = EEPROM.read(2);
    var_Data[2] = EEPROM.read(3);
    autoOFF = atoi(var_Data)*100;       // variable autoOFF ms
  }
  else if(sw_autoOFF == '0'){
    autoOFF = 999999;
  }
  else if(sw_autoOFF == 255){
    autoOFF = 2500;                     // if the EEPROM is blank, dafault value is 2.5 sec
  }
  currentTime = millis();               // read the time elapsed since application start
}

void loop() {
  if (Serial.available() > 0) {         // if received UART data
    incomingByte = Serial.read();       // raed byte
    if(incomingByte == cmdL) {          // if received data for left motor L
      command = cmdL;                   // current command
      memset(L_Data,0,sizeof(L_Data));   // clear array
      L_index = 0;                      // resetting array index
    }
    else if(incomingByte == cmdR) {     // if received data for left motor R
      command = cmdR;
      memset(R_Data,0,sizeof(R_Data));
      R_index = 0;
    }
    else if(incomingByte == cmdH) {     // if received data for additional channel
      command = cmdH;
      memset(H_Data,0,sizeof(H_Data));
      H_index = 0;
    }
    else if(incomingByte == cmdF) {     // if received data for EEPROM op
      command = cmdF;
      memset(F_Data,0,sizeof(F_Data));
      F_index = 0;
    }
    else if(incomingByte == '\r') command = 'e';   // end of line
    else if(incomingByte == '\t') command = 't';   // end of line for EEPROM op

    if(command == cmdL && incomingByte != cmdL){
      L_Data[L_index] = incomingByte;              // store each byte in the array
      L_index++;                          // increment array index
    }
    else if(command == cmdR && incomingByte != cmdR){
      R_Data[R_index] = incomingByte;
      R_index++;
    }
    else if(command == cmdH && incomingByte != cmdH){
      H_Data[H_index] = incomingByte;
      H_index++;
    }
    else if(command == cmdF && incomingByte != cmdF){
      F_Data[F_index] = incomingByte;
      F_index++;
    }
    else if(command == 'e'){                  // if we take the line end
      Control4WD(atoi(L_Data),atoi(R_Data),atoi(H_Data));
      delay(10);
    }
    else if(command == 't'){                  // if we take the EEPROM line end
      Flash_Op(F_Data[0],F_Data[1],F_Data[2],F_Data[3],F_Data[4]);
    }
    lastTimeCommand = millis();               // read the time elapsed since application
start
  }
  if(millis() >= (lastTimeCommand + autoOFF)){     // compare the current timer with
variable lastTimeCommand + autoOFF
    Control4WD(0,0,0);                        // stop the car
  }
}

void Control4WD(int mLeft, int mRight, uint8_t Horn){

  bool directionL, directionR;      // direction of motor rotation L298N
  byte valueL, valueR;              // PWM M1, M2 (0-255)

  if(mLeft > 0){
    valueL = mLeft;
    directionL = 0;
  }
  else if(mLeft < 0){
    valueL = 255 - abs(mLeft);
    directionL = 1;
  }
  else {
    directionL = 0;
    valueL = 0;
  }

  if(mRight > 0){
    valueR = mRight;
    directionR = 0;
  }
  else if(mRight < 0){
    valueR = 255 - abs(mRight);
    directionR = 1;
  }
  else {
    directionR = 0;
    valueR = 0;
  }

  analogWrite(M1, valueL);            // set speed for left motor
  analogWrite(M2, valueR);            // set speed for right motor
  digitalWrite(D1, directionL);       // set direction of left motor rotation
  digitalWrite(D2, directionR);       // set direction of right motor rotation

  digitalWrite(HORN, Horn);           // additional channel
}

void Flash_Op(char FCMD, uint8_t z1, uint8_t z2, uint8_t z3, uint8_t z4){

  if(FCMD == cmdr){           // if EEPROM data read command
    Serial.print("FData:");       // send EEPROM data
    Serial.write(EEPROM.read(0));     // read value from the memory with 0 address and print it to UART
    Serial.write(EEPROM.read(1));
    Serial.write(EEPROM.read(2));
    Serial.write(EEPROM.read(3));
    Serial.print("\r\n");         // mark the end of the transmission of data EEPROM
  }
  else if(FCMD == cmdw){          // if EEPROM data write command
    EEPROM.write(0,z1);               // z1 record to a memory with 0 address
    EEPROM.write(1,z2);
    EEPROM.write(2,z3);
    EEPROM.write(3,z4);
    timer_init();             // reinitialize the timer
    Serial.print("FWOK\r\n");         // send a message that the data is successfully written to EEPROM
  }
}
```

CONCLUSION

The project is indeed a cost-effective and efficient project. The project has a simple and easy to use interface compared to existing ones. Also the Bluetooth RC Controller application is more user friendly. The robot is small in size so it can be used in spying purpose.

With the help of this project, parking becomes easier as this car helps in automatic control and hence reduces human efforts and involvement.

FUTURE WORK

- With few additions and modifications, this robot can be used in army for detecting and disposing hidden land mines.
- The robot can be used for surveillance. In future we can interface sensors to this robot so that it can monitor some parameters and we can improve the efficiency using Internet of Things (IoT) technology.
- We can also add wireless camera, in order to incorporate other security features.

REFERENCES

1. https://electronicsforu.com/electronics-projects/hardware-diy/arduino-android-phone-controlled-robot
2. http://indianexpress.com/article/india/kerala-to-use-robots-to-clean-sewers-end-manual-scavenging-5020954/
3. https://circuitdigest.com/microcontroller-projects/bluetooth-controlled-robot-car-using-arduino
4. https://www.pantechsolutions.net/robotics-projects/android-controlled-arduino-robot-car
5. https://www.youtube.com/watch?v=PNm-9I9KAR8