International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 11 | Nov 2020 www.irjet.net p-ISSN: 2395-0072

ARDUINO BLUETOOTH RC MOTOR CAR

Dr. K M Purushothama[1], Prof. M. Rathika[2], Prof. Srinuvasu. N[3]

[1, 2,3]*Professor, Department of Mechanical Engineering, Dr. Ambedkar Institute of Technology -Bangalore-056*

Abstract - *Radio controlled (or R/C) cars are battery/gas-powered model cars or trucks that can be controlled from a distance using a specialized transmitter or remote. It is through efficient electronic programming that a computer can control a car, hence a car can be thought of as an electromechanical machine. Some of the essential characteristics that a car must have are - sensing, movement, energy, intelligence. It performs a task using control systems, various power supplies and software all working together. We developed an Android application which uses remote buttons to guide an RC car's motion. Hence, the mobile device harboring the Android application acts as the car's remote control. Bluetooth is the basis of communication between the controller and Android.*

Key Words*: *Bluetooth module, Ultrasonic Sensor, Arduino Mega 2560, DC Motor, L298n motor controller, Android Application.

1. INTRODUCTION

The term "R/C" has been used to mean both "remote controlled" and "radio controlled", where "remote controlled" includes vehicles that are connected to their controller by a wire, but common use of "R/C" today usually refers to vehicles controlled by a radiofrequency link.

As Bluetooth is used for Data Exchange, it is through this data exchange capability that through Bluetooth, Devices are now being controlled and monitored. Bluetooth technology was created in 1994 by the telecom dealer "Ericsson" for integrating with Smart phones. But through the years, with dramatic increase in smart phone Users, Bluetooth has turned them into all-purpose portable devices by redefining the world of data exchange and transferring wired devices into wireless devices; capable of efficient communication and the fact that host Bluetooth device is capable of communicating with as many as seven Bluetooth modules simultaneously through one link is proof enough. In this paper, we present a review of current robots, which are controlled by mobile phones and tablets. Specifically speaking, robots which on receiving the commands can perform simple actions like moving in all four directions, by an Android application.

1.1 Bluetooth module: The HC-05 Bluetooth sensor is used to send signals from the android app to the arduino board. The BT sensor has 4 mains pins which are being used:

a. Vcc pin for supply voltage form 3.3v to 5v
b. GND or ground is connected to GND on the arduino board
c. RX and TX pins are used as UART interface for communication and is connected to TX and RX pins on arduino board respectively.

1.2 Ultrasonic Sensor: The ultrasonic distance sensor HC-05 has 4 pins which are connected to the arduino mega 2560 board using jumper wires. The sensor has 4 pins:

a. Vcc or 5v pin connected to 5v pin on arduino board.
b. Trig pin connected to terminal 7 on arduino board.
c. Echo pin connected to terminal 6 on arduino board.
d. GND or ground connected to any of the ground on arduino board.

1.3 Arduino Mega 2560: The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 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 an AC-to-DC adapter or battery to get started.

1.4 DC Motor: A Dc Motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

1.5 L298n motor controller: The out 1 through 4 of the L298n microcontroller is connected to 4 terminals of the 2 DC motors. The DC enable jumper 1 and 2 is connected to pins 2 and 3 of the Arduino board. The IN1 IN2 IN3 IN4 pins are connected to terminals 10, 11, 12, 13 of the Arduino board respectively. The 12v and the GND pin of the microcontroller is connected to the +ve and –ve terminals of the battery. The same ground pin is connected to any GND pin on the board.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 11 | Nov 2020 www.irjet.net p-ISSN: 2395-0072

2. PROGRAMMING:

```
#define trigPin 7
#define echoPin 6

#define motorAen  2
#define motorBen  3
#define motorAin1 10
#define motorAin2 11
#define motorBin1 12
#define motorBin2 13

#define forwardon 49    // ascii 1
#define forwardoff  50  // ascii 2
#define backwardon 55   // ascii 7
#define backwardoff  56 // ascii 8
#define righton 51      // ascii 3
#define rightoff  52    // ascii 4
#define lefton 53       // ascii 5
#define leftoff  54     // ascii 6


int readBTinput() {

  int incomingByte = 0;   // for incoming serial data
  int incomingtemp=0;
      if (Serial1.available() > 0) {
              // read the incoming byte:
        incomingtemp = Serial1.read();

        if (incomingtemp != 10) {  //check if it is ^J
          incomingByte = incomingtemp;
        }
      }
      return(incomingByte);
}

long   moveforward(int motor_en, int motorin1, int motorin2, int mspeed)  {
   digitalWrite(motorin1, HIGH);
   digitalWrite(motorin2, LOW);
   analogWrite(motor_en, mspeed);
}


long   stopmotor(int motorin1, int motorin2){
   digitalWrite(motorin1, LOW);
   digitalWrite(motorin2, LOW);
}


long   movebackword(int motor_en, int motorin1, int motorin2, int mspeed)  {
   digitalWrite(motorin1, LOW);
   digitalWrite(motorin2, HIGH);
   analogWrite(motor_en, mspeed);
}
```

```
long ultrasonicdistance() {
  long duration;
  digitalWrite(trigPin , LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin , HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin , LOW);

  duration = pulseIn(echoPin , HIGH);
  return ((duration/2) / 29.1);


}


void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  pinMode(trigPin , OUTPUT);
  pinMode(echoPin , INPUT);

  pinMode(motorAen, OUTPUT);
  pinMode(motorBen, OUTPUT);
  pinMode(motorAin1, OUTPUT);
  pinMode(motorAin2, OUTPUT);
  pinMode(motorBin1, OUTPUT);
  pinMode(motorBin2, OUTPUT);

void loop() {
  long distance;
  int incommand=0;

  incommand=0;
  // first, make sure we dont have an obstacle
  distance = ultrasonicdistance();
  Serial.print(distance);
  Serial.println(" cm");
  if(distance <= 25 || distance <= 0)  {
     Serial.print("In range : ");
     Serial.println(distance);
     stopmotor(motorAin1,motorAin2);
     stopmotor(motorBin1,motorBin2);
  } else {

    Serial.print("Good to go : ");
    Serial.println(distance);
//      delay(500);
// check if we have an input command
// and do stuff
    incommand=readBTinput();
    Serial.print("Got input: ");
    Serial.println(incommand);
```

```
if (incommand >0) {
  Serial.println("We have to act");
// do the stuff for moving the vehicle
  switch (incommand) {
    case forwardon:
      Serial.println("Forward move on");
      moveforward(motorAen, motorAin1, motorAin2, 200);
      moveforward(motorBen, motorBin1, motorBin2, 200);
      break;

      case forwardoff:
        Serial.println("Forward move off");
        stopmotor(motorAin1,motorAin2);
        stopmotor(motorBin1,motorBin2);
        break;
      case backwardon:
        Serial.println("Backward move on");
       movebackword(motorAen, motorAin1, motorAin2, 200);
        movebackword(motorBen, motorBin1, motorBin2, 200);
        break;
      case backwardoff:
        Serial.println("Backward move off");
        stopmotor(motorAin1,motorAin2);
        stopmotor(motorBin1,motorBin2);
        break;
       case righton:
        Serial.println("right move on");
        stopmotor(motorAin1,motorAin2);
        moveforward(motorBen, motorBin1, motorBin2, 200);
        break;
       case rightoff:
        Serial.println("right move off");
        stopmotor(motorAin1,motorAin2);
        stopmotor(motorBin1,motorBin2);
        break;
       case lefton:
        Serial.println("left move on");
        stopmotor(motorBin1,motorBin2);
        moveforward(motorAen, motorAin1, motorAin2, 200);
        break;
       case leftoff:
        Serial.println("left move off");
        stopmotor(motorAin1,motorAin2);
        stopmotor(motorBin1,motorBin2);
        break;
      default:
        Serial.println("Default");
        break;
     }
   }
 }
```

3. RESULT AND DISCUSSION

3.1 Obstacle Detection Calibration: The calibration curve for the center infrared ranger of the obstacle detection array is shown below in. This calibration curve is used to determine the upper limit on the distance when microcontroller running the obstacle detection program signals the second microcontroller to override the PID algorithm and set the Smart Car in neutral.

3.2 Android Application: We make use of "Robo Remo Free Arduino Bluetooth" app from the Google play store. It's an open source app, we used it to create necessary control buttons for control of directions i.e. forward, backward, right, left by setting the press action and release action as 1,2,3,4,5,6,7,8 respectively for various directions. The Android app send ASCII values via Bluetooth and this is received by HC05 Bluetooth module, as a result of which the Arduino Mega 2560 executes the set of instructions in the program as mentioned earlier.

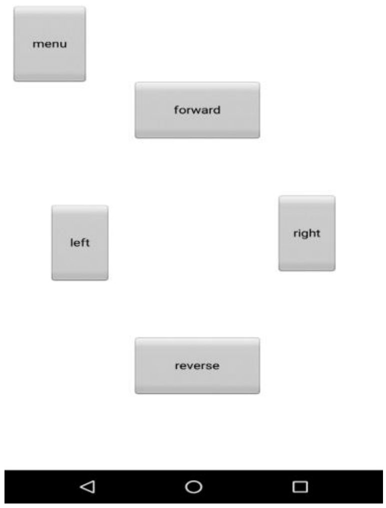


Figure3.1: Layout of Robo Remo Free

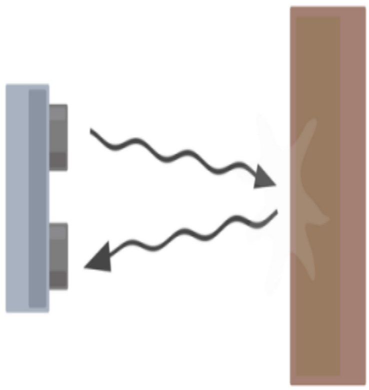
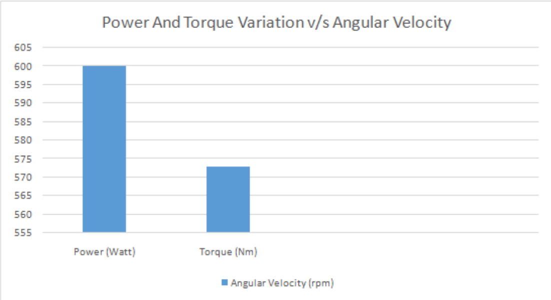


Figure 3.2: ultrasonic sensor operation

Arduino Bluetooth App



Graph 3.1 Power and torque variations V/S Angular Velocity

3.3Necessary Calculations:

I_{max} = P_{max} = (12 Volt)*(I_{max}) = 600 Watt

Showing torque as T, and speed as w (in rpm), and assuming no loss:

T * ω * (2*π/60) = P_{max}

For the chosen typical value:

T * ω = 5730

Now you can have T = 5.73 N.m for ω = 1000 rpm, Or T = 573 N.m with ω= 10 rpm, or any acceptable solution that is safe for the mechanical system.

4. CONCLUSION

The focus of the pre-programmed RC car is to develop a system that can realize a user input in the form of motion. In order to accomplish this, the project will be broken into several components. First, input will be gathered and processed as a user inputs the command. Once the command has been processed, the information will be sent across a wireless connection to the radio-controlled vehicle. The vehicle will then execute the desired function. If an obstacle is detected, an interrupt will be sent through the vehicle's Arduino board to the stop the vehicle for a new route to be determined.

5. REFERENCES

[1]. Potts, J. and Sukittanon, S. (2012) "Exploiting Bluetooth On android mobile devices for home security application", Proceedings of southeast can, 15-18 March 2012, Orlando, Florida, USA.
[2]. Benjamin C. Kuo, Digital Control Systems, (2nd ed.), Oxford University Press, Inc, New York, 1992. J. Lowen Shearer, Bohdan T. Kulakowski, John F. Gardner, Dynamic Modeling and Control of Engineering Systems, (2nd ed.), Prentice Hall, Upper Saddle River, New Jersey, 1997, pp. 290 – 294.
[3]. Willie D. Jones, "Building Safer Cars", IEEE Spectrum, January 2002, pp. 82 – 85.
[4]. Richard Bishop, "A Survey of Intelligent Vehicle Applications Worldwide", Proceedings of the IEEE Intelligent Vehicles Symposium 2000, October 3 – 5, 2000, pp. 25 – 30.
[5]. 6.HarshavardhanB.Patil, Prof.V.M.Umale Arduino Based Wireless Biomedical Parameter Monitoring System Using Zigbee.|| IJETT, Volume 28 Number 7 - October 2015.
[6]. Michael B. Histand, David G. Alciatore, Introduction to Mechatronics and Measurement Systems, McGraw-Hill Companies, 1999.