**Index**

|  |  |  |
| --- | --- | --- |
| **Si no.** | **Topic** | **Page No.** |
| 1. | Abstract | 2 |
| 2. | Bluetooth technology | 3 |
| 3. | Arduino UNO | 4 |
| 4. | Design | 7 |
| 5. | Analysis | 10 |
| 6. | Source Code | 11 |
| 7. | Expenditure | 15 |
| 8. | Conclusion | 16 |
| 9. | References | 16 |

**Abstract**

A remote control vehicle is defined as any mobile device that can be controlled by a means that does not restrict its motion with an origin external to the device. It is often a radio controlled device, cable between control and vehicle or an infrared or a Bluetooth controller.

A remote control vehicle or RCV is always controlled by a human and takes no positive action autonomously. It is vital that the vehicle should be capable of proceeding precisely to a target area, manoeuvring within that area to fulfil its mission and returning equally and accurately back to base.

In this project, we have used **Bluetooth technology** in a remote controlled vehicle. This a very basic model which has a simple communication system and can be easily implemented. The vehicle movements can be controlled using an android based application that has a built- in Bluetooth feature. The user has to install the application in order to control the car. Operations like moving FORWARD, BACKWARD, LEFT AND RIGHT can be carried out using the commands sent from the application.

The task of controlling movements of the car is done by the **Arduino Uno** having micro-controller ATMEGA32. The main advantage of the Arduino technology is, you can directly load the programs into the device without the need of a hardware programmer to burn the program. Arduino has hence made it easier to convert digital and analog signals to physical movements of the RCV.

A Bluetooth module has also been used in the remote car which gets connected to the phone‘s Bluetooth, that allows us to communicate with it. Being Bluetooth based, it gives us the advantage of changing the remote anytime and to any device including phones, tablets and computers. Physical barriers like walls, doors, do not affect the controlling of the device provided the car and the remote remain in the Bluetooth range.

**Bluetooth Technology**

Bluetooth wireless technology is a short-range radio technology, which is developed for Personal Area Network (PAN). Bluetooth is a standard developed by a group of electronics manufacturers, that allows any sort of electronic equipment, from computers and cell phones to keyboards and headphones, to make its own connections, without wires, cables or any direct action from a user. It is an ad-hoc type network operable over a small area such as a room. Bluetooth wireless technology makes it possible to transmit signals over short distances between telephones, computers and other devices and thereby simplify communication and synchronization between devices. It is a global standard that eliminates wires and cables between both stationary and mobile devices and facilitates both data and voice communication. Bluetooth offers the possibility of ad hoc networks and delivers the ultimate synchronicity between all your personal devices. Bluetooth is a dynamic standard where devices can automatically find each other, establish connections, and discover what they can do for each other on an ad hoc basis.

Bluetooth is intended to be a standard that works at two levels:

1. It provides agreement at the physical level - Bluetooth is a radio-frequency standard.
2. It also provides agreement at the next level up, where products have to agree on when bits are sent, how many will be sent at a time and how the parties in a conversation can be sure that the message received is the same as the message sent.

Bluetooth is a standard for a small, cheap radio chip to be plugged into computers, printers, mobile phones, etc. A Bluetooth chip is designed to replace cables by taking the information normally carried by the cable, and transmitting it at a special frequency to a receiver Bluetooth chip, which will then give the information received to the computer, phone, etc.

**Arduino UNO**

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino consists of both a physical programmable circuit board and a piece of software, or IDE (Integrated Development Environment) that runs on user's computer, used to write and upload computer code to the physical board.

The Arduino Uno microcontroller board (Fig.) is based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. It can be powered by simply connecting to a computer with a USB cable or with a AC-to-DC adapter or battery to get started.

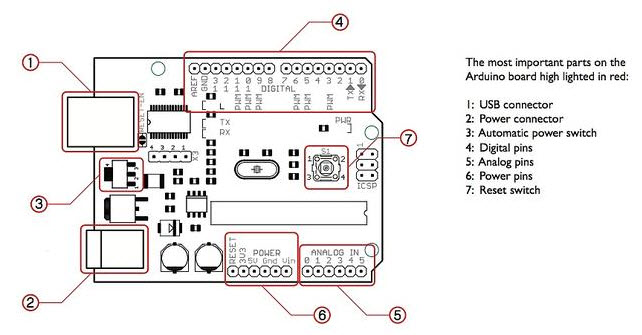


Figure 1: Arduino UNO pin diagram

**Arduino Architecture:**

The processor of the Arduino board uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories such as program memory and data memory. Wherein the data is stored in data memory and the code is stored in the flash program memory. The Atmega328 microcontroller has 32kb of flash memory, 2kb of SRAM  1kb of EEPROM and operates with a 16MHz clock speed.

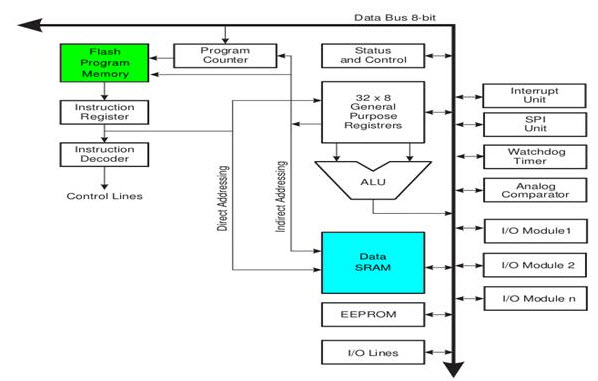


Figure 2: Arduino Architecture

The main advantage of the Arduino technology is, you can directly load the programs into the device without the need of a hardware programmer to burn the program. This is done because of the presence of the 0.5KB of boot loader, that allows the program to be dumped into the circuit. The Arduino tool window contains a toolbar with a various buttons like new, open, verify, upload and serial monitor. And additionally it comprises of a text editor (employed to write the code), a message space (displays the feedback) like showing the errors, the text console, that displays the o/p & a series of menus just like the file, tool menu & edit.

* [Programming into the](http://www.edgefxkits.com/blog/labview-programming-language-uses/) Arduino board is called as sketches. Each sketch contains of three parts such as Variables Declaration, Initialization and Control code. Where, Initialization is written in the setup function and Control code is written in the loop function.
* The sketch is saved with .ino and any operation like opening a sketch, verifying and saving can be done using the tool menu.
* The sketch must be stored in the sketchbook directory.
* Select the suitable board from the serial port numbers and tools menu.
* Select the tools menu and click on the upload button, then the boot loader [uploads the code on the microcontroller](http://www.edgefxkits.com/blog/free-ebook-to-learn-and-design-your-own-microcontroller-projects/).

Basic Functions of Arduino Technology:

* Digital read pin reads the digital value of the given pin.
* Digital write pin is used to write the digital value of the given pin.
* Pin mode pin is used to set the pin to I/O mode.
* Analog read pin reads and returns the value.
* Analog write pin writes the value of the pin.
* Serial. Begins pin sets the beginning of serial communication by setting the rate of bit.

Advantages of Arduino Technology

* It is cheap
* It comes with an open supply hardware feature that permits users to develop their own kit
* The software of the Arduino is well-suited with all kinds of in operation systems like Linux, Windows, and Macintosh, etc.
* It also comes with open supply software system feature that permits tough software system developers to use the Arduino code to merge with the prevailing programming language libraries and may be extended and changed.
* For beginners, it is very simple to use.

**Design**

Hardware:

Hardware of this project consists of Arduino UNO, Bluetooth module and a motor driver IC. The Bluetooth module is connected with the Arduino UNO board for the connection with the user. Through the Bluetooth module for monitoring and controlling the particular motor reaches the board and process accordingly and the output of the Arduino goes to the motor driver IC and it controls the particular motor

1. **Arduino UNO** – As mentioned earlier, Arduino UNO houses microcontroller ATMEGA328P which is responsible for the working of the car.

We have used Arduino because it is an open source device which can be programmed through any operating system like Windows, Mac, Linux, etc. The language used is understandable and easy. Changing of program is also very easy. Various shields and modules that are easily connected to Arduino are available for various purpose like, if we want to connect the Arduino to a network then a Wi-Fi shield is available. For controlling the motor a motor shield is available, and for this project a Bluetooth module is used.



BFigure 3: Arduino UNO

1. **Bluetooth Module HC05**- HC‐05 module(Fig.4) is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04‐External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).



Figure 4: Bluetooth Module HC05

1. **L293D motor driver:** L293D is a typical Motor driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motors with a single L293D IC.

In a single L293D chip there are two H-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors. Given below is the pin diagram of a L293D motor controller.

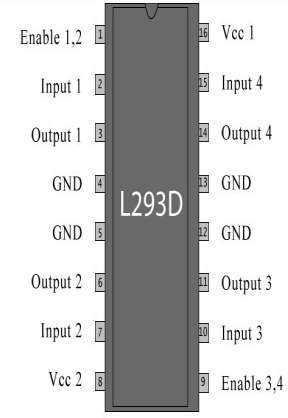


Figure 5: Motor driver L293D

Figure 6: Motor

## Motors:

## The motors used are geared L-shaped battery operated dc motors. Gears reduce the speed of the vehicle but increase its torque. This is known as gear reduction. Gear Ratio of motors used is 150:1 and run at 60RPM. The setup assembly helps in increasing the torque and reducing the motor speed.

## Software:

1. **IDE**: The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

1. **Mobile App:** Here, we have used ‘Arduino BlueControl’ Application, available on the Play Store, that allows users to directly control their ARDUINO based device without having built a specific application for the same. This application can be downloaded on any device to control the car.

**Analysis of the project**

The working model of our remote controlled car can be understood by the simple block diagram given below:

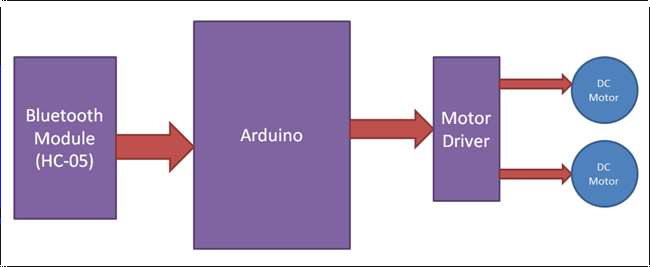


Figure 7: Block Diagram

A DC power supply is required to run the system. The DC power supple 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. The microcontroller, thereby, sends instructions, which when executed, helps in functioning of the motor driver. The movement and functioning of the motor can be controlled by using the android based application software.

Implementation:

The project is implemented on top of the chassis. The Bluetooth module HC05 is connected to the Arduino UNO- GND pins to each other and Vcc pin to 5V power from the Arduino to the Bluetooth module. Transmitter pin Txd of Bluetooth module is connected to Receiver pin Rx (0) of Arduino and vice versa. Motor driver module L293D input pins’ 14, 13, 12 and 11 are connected to Arduino digit output pins 9, 8, 10 and 11 respectively. The output pins from the motor driver module (M1, M2) connected to the dc motor 1 (right) terminals and M3, M4 connected to the terminals of dc motor 2 (left).

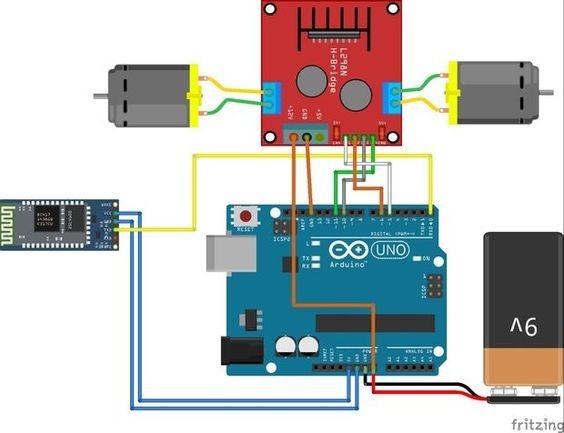


Figure 8: Circuit Diagram

**Source Code**

/\* Source code for Bluetooth controlled car \*/

int lf=8;

int lb=9;

int rf=10;

int rb=11;

int stopped=1;

char input;

void setup()

{

pinMode(lf,OUTPUT);

pinMode(rf,OUTPUT);

pinMode(lb,OUTPUT);

pinMode(rb,OUTPUT);

Serial.begin(9600);

}

void stop()

{

digitalWrite(lb,LOW);

digitalWrite(lf,LOW);

digitalWrite(rb,LOW);

digitalWrite(rf,LOW);

}

void front()

{

digitalWrite(lf,HIGH);

digitalWrite(lb,LOW);

digitalWrite(rb,LOW);

digitalWrite(rf,HIGH);

}

void back()

{

digitalWrite(lf,LOW);

digitalWrite(rf,LOW);

digitalWrite(lb,HIGH);

digitalWrite(rb,HIGH);

}

void right()

{

digitalWrite(rb,LOW);

digitalWrite(lf,LOW);

digitalWrite(rf,HIGH);

digitalWrite(lb,HIGH);

}

void left()

{

digitalWrite(lb,LOW);

digitalWrite(lf,HIGH);

digitalWrite(rb,HIGH);

digitalWrite(rf,LOW);

}

void loop()

{

if(Serial.available())

{

input='z';

input=Serial.read();

Serial.print(input);

Serial.print("\t");

if(input=='w')

{

if (stopped ==1)

{

front(); //input=Serial.read();

stopped=0;

}

else

{

stopped=1;

stop();

}

}

if(input=='a')

{

if(stopped==1)

{

left();

stopped=0;// input=Serial.read();

}

else

{

stopped=1;

stop();

}

}

if(input=='s')

{

if(stopped==1)

{

back();

stopped=0;// input=Serial.read();

}

else

{

stopped=1;

stop();

}

}

if(input=='d')

{

if(stopped==1)

{ right();// input=Serial.read();

stopped=0;

}

else

{

stopped=1;

stop();

}

} } }

**Expenditure**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **No. of components** | **Price of each** | **Total cost** |
| Arduino UNO | 1 | 405 | 405 |
| Bluetooth Module HC05 | 1 | 240 | 240 |
| Motor Driver L2930 | 1 | 120 | 120 |
| Dc Motor | 2 | 30 | 60 |
| Chassis and Wheels | 1 | 130 | 130 |
| Connecting wires | 6 |  | 20 |
| Jumper wires | 8 |  | 40 |
| Batteries | 2 | 20 | 40 |

**Conclusion**

We can therefore implement a Bluetooth controlled car using this simple and cost efficient method. With few additions and modifications, this car can be used for various other applications such as, in army for detecting and disposing hidden land mines. The robot can be used for surveillance. In future we can interface sensors such as Ultrasound sensor (obstacle detection) 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.

Arduino Uno has enormous utilities especially while implementing lower scale projects as the code isn’t embedded onto the rom but can be changed as and when required. Arduino provides a cheap and easy way for beginners to get a gist of developing such projects.

**References**

<https://www.edgefxkits.com/blog/arduino-technology-architecture-and-applications/>

<https://howtomechatronics.com/tutorials/arduino/how-to-configure-pair-two-hc-05-bluetooth-module-master-slave-commands/>

<http://www.ijettjournal.org/2016/volume-33/number-8/IJETT-V33P274.pdf>

<https://www.youtube.com/watch?v=zhWGeJnIfeY>