### MINOR PROJECT REPORT ON

PHONE CONTROLLED R.C. CAR

### Submitted in partial fulfillment of requirements for the award of the

Degree of

**Bachelor of Technology In**

**Computer Science and Engineering**

**Submitted By:**

Prabhav (00655202718)

Utkarsh Tiwari (40355202718)

Tarana Singhal (53155202718)



**BHAGWAN MAHAVEER COLLEGE OF ENGINEERING AND MANAGEMENT**

**Affiliated to**

**Guru Gobind Singh Indraprastha University**

**Dwarka, Sector 16, New Delhi**

**2018-2022**

*ACKNOWLEDGEMENT*

We express our sincere gratitude to Mrs. Gurpreet Kaur for his valuable guidance and timely suggestions during the entire duration of my dissertation work, without which this work would not have been possible. we would also like to convey my deep regards to all other faculty members, who have bestowed their great effort and guidance at appropriate times without which it would have been very difficult on our part to finish this work. Finally, we would also like to thank our friends for their advice and pointing out our mistakes.

#### *CANDIDATE’S DECLARATION*

We hereby declare that the work presented in this report entitled “PHONE CONTROLLED R.C. CAR”, in fulfillment of the requirement for the award of the degree Bachelor of Technology in Information Technology, submitted in CSE Department, BMCEM affiliated to Guru Gobind Singh Indraprastha University, New Delhi, is an authentic record of our work carried out during our degree under the guidance of Mrs. Gurpreet Kaur.

The work reported in this has not been submitted by us for award of any other degree or diploma.

Date: Utkarsh Tiwari (40355202718)

Place: Prabhav (00655202718)

Tarana Singhal(53155202718)

#### *ABSTRACT*

A remote controlled vehicle is any mobile machine controlled by means that is physically not connected with origin external to the machine. There are many types in it, based on the controls – radio control device, Wi-Fi controlled and even Bluetooth controlled. These devices are always controlled by humans and take no action autonomously. The main target in such vehicles would be to safely reach a designated point, maneuver the area and reach back to the point of origin. In this project we make use of the Bluetooth technology to control our machine car. We don’t call this as a robot as this device doesn’t have any sensors. Thereby, sensor less robots are machines. This machine can be controlled by any human using his android mobile phone, by downloading an app and connecting it with the Bluetooth module present inside our car. User can perform actions like moving forward, backward, moving left and right by the means of command using his-her mobile phone app. The task of controlling our car is taken car by the Arduino UNO with micro. Arduino play a major role in the control section and had made it easier to convert digital signals and analogue signals into physical movements. The major reason for using a Bluetooth based tech is that we can change the remote anytime – mobiles phones, tablets and laptops and physical barriers like wall or doors do not affect the car controls

*TABLE OF CONTENTS*

### 

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **CONTENT** | **SIGN** |
| 1 | Material Used |  |
| 2 | Theory |  |
| 3 | Circuit Diagram |  |
| 4 | Working |  |
| 5 | Arduino source code |  |
| 6 | Mobile Application |  |
| 7 | Mobile app source code |  |
| 8 | Result |  |
| 9 | Conclusion |  |
| 10 | References |  |

***Materials Used***

|  |  |  |
| --- | --- | --- |
| Materials | Model or Specifications (if any) | Model Picture |
| Arduino Board | UNO with ATMEGA32  micro controller |  |
| Bluetooth Modules | HC-05 | C:\Users\K S Kishore Kumaar.LAPTOP-T5PUTT6M\Desktop\hc-05-bluetooth-module-prayogindia.jpg |
| Motor Drive Shield | L293D Motor Driver | C:\Users\K S Kishore Kumaar.LAPTOP-T5PUTT6M\AppData\Local\Microsoft\Windows\INetCache\Content.Word\L293D-Motor-Driver-Shield-for-Arduino-Uno-1.png |
| DC Gear Motor | 12V, 200rpm | C:\Users\K S Kishore Kumaar.LAPTOP-T5PUTT6M\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.jpg |
| Stainless Steel Nails and Nylon Jumpers |  | |
| Connecting Wires and Jumper Cables | C:\Users\K S Kishore Kumaar.LAPTOP-T5PUTT6M\Desktop\A_few_Jumper_Wires.jpg C:\Users\K S Kishore Kumaar.LAPTOP-T5PUTT6M\Desktop\download (2).jpg | |
| Castor Wheel  And  Normal Wheels | Buy Castor Wheel : ElementzOnline INDIA65mm Rubber Tyre/Wheel, Bo Wheel, Smart Car Wheels, RC Car Wheels, Wheels  for bo Motor -YELLOW (Pack of 2) : Amazon.in: Toys &amp; Games | |
| Chassis |  | |
| Bread Board |  | |

*Theory*

Bluetooth-

Bluetooth wireless technology is a short range communications technology intended to replace the cables connecting portable unit and maintaining high levels of security. Bluetooth technology is based on Ad-hoc technology also known as Ad-hoc Pico nets, which is a local area network with a very limited coverage.

WLAN technology enables device connectivity to infrastructure based services through a wireless carrier provider. The need for personal devices to communicate wirelessly with one another without an established infrastructure has led to the emergence of Personal Area Networks (PANs). Bluetooth specification details the entire protocol stack. Bluetooth employs Radio Frequency (RF) for communication. It makes use of frequency modulation to generate radio waves in the ISM band.

The usage of Bluetooth has widely increased for its special features. Bluetooth offers a uniform structure for a wide range of devices to connect and communicate with each other. Bluetooth technology has achieved global acceptance such that any Bluetooth enabled device, almost everywhere in the world, can be connected with Bluetooth enabled devices. Low power consumption of Bluetooth technology and an offered range of up to ten meters has paved the way for several usage models. Bluetooth offers interactive conference by establishing an adhoc network of laptops. Bluetooth usage model includes cordless computer, intercom, cordless phone and mobile phones

Arduino UNO Board-

The Arduino Uno is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits. The board has 14 digital I/O pins (six capable of [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) output), 6 analog I/O pins, and is programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment), via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable). It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts.

It is similar to the [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) and Leonardo. The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available. The word "[uno](https://en.wiktionary.org/wiki/uno)" means "one" in [Italian](https://en.wikipedia.org/wiki/Italian_language) and was chosen to mark the initial release of [Arduino Software](https://en.wikipedia.org/wiki/Arduino_Software).

The Uno board is the first in a series of USB-based Arduino boards; it and version of the Arduino [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment) were the reference versions of Arduino, which have now evolved to newer releases.

The ATmega328 on the board comes preprogrammed with a [bootloader](https://en.wikipedia.org/wiki/Bootloader) that allows uploading new code to it without the use of an external hardware programmer.

While the UNO communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a [USB-to-serial converter](https://en.wikipedia.org/wiki/USB-to-serial_converter).

# Motor Driver-

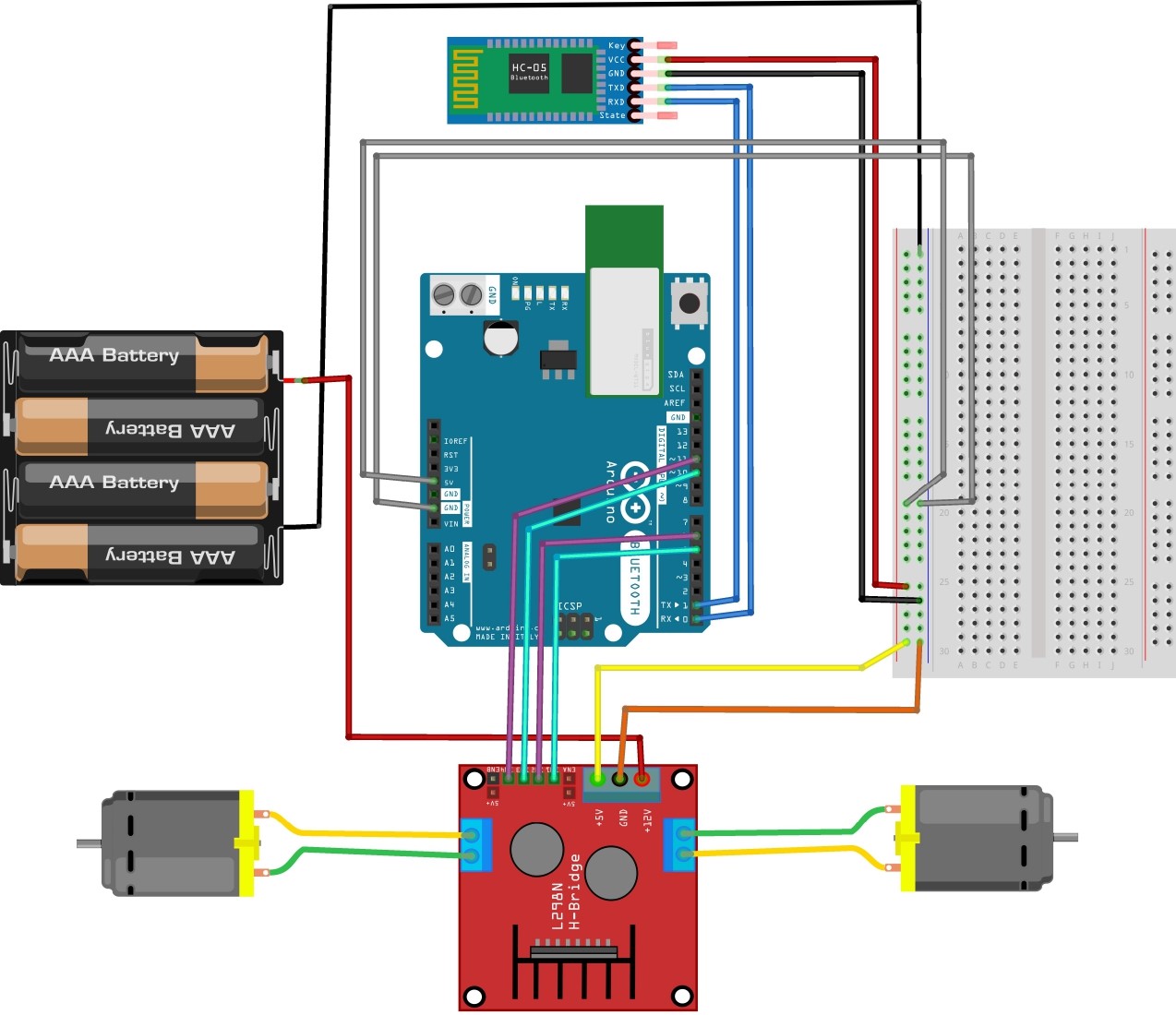
The Motor Driver is a module for motors that allows you to control the working speed and direction of two motors simultaneously. This Motor Driver is designed and developed based on L293D IC.

L293D is a 16 Pin Motor Driver IC. This is designed to provide bidirectional drive currents at voltages from 5 V to 36 V.

Rotation of motor depends on Enable Pins. When Enable 1/2 is HIGH, motor connected to left part of IC will rotate according to following manner:

|  |  |  |
| --- | --- | --- |
| Input 1 | Input 2 | Result |
| 0 | 0 | Stop |
| 0 | 1 | Anti-Clockwise |
| 1 | 0 | Clockwise |

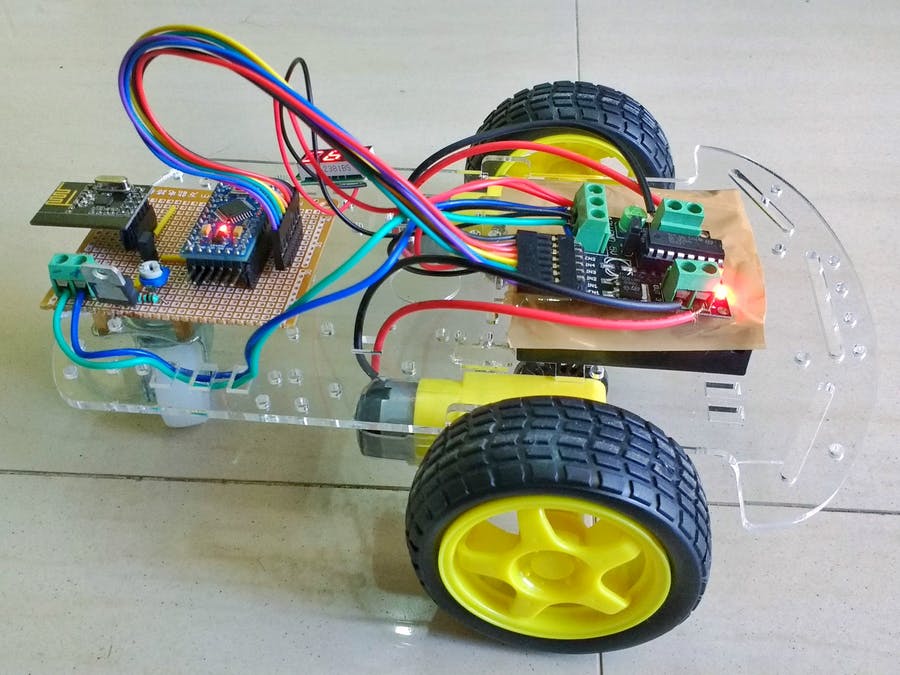
*Circuit Diagram*

**

*Working*

Take a closer look on the Wiring Diagram. We could notice the power source, four 1.5 volt batteries connected to the 12V power pin of L298 Motor Drive and ground of Motor Drive and Arduino UNO. This supplies essential power to the circuit. A total of 6 volts is being supplied to this system, where the maximum permissible amount is 12 volts. Digital wires of Arduino are connected with the input1, input2, input3 and input4 of the motor drive. Motors are connected to the either sides of Motor Drive which are the outputs terminals. To complete the power source circuit, 5V of Motor Drive is connected to Vin power pin of Arduino UNO. Followed by this, HC05 Bluetooth Module’s Vcc is connected to 5V pin of Arduino UNO, which supplies power to Bluetooth Module. Ground to Ground connections are also made. Transistor logic pins, Transmitter (TX) and Receiver (RX) of Arduino UNO are connected to RXD and TXD of HC05 respectively. The program is uploaded to Arduino before connecting the Bluetooth module.

After all successful connections, switch on the power source. Lights at Motor Drive, Arduino UNO and HC05 would indicate the correct connection. Upon successful connection of your Bluetooth module with any android device, we could control this device. By passing the command, for example, to move forward we pass ‘F’. This command is transmitted by our device to Bluetooth module, which in turn transmits to Arduino UNO. Arduino receives is and passes the same to Motor Drive through its digital pins. Motor Drive will get this through their input pins and exercise them through their output pins were motor is connected.



*Arduino Source Code*

*#include<SoftwareSerial.h> //header file*

*SoftwareSerial BT(0,1); //(tx,rx)*

*String readdata;*

*void setup()*

*{*

*// put your setup code here, to run once:*

*BT.begin(9600);*

*Serial.begin(9600);*

*pinMode(6,OUTPUT);*

*pinMode(9,OUTPUT);*

*pinMode(10,OUTPUT);*

*pinMode(11,OUTPUT);*

*}*

*void loop()*

*{*

*// put your main code here, to run repeatedly:*

*while(BT.available()) //if bluetooth is connected to a device*

*{*

*delay(10); //power off for some time to stable the bluetooth*

*char c = BT.read(); //read the value for raddata so that it can change*

*readdata += c; //(forward,backward,left,right,stop)*

*}*

*if (readdata.length()>0)*

*{*

*Serial.println(readdata);*

*if (readdata == "forward")*

*{*

*digitalWrite(6,LOW);*

*digitalWrite(9,HIGH);*

*digitalWrite(10,LOW);*

*digitalWrite(11,HIGH);*

*delay(100);*

*}*

*else if (readdata == "backward")*

*{*

*digitalWrite(6,HIGH);*

*digitalWrite(9,LOW);*

*digitalWrite(10,HIGH);*

*digitalWrite(11,LOW);*

*delay(100);*

*}*

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

*{*

*digitalWrite(6,HIGH);*

*digitalWrite(9,LOW);*

*digitalWrite(10,LOW);*

*digitalWrite(11,HIGH);*

*delay(100);*

*}*

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

*{*

*digitalWrite(6,LOW);*

*digitalWrite(9,HIGH);*

*digitalWrite(10,HIGH);*

*digitalWrite(11,LOW);*

*delay(100);*

*}*

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

*{*

*digitalWrite(6,LOW);*

*digitalWrite(9,LOW);*

*digitalWrite(10,LOW);*

*digitalWrite(11,LOW);*

*delay(100);*

*}*

*readdata=""; //reset the value of readdata*

*}*

*}*

*Mobile Application*

MIT App Inventor-

MIT App Inventor is an intuitive, visual programming environment that allows everyone even children to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes. And what's more, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments.

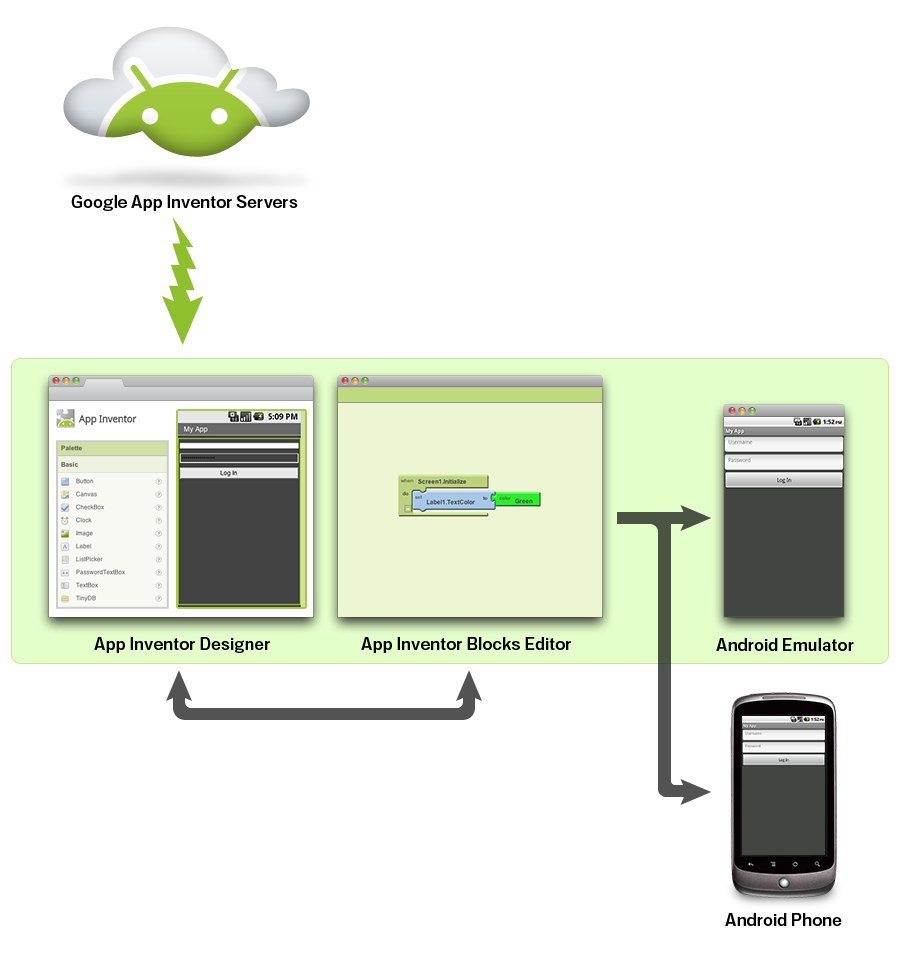
The MIT App Inventor project seeks to democratize software development by empowering all people, especially young people, to move from technology consumption to technology creation.

We build apps by working with:

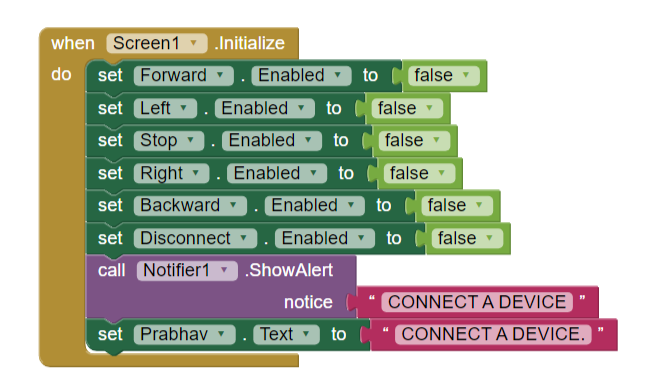
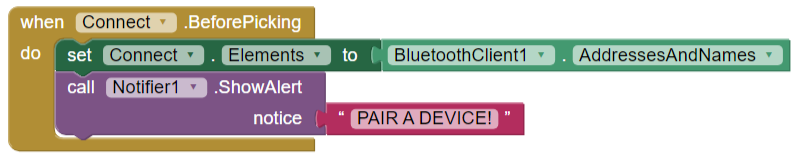
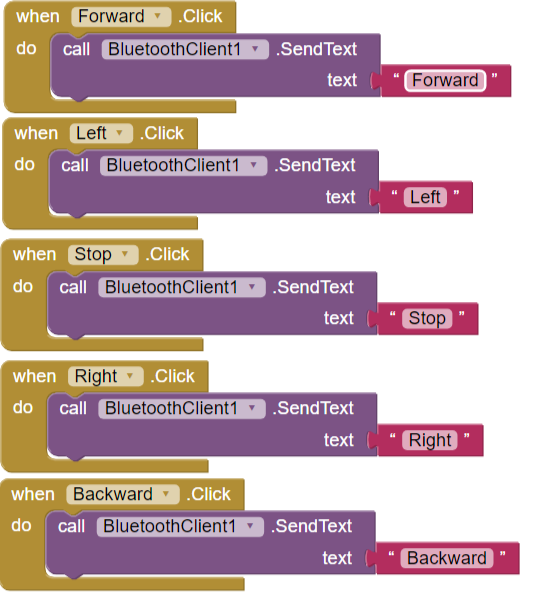
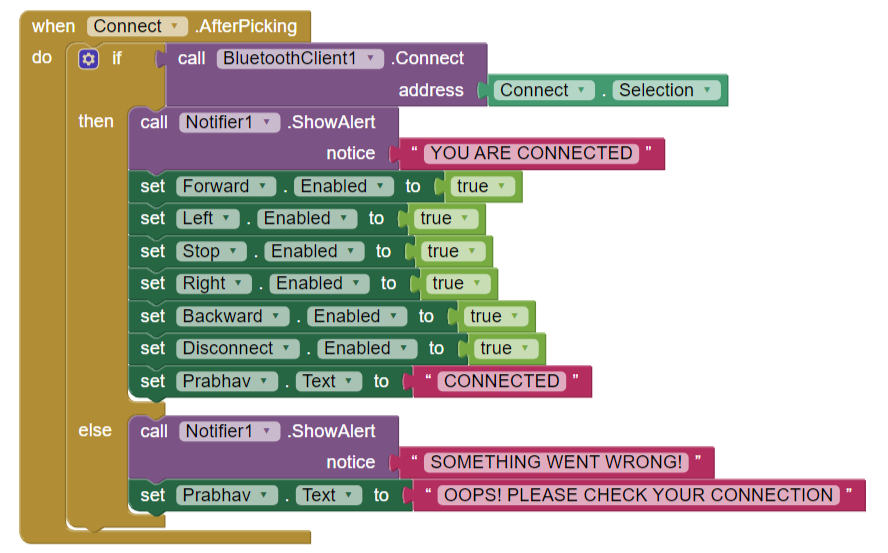
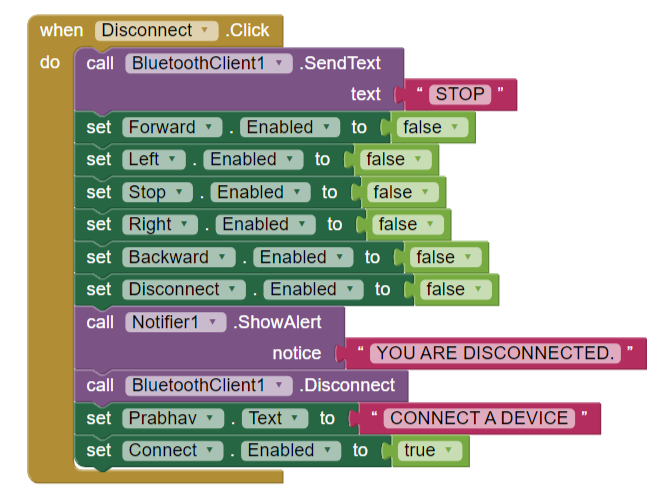
* The *App Inventor Designer*, where we select the components for our app.
* The *App Inventor Blocks Editor*, where we assemble program blocks that specify how the components should behave. We assemble programs visually, fitting pieces together like pieces of a puzzle.

Our app appears on the phone step-by-step as we add pieces to it, so we can test your work as you build. When we're done, we can package our app and produce a stand-alone application to install.

If we don't have an Android phone, we can build our apps using the *Android emulator*, software that runs on our computer and behaves just like the phone.



*Mobile App Source Code*

**

***Explanation:***

***Disconnect***

* *Upon clicking this a text alert is sent to the application saying STOP*
* *Rest of the functions i.e., Forward, left, right, backward, disconnect, stop, are disabled.*
* *Application alert YOU ARE DISCONNECTED is sent*
* *Notification to connect a device will now pop on screen.*
* *Connect button is enabled.*

***Connect***

* *Upon connecting application is notified with a message saying YOU ARE CONNECTED*
* *All of the functions i.e., Forward, left, right, backward, disconnect, stop, are now enabled.*
* *An Else condition is added, if the above steps don’t take place application is sent alert that SOMETHING WENT WRONG.*
* *A text is also sent to the application saying OOPS! PLEASE CHECK YOUR CONNECTION*

***Forward***

* *FORWARD text is sent to application to notify about the forward movement.*

***Left***

* *LEFT text is sent to application to notify about the left movement.*

***Stop***

* *STOP text is sent to application to notify about the stopping of all movement.*

***Right***

* *RIGHT text is sent to application to notify about the right movement.*

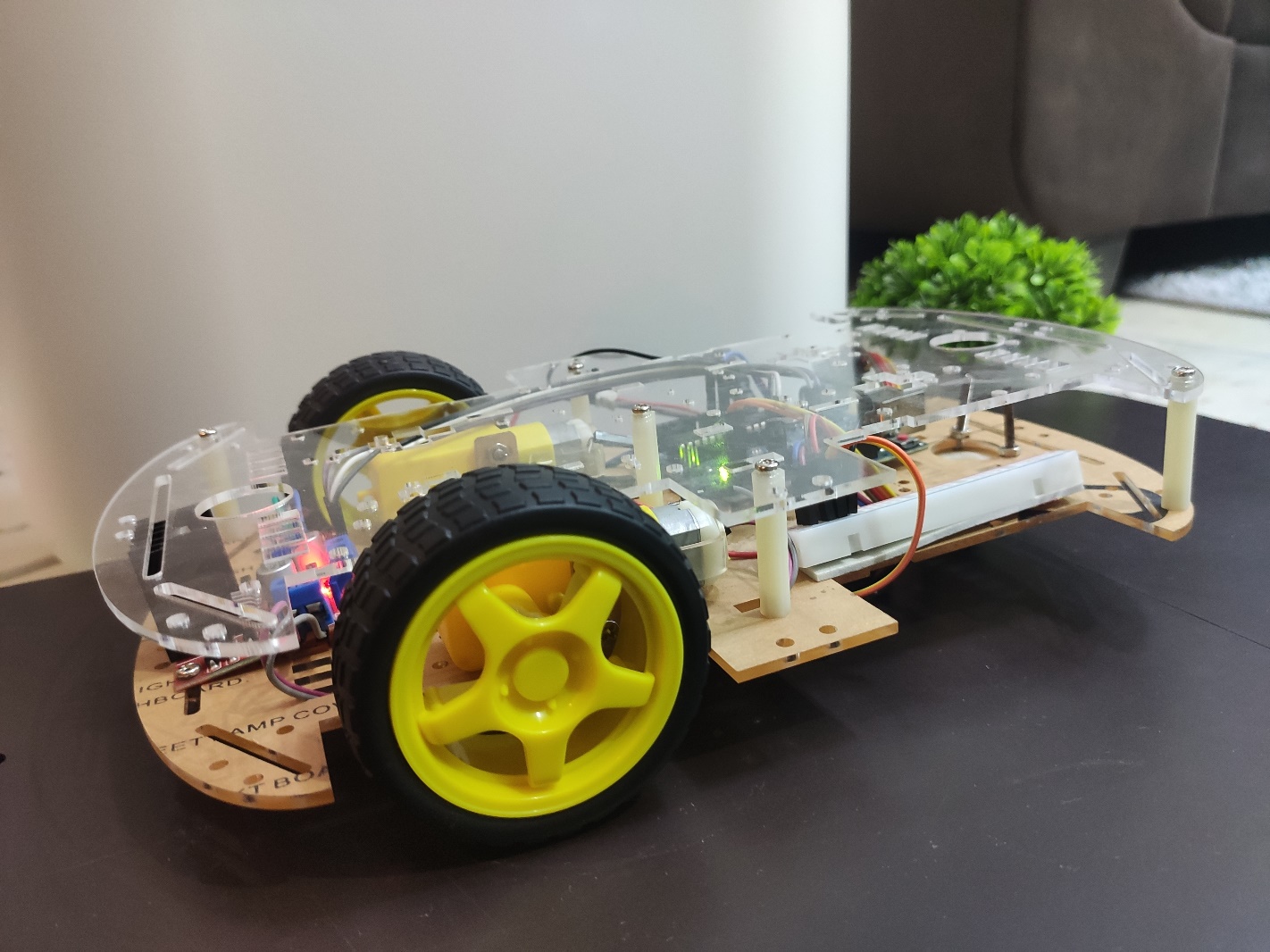
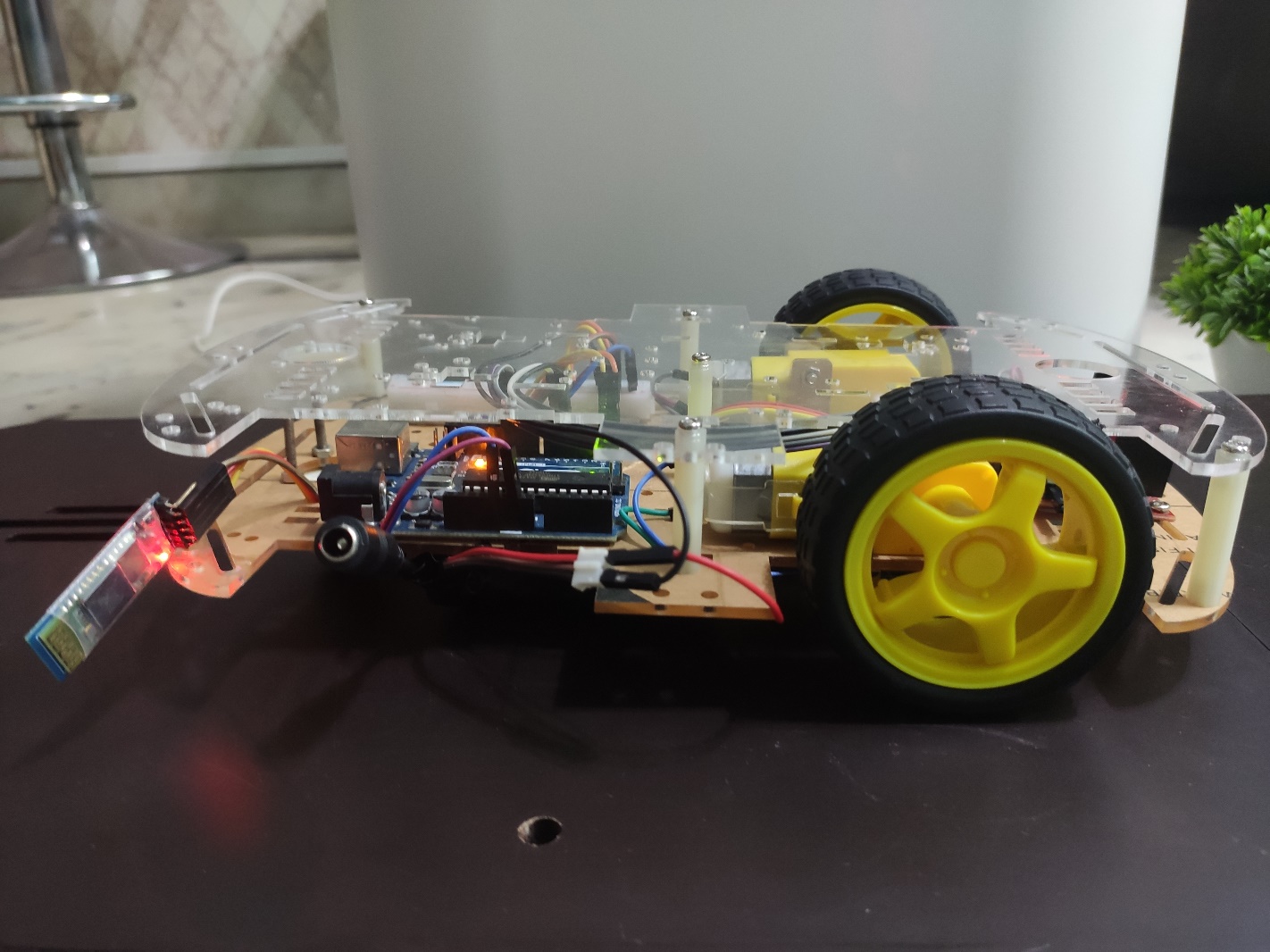
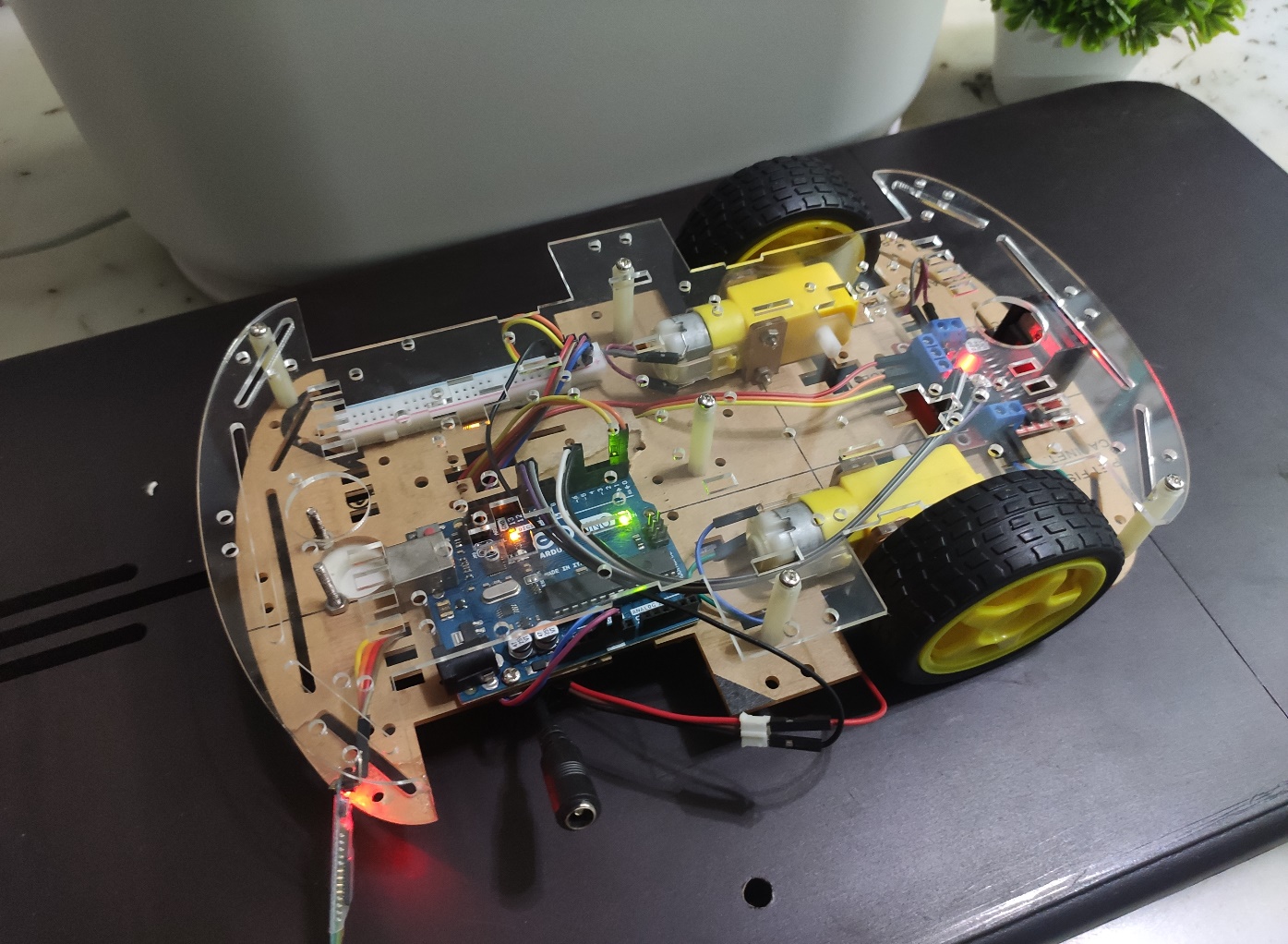
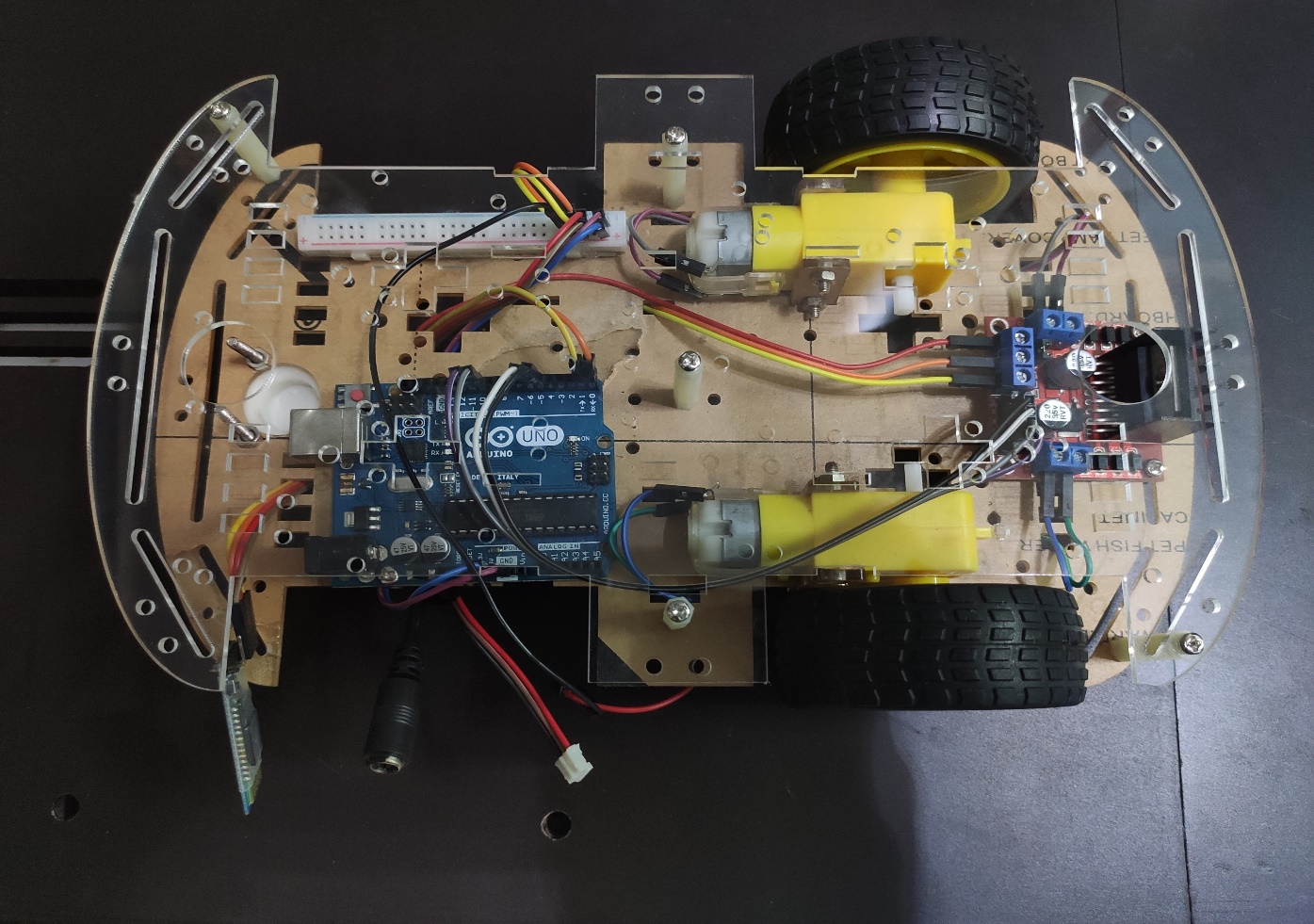
***Backward***

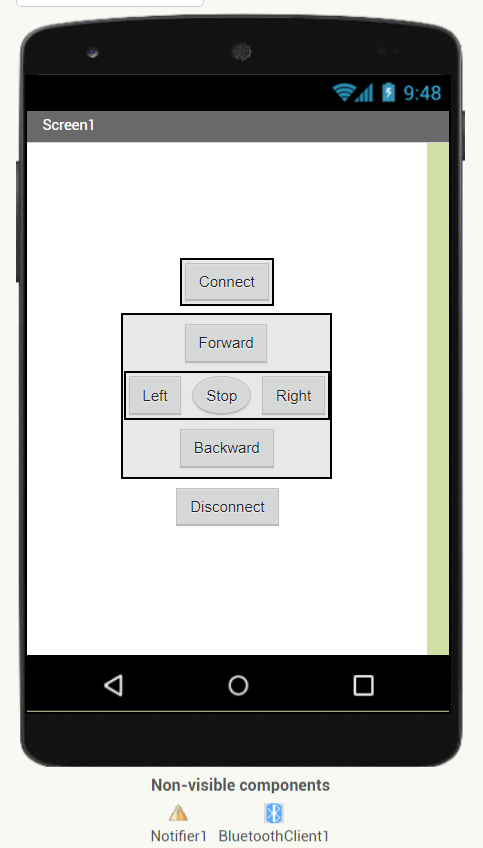
* *BACKWARD text is sent to application to notify about the backward movement.*

***Screen1***

* *All of the functions i.e., Forward, left, right, backward, disconnect, stop, are disabled.*
* *A notification is sent to the text and an alert as well saying CONNECT A DEVICE*

*Result*

**

**

*App Based Remote Control*

*Conclusions*

The final product we obtained is just the skeleton of those Remote Control Cars we see in the market. The mechanical design of this product is also proposed, which could be practically made to give a much better looking commercial product. For future plans, this product could be added with sensors like, accelerometer and humidity sensor, thereby widening their field of use.

The present product however could show some latency. The reason is, due to many connections and least power source of 6V, which result in loss of energy. So in future, we could use rechargeable batteries like Ni-Cd Battery or Li-ion battery that could avoid the present disadvantage.

Also, we could make use of this RC Motor Car as a surveillance system or rovers by adding a few more sensors and updating the code. This would make them into robots. These robots could self monitor under any human supervision, thereby reducing man power. These are just the alternatives, on which this project could be improvised and updated.

*References*

* + 1. [https://create.arduino.cc/projecthub/samanfern/bluetooth-controlled- car-d5d9ca](https://create.arduino.cc/projecthub/samanfern/bluetooth-controlled-car-d5d9ca)
    2. [https://create.arduino.cc/projecthub/JANAK13/bluetooth-controlled- car- 2c60e9?ref=search&ref\_id=Elelctric%20Car%20with%20Bluetooth%20c ontrol&offset=2](https://create.arduino.cc/projecthub/JANAK13/bluetooth-controlled-car-2c60e9?ref=search&ref_id=Elelctric%20Car%20with%20Bluetooth%20control&offset=2)
    3. <https://www.engpaper.net/arduino.htm>
    4. [https://www.shutterstock.com/search/car+drawing](https://www.shutterstock.com/search/car%2Bdrawing)
    5. [https://www.instructables.com/Smartphone-Controlled-RC-Car-Using- Arduino/](https://www.instructables.com/Smartphone-Controlled-RC-Car-Using-Arduino/)
    6. Getting Started with Arduino by Massimo Banzi
    7. Programming Arduino: Getting Started with Sketches by Simon Monk