

Controlling of a Toy Car using an Android Device (Motor Droid)

Abhi Vora¹, Vaibhav Chincholkar², Omkar Lad³, Ms. Avani Sakhapara⁴

I.T Department, K.J. Somaiya College of Engineering, Mumbai, India^{1,2,3}

Assistant Professor, I.T Department, K.J. Somaiya College of Engineering, Mumbai, India⁴

Abstract: The need for Unmanned Vehicles like drones and automatic driven cars is growing at an exponential rate. UAVs and other radio transmitted devices are used in battlefields and in the entertainment sector. In this paper we discuss building a cost effective system in which a user can remotely operate a toy car with the help of a Live Video Feed. The paper will also cover various modules on how the project was built and its Applications in the real world.

Keywords: Android, Arduino, Collision, Remote Control Car, Live-Video.

I. INTRODUCTION

In the current scenario, RC cars are restricted to the range of the Radio Frequencies and the drawbacks which come with it. To counter this problem, we have developed this project which will give the user unrestricted access to anywhere he/she wants to take the car. This prototype technology can be scaled and implemented in Military or Exploratory Missions. In our System, there are two Android devices, one with the user which will serve as a Remote (Parent) and the second will be placed on the car (Child) connected to the Arduino Board with the help of an On the Go (OTG) Cable. The communication between the Android devices will be done via the Internet thus solving the problem of range. The modules which will be explained in further detail ahead in this paper are Interface Module, GPS Module, Ultrasonic Module, Live Video Feed Module and Hardware Module.

II. LITERATURE REVIEW

Similar concepts to this idea have not yet been developed, although there are a few projects which have been undertaken which comprise of similar modules to that of our project. One such was a home automation system in which the home was controlled with the help of an Android device via Bluetooth or Internet [4]. This paper put forth a system architecture using REST ful web services as the interoperable layer. They managed to create a cost effective system which included all the basic features like alarms and hazards. Video Transmission is of high importance in our System, since the user will may not be present in the vicinity of the car. A paper discussed a similar concept which utilized Server – Client Architecture approach to display signals in real time. They have utilized Internet as the medium of communication citing that it's the fastest in the modern era. [7] To have a smooth transmission between the hardware and software and the integration into our project we referred to an IOT model in which Android based GUI was used to control electronic

devices such as Raspberry Pi. The prototype was made using hardware technologies and open source software, where several areas of computer applied, as: the development of applications, databases and computer networks [3] Since the car is to be driven remotely by the user, collision detection becomes an integral feature of the System. Multiple projects have been done in this field where sensors are attached onto automobiles and other vehicular applications which measure and utilize distance measurement. In this paper they have implemented GPS-GIS based system to measure highway alignment with the help of sensors and validate it with the actual data available. [5]

III. IMPLEMENTATION

In this Paper we are first going to discuss the overall operation of our model and then break it up into individual modules which will be explained in detail.

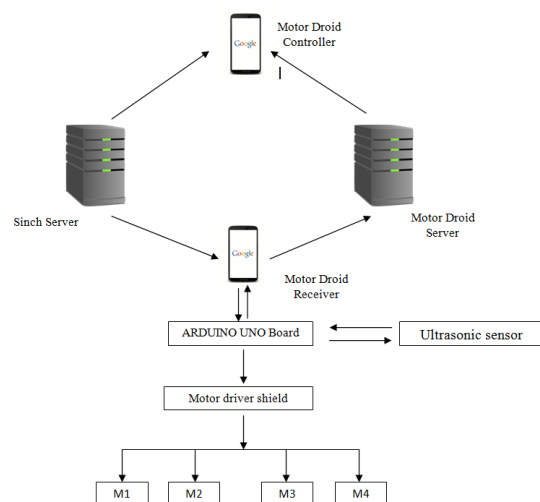


Figure 1. Architectural Diagram



Our main aim was to solve the problem of range. In the present scheme, the Radio controlled devices are restricted to about 100 feet, which is basically till the line of sight. In our proposed system, we implement Internet technology to connect the user and the car. This way the car can be operated from anywhere with internet access, Thus eliminating the problem of proximity.

In Figure 1, the architecture of the proposed system is depicted. The user starts by registering themselves onto the System by creating an account. Only after this can they login and utilize the service. Once logged in, the User initiates a call which needs to be received on the Child Application. Once this connection is established a Live Video Stream will be initiated and the user will be able to control the car. We are using Sinch Servers to facilitate the Video Stream. To control the car itself the user selects a direction (parameter) from the Parent application, which is sent to our dedicated Server. Our Child application continuously waits for new parameters and as soon as it encounters one on the server, it fetches it. This parameter is then passed onto the Arduino board which makes changes to the motors of the car via the Motor Driver Shield.

Now we are breaking down the system into specific modules: -

A. Interface Module

We have developed two Android Applications, one for the user and second to be placed on the car itself. The Users GUI consists of the navigation buttons which he/she will utilize to drive the car. It also has a view of the live feed which is captured by the car. The second action consists of the GPS location via the Google API which can be utilized to pinpoint the accurate location of the car.

B. Communication Module

This module is dedicated for transmissions between the Parent device (Remote) and Child device (Car). These devices have our Applications which communicate with each other using the HTTP Protocol. Once a direction is selected on the Parent Application, it is passed onto our Server. The Child Application continuously pings the server for any new parameters. As soon as it encounters one, the server passes on the parameter (direction) onto the Child device. These parameters are then transmitted onto the Arduino board with using Serial Communication with the help of an OTG Cable.

C. Live Video Feed Module

In our project Live Video is of utmost importance since the user may not be in the vicinity of the car itself. To initiate this call, the Parent device makes a call to the Child device's ID. To facilitate the Video Streaming, we have used Sinch Video Library version 3.9.8. To maintain the authenticity of the stream the video is encrypted with App-Secret Key provided by the above mentioned provider.

D. Geo-Positioning Module

In this module, we display the accurate real time location, which will be beneficial to the driver for further navigation of the car. To procure the cars current location we have used Google Location Services which will fetch the Latitude, Longitude after a predefined interval which in our case is 5 seconds and is stored on our server. Further when the user clicks on the "Maps" icon, the coordinates are fetched and displayed with the help of a marker on Google Map Fragment.

E. Collision Detection Module

To provide safety during the driving of the vehicle, we have implemented Collision Detection. We have achieved this with the help Ultra Sonic Sensors, in front of the car. This sensor measures the distance between the vehicle and an obstacle in direct line of the car. If an obstacle is encountered within 7 inches in front of the vehicle the car will come to a halt.

F. Hardware Module

This module behaves as an actuator in our System. It is made up of multiple components: -

- Arduino Boards – We use an Arduino Uno board on which we mount the Motor Driver Shield.
- Motors – There are four motors, the front motors connect to the front wheels of the vehicle which maneuver the direction and the back motors moderate the speed of the car.

IV. DISCUSSION

We have taken one example to simulate the working of our System with the help of a flowchart.

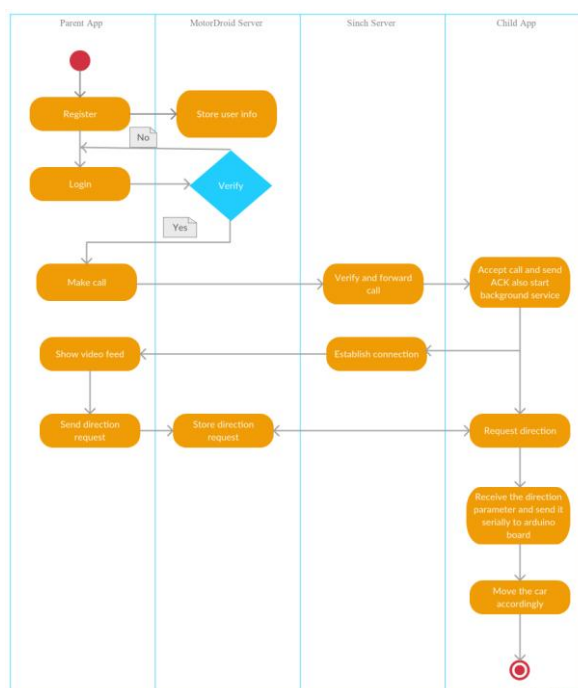


Figure 2. Flowchart of the System



Consider a new user who wants to use our System. He/she will first have to register themselves into our System. After doing so the user Log's in, only after which they will be able to make a call. Once the user makes a call from the Parent Application, the user will have to receive the call on the Child Application. After this connection is established, the user will have complete access to the car and its movement. The user will be able to maneuver the car with the help of the Live Video Feed which is being relayed from the Child application. Now consider that the user wants to take the car forward. Then the user will press the Forward button on the GUI. As seen in Figure 2, this parameter will first go onto the Server and then from that to the Child Application. This parameter is then passed onto the Arduino board via the OTG Cable. The Arduino board will then decode this parameter and move the back motors of the car in the Forward direction.

V. RESULTS

With the architectural diagram i.e. Figure 1 in mind we implemented the Android Applications, the Server Connectivity and Live Video Feed. Figure 3 – 7 shows the GUI of the Applications as well as features such as Live Video Feed in Figure 6 and the current GPS location of the car in Figure 7.

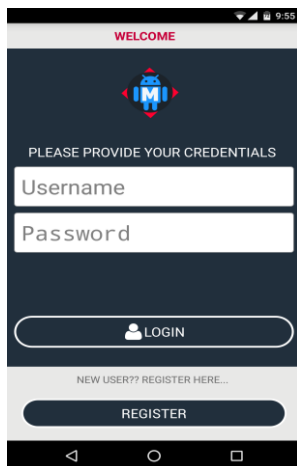


Figure 3. Login/Registration (Parent Application)

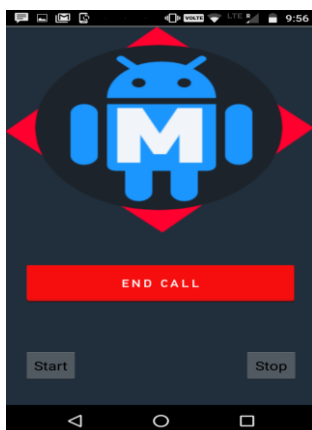


Figure 4. Graphic User Interface (Child Application)

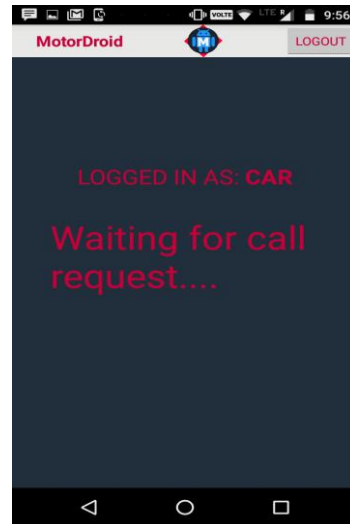


Figure 5. Child Application

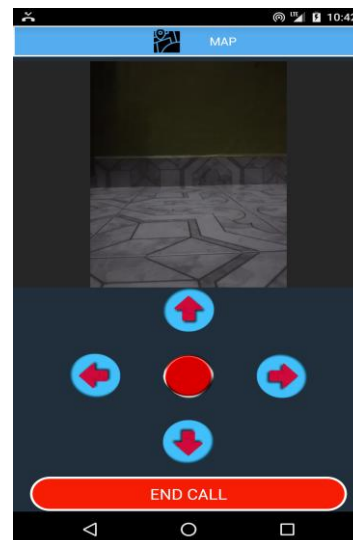


Figure 6. Live Video Feed as seen on the Parent Application

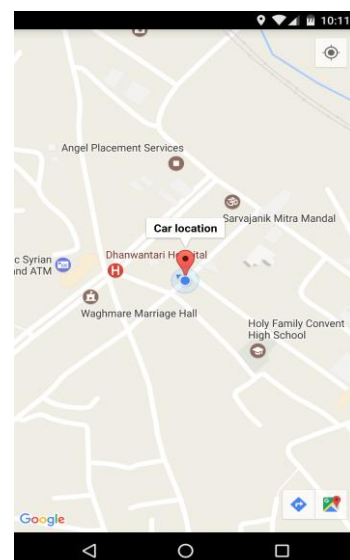


Figure 7. GPS Location



VI. CONCLUSION

In this paper we have successfully discussed a cost-efficient and more versatile alternative to the current Radio Controlled Devices. Our System solves the main problem of Range which is a limiting factor in the current scenario. But with our System, we have the ability to drive this car from hundreds of Kilometers away provided there is a strong Internet connection at both ends. This system can not only be used in toy cars but also can be scaled to be integrated in other unmanned vehicles such as rovers and drones.

REFERENCES

- [1] Prof. D. S. Vidhya, Miss Delicia PerlinRebello, Miss Cecilia Jane D'Silva, Mr. Linford William Fernandes, "Obstacle Detection using Ultrasonic Sensors." IJRST Volume 2, Issue 11, April 2016.
- [2] Edgar Ballesteros, "Android Management Tool, That Controls Electronic Devices Through A Raspberry Pi Under The IOT Model." IEEE, Computing Columbian Conference 2015.
- [3] R. Piyare, "Ubiquitous Home Control and Monitoring System using Android based Smart Phone," International Journal of Internet of Things, vol. 2, pp. 5-11, 2013.
- [4] Shiv Kumar, Seong Ro Lee, "Android based smart home system with control via Bluetooth and internet connectivity." Consumer Electronics, The 18th IEEE International Symposium 2014.
- [5] Imran, M. Hassan, Y. and Patterson, D. 2006. "GPS-GIS-Based Procedure for Tracking Vehicle Path on Highway Horizontal Alignments", Computer-Aided Civil and Infrastructure Engineering, 21 (2006), pp 383-394.
- [6] Vishal Kumar, "Mobile Web Server on Android Platform with Live Video Streaming" International Journal of Computer Science and Mobile Computing, Vol 4, Issue 4, April 2015, pg 561-564.
- [7] Malinowski, T. Konetski, B. Davis, "Web-controlled robotic manipulator using Java and client-server architecture." Industrial Electronics Society. The 25th Annual Conference of the IEEE, 1999.
- [8] <https://www.parallax.com/product/28015> (last access: 03/04/2017)
- [9] <https://www.sinch.com/docs/video/android> (last access: 03/04/2017)
- [10] <https://developer.android.com/guide/topics/connectivity/usb/host.html> (last access: 03/04/2017)
- [11] <https://www.arduino.cc/en/Tutorial/HomePage> (last access: 03/04/2017)
- [12] <https://developer.android.com/training/location/receive-location-updates.html> (last access: 03/04/2017)