

Wireless Network Controlled Robot using a Website, Android Application or Simple hand Gestures

Adarsh Modh^{1,*}, Mosam Dabhi², Lakshmi Narayan Mishra^{3,4}, Vishnu Narayan Mishra⁵

¹Department of Electrical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat-Dumas Road, Surat- 395007, India

²Department of Electronics & Communication Engineering, Sardar Vallabhbhai National Institute of Technology, Surat-Dumas Road, Surat- 395007, India

³Department of Mathematics, National Institute of Technology, Silchar - 788 010, District - Cachar (Assam), India

⁴L. 1627 Awadh Puri Colony Beniganj, Phase – III, Opposite – Industrial Training Institute (I.T.I.), Faizabad - 224 001 (Uttar Pradesh), India

⁵Applied Mathematics and Humanities Department, Sardar Vallabhbhai National Institute of Technology, Ichchhanath Mahadev Dumas Road, Surat – 395 007 (Gujarat), India

*Corresponding author: adarsh.modh@gmail.com

Received December 24, 2014; Revised April 05, 2015; Accepted June 11, 2015

Abstract This paper presents a robot which is controlled over a wireless network by using a laptop's web browser and accessing a webpage. The live video feed of the IP-camera (installed on the robot) can be viewed on this website. Thus with such minimal requirements, the robot can be controlled irrespective of the position of the user. Another method to control this robot is by using an android application in which the mobile format of the above website will be viewed and there will be virtual buttons created in the app so as to control the movements. This robot can also be controlled by analysing hand gestures by using an accelerometer.

Keywords: wireless network, control android, application embedded system, web application

Cite This Article: Adarsh Modh, Mosam Dabhi, Lakshmi Narayan Mishra, and Vishnu Narayan Mishra, "Wireless Network Controlled Robot using a Website, Android Application or Simple hand Gestures." *Journal Name*, vol. 3, no. 1 (2015): 1-5. doi: 10.12691/jcn-3-1-1.

1. Introduction

Robotics and automation has emerged out on a very large scale in the beginning of the 21st century. The main issues we all encounter in robotics are how to control the robot and what channel or medium should be used so as to get the expected results from it. Usually the most common ways that we have come across for controlling the robots are RF (radio controlled) and then by various sensing techniques like Infrared, Ultrasonic etc. [1,3,4,9]. Then there are also some advanced techniques like image processing where the robot will understand your movements; then voice recognition techniques where the robot responds by analysing what you say to it. But in most of these cases we have to control the robot by staying in some close range surrounding it which is a huge drawback.

Hence, in this research work a robot that can be controlled via a wireless network connection is presented. The most significant advantage of this system is that the user need not be in close range of the robot as the entire system is controlled online.

Compared to the other systems used [2,5,6,7,8,10] this system is so dynamic that the program can be edited online and also burned on the spark core through the wireless network. Since we are already aware that the technology of Wi-Fi has developed immensely over the

period of time and now with advanced communication systems it is not a complicated component as it used to be before. In Indian metro cities Wi-Fi connections are available at malls, various shops and even at homes. But in developed nations like US, UK, Japan etc. public places like parks and bus stops are facilitated with high speed Wi-Fi connections. Hence, in this research work the only condition of Wi-Fi connection is easily met.

A significant item from the equipment used is the Spark core. It is an integrated chip consisting of various other sub-components but out of them the two main components are the Wi-Fi module CC3000 and the ARM CORTEX M3 STM32F103CBT6 microcontroller. Its manufacturers have defined spark as 'An IoT (internet of things) operating system that runs in the Cloud'. In this research work the concept of 'internet of things' is implemented. In layman's language, IoT can be defined as the automatic communication and coordination of different machines that are connected to the internet. The advantage and the main difference of using this spark core from other methods (like simply interfacing a Wi-Fi module) is the spark cloud which is allotted to every core and is unique for each of them. This spark cloud is nothing but a virtual space or in other words it acts as a control room for our spark core. This control room (spark cloud) can be accessed from the internet at any corner of the world if the unique spark core ID and its access token (password) of that particular core is known. The spark core coordinates

with its own cloud through the Wi-Fi connection wherein it receives the commands that we have given via its cloud. After receiving these commands the corresponding operations are executed by the spark core.

The following **equipment** is required for the desired setup –

Table 1. Equipment List

no.	Component Name	Specifications
1	Spark core	Wi-Fi module + microcontroller
2	plastic chassis	With slots for four wheels, differential steering in front
3	4 wheels	Good grip and traction
4	two DC motors	300 rpm for rear wheels
5	low rpm DC motor	10 rpm for differential steering
6	voltage regulator IC 7805	Constant 5V output at max 0.5 A current
7	voltage regulator IC LM317	Constant 5V output at max 1.5 A current
8	Motor Driver IC L293D	12V output to run the motors, Speed regulator
9	IP-camera(internet protocol)	Model-Tenvis JPT3815W with night vision enabled
10	two Lithium polymer LIPO batteries	12V, 4400mAh
11	3-axis accelerometer	MMA7361L High Sensitivity (800 mV/g @ 1.5g)

2. Overview of the System

As it is shown in [Figure 1](#), the basic body of the robot is constructed by using a simple chassis which has four wheels (2 front wheels & 2 rear wheels) and three motors. The two wheels on the rear are connected to two of these motors which are of same speed (300rpm) and are used for accelerating in the forward or reverse direction. The 3rd motor is a very low 10 rpm DC motor which is mounted on the centre of axis of the front two wheels. This motor is then connected to a differential steering mechanism which is used to efficiently steer the robot. The clockwise rotation of this motor causes the robot to steer right and anticlockwise causes it to steer left.



Figure 1. Basic Structure of the Robot

After having some basic information about the spark core it can be discussed further how exactly the robot works. Each spark core has its unique ID and a spark cloud allocated to it. So what the spark core does is that it connects to the Wi-Fi and then it connects to its spark cloud. Now it is ready to accept instructions. The spark cloud has a unique way of working. We can write and compile the programs that we want to execute on the core through the cloud. All the programs can be burned directly

to our spark core from this spark cloud. Now to access and make changes in the programs in our spark cloud there is a prebuilt online API (application programming interface) developed by the spark community [14]. So the program can be written online on the API on a PC and then the code is transmitted to the core. (Note: for that to happen the core should also be connected to the cloud) This gives a huge benefit compared to other systems where we have to manually burn our code onto the microcontroller. Hence the programs can be edited even if we are far from our robot making our system dynamic in true terms. Regarding the language for our coding the easiest Arduino is compatible for the core. Their predefined libraries have to be used at numerous instances. Like for example if some of the functions are required to be available throughout the API (i.e. not just the core). These functions are declared as “Spark.function”. We will come to know later that why there is a need for such kind of functions in section ‘III’ part ‘a’. Initially when the spark core connects to the Wi-Fi, it should be done manually from a PC using the USB port that is available in the core. But once if the spark core is synchronised with a Wi-Fi network then we need not repeat the process again. Then only the core needs to be powered up and it will automatically connect itself to the cloud and it will start syncing with the instructions that have been given through the spark cloud.

There are numerous methods by which these instructions can be given to the robot. To control the robot, three methods are presented in this research work. As we can see in the below given flowchart, once the commands are given by the user via any of the three methods, they will be sent to the spark cloud. Here the cloud will recognize which one of the ‘Spark.function’ has been called and thus it will send the instructions of that function to the in-sync spark core which will then receive them by using its inbuilt Wi-Fi module CC3000 and perform the corresponding operations accordingly by using its inbuilt ARM CORTEX M3 STM32F103CBT6 microcontroller.

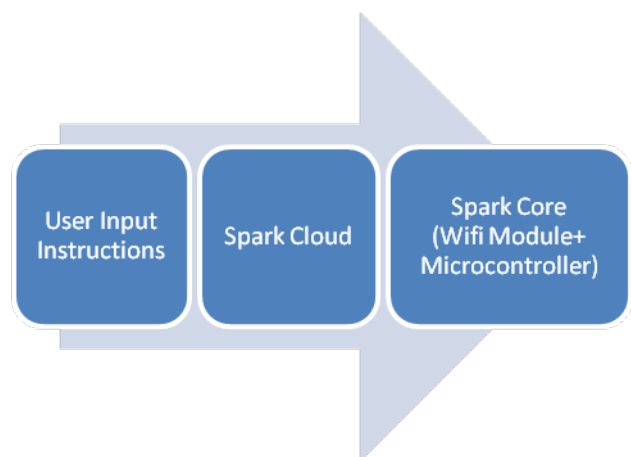


Figure 2. Flow of the instructions

This inbuilt microcontroller provides only an output signal voltage of 3.3 V (when it is at logic level 1) which is not enough for the motors to run. Thus we have used a motor driver IC L293D as well. [Figure 3](#) explains how the connections were made between the spark core and the motor driver IC. But let us say that our operations are very complex and we require more number of pin outs

compared to the ones that are provided on the spark core. In such a case we can interface an Arduino board or any other microcontroller (with more number of pin outs) with our spark core and then give the output signals from the controller to the respective motor drivers.

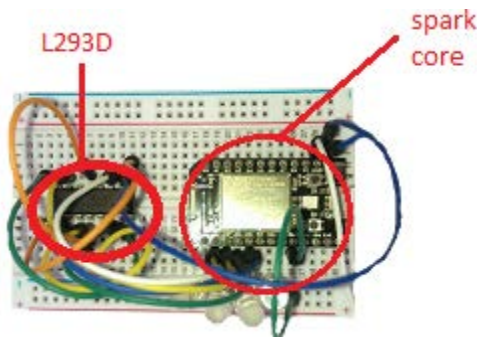


Figure 3. Circuit of Spark Core and Motor Driver IC

A 5V voltage regulator circuit is also used in our system. Since the entire setup of the robot is wireless, a regulated 5V voltage supply is essential at various places. For example- power supply to the spark core (3.6V to 6V), VCC to the Motor driver L293D and also 5V power supply to the IP-camera. Hence, a simple circuit having an input filter (10uF capacitor) and an output filter (100uF capacitor) with a voltage regulator IC 7805(max output current 0.5A) or LM317 (higher output current 1.5A) is designed. This circuit is used to convert the input voltage of 12V from the batteries into a constant output supply of 5V.

A very important part that is presented in this research work is about the IP-camera. Since, the user can be anywhere from the robot, it is very essential we keep a track of the position and the movement of our robot with respect to its surroundings. Due to this purpose an IP-camera is installed on our robot as we are already in range of Wi-Fi for the working of the spark core. An IP-camera connects to a Wi-Fi or a LAN connection and then transmits the live video feed to its own company website. As a result, various obstacles are avoided. This works better than most of the technologies for detecting obstacles like IR sensing or ultrasonic sensing. The IP-camera that is used has 2 rotational degrees of freedom. Hence there are two servo motors installed inside it, one of them is to move the entire camera about its axis (180 degrees) and the other one is to just move the lens part of the camera up and down (90 degrees). As a result though the robot may be on the ground it can see its surroundings and even the objects that are above the robot. This is also helpful if the robot has to perform complex functions rather than just simple surveillance (picking things up or mending things with a robotic arm). Another huge benefit of this IP-camera is that it has night vision enabled so even if there were no lights we can still navigate throughout the robot's surroundings.

3. ROBOT Controlling Mechanisms

For a user, it all comes down to how easily you can manoeuvre the robot for performing complex functions. Hence, the method that you use for controlling your robot becomes a crucial part of this system. In this research work three methods are presented.

a) Using a Laptop –

A very efficient and powerful method for controlling the robot is by using a laptop. Here the only thing required is an internet connection on the laptop (Wireless or LAN). This method is better than the other two methods that we have presented but it has only one drawback which is compactness. In today's emerging world of smartphones and tablets, laptops seem to be very bulky. But at the same time with laptops, better functionality is easily achieved.

In this method a simple website is created that we will access through our laptop's web browser. Now what happens is that it shows us the live video feed of the IP-camera on our website and the movement of the robot can be controlled by using our basic keyboard keys such as up, down, right and left arrow keys.

As we had mentioned earlier, that the IP-camera gives the live video feed on its company website. A basic embed program is implemented to stream that live video feed from the company's website on our website.

In this research work, we have attempted to keep things simple. And by this we mean to say that the way the user presses the keys of the keyboard to control the movement of the robot is kept very intuitive. While the user is on our created website he/she will get a similar experience of playing a car racing game.

As soon as the key is pressed, in the background a special type of http request is fired (for spark cloud). Whenever this kind of an http request is fired, some predefined function from the program is called that we had initially written on the API for the cloud operations.

E.g. of the special http request:
https://api.spark.io/v1/devices/53ff6e065075535142251687/led_1?access_token=eabf911dc1619fcb51d56d543b88ba6e318b4f8

Such requests consists of the device ID in the first part so that there is no confusion as to which device is the request meant for, then there is the access token which acts as a password protection for our system and then we give command to turn on the LED1 (which implies pin1) of the spark core and one of our cases will be executed. The access token is a very essential component as it makes our system secure and it ensures that no unwanted access is given to any external sources. In the example given above, the output is that the robot goes forward when the up arrow key is pressed. So as long as the up arrow key is pressed the robot will keep moving forward. But now the question arises that how the key press turns out to be an http request. The answer is the JavaScript program used on the website. So when the user accesses the website, the JavaScript takes care of the background functions that need to be performed when events like various key presses and mouse clicks occur.

Note that JavaScript always detects the key pressed; it counts pressing of a key as an event not releasing of the key.

In this case, four directional keys and two more keys for controlling the motion of the camera are used. Then there are some two or more keys which are optional if we are not going to perform any complex operations. In JavaScript each key on the keyboard has a unique event key code (e.key code). Hence when we press the key the respective 'e.key code' is detected and then whatever event has been defined under it is executed (in our case the http request).

Now the algorithm used in JavaScript is presented. An algorithm is required as there is a possibility that multiple keys will be hit since we want our controlling experience to be very intuitive (like a car racing game). For example- while we are going forward and we want to take a left turn we will press the up and the left key together. Since we will be always going either forward or backwards, what we have done is that we have defined separate governing cases for the up key as well as the down key and then there are sub cases inside them (up, up + left, up + right and down, down + left, down + right). Hence, it will first check the condition for either up or down key and then it will check the sub condition for right or left key inside that condition and then it will fire the http requests for both the conditions simultaneously. It will stay in this subcase and keep firing the respective http requests as long as the combination of keys is kept pressed and when it is released it will resume to normal forward or reverse motion.

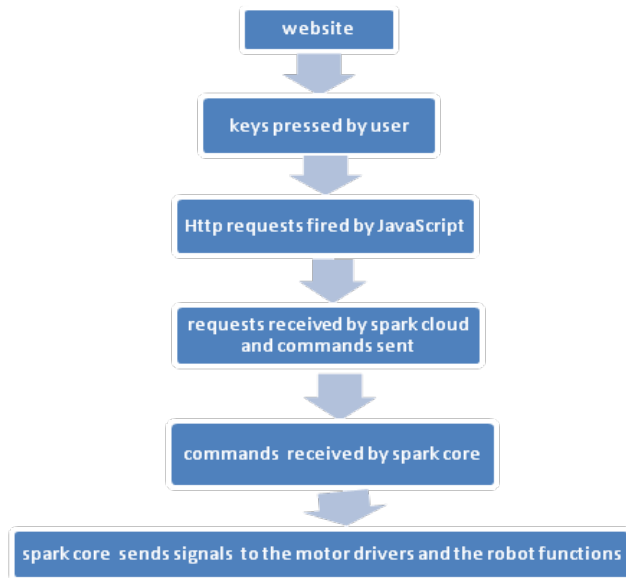


Figure 4. Flowchart explaining the step by step process of the entire website system:

The pseudo code for a case (up+right) is presented over here-

```

$(document).keyup(function(e)           //governing
condition
{
if(e.keyCode == 39 && right)
//checking sub condition
{
$.post("https://api.spark.io/v1/devices/53ff6e06507553514
2251687/led_4?access_token=eabf911dc1619fcba51d56d
543b8bba6e318b4f8",
//firing the http request to the cloud
function (data)
{

console.log(data["return_value"]);

if(data["return_value"] != 0)
{

$.post("https://api.spark.io/v1/devices/53ff6e065

```

```

075535142251687/led_4?access_token=eabf911dc1619fc
ba51d56d543b8bba6e318b4f8");
}
});
console.log("right up");
right = false;
}
}

```

Hence when these multiple http requests will keep getting fired to the spark cloud, the cloud will constantly send the corresponding instructions accordingly to the spark core to move the bot in various directions.

b) Android Application:

It is a bit similar to our previous approach. Here an android application is made which first of all opens the website that we had designed in the above method. But the website needs to be made android compatible (mobile format), and then only it will give suitable results. After this, lower resolutions (due to smaller screen) of the live video feed needs to be embed so as to stream it correctly on our application. Now for convenience, the inbuilt android keyboard is not used for controlling the robot. Instead on the same page on which we are getting the live video feed in the application we will keep some virtual keys which on pressing will do the same job of firing the http requests to the spark cloud. Rest of the things will mostly remain unchanged. Hence, this is a compact and a handy method for controlling the robot. But we need to compromise on the number of operations that we can perform with our robot.

c) Using simple hand gestures:

This is a very different method than the other two. In this method we will not use any website or any application instead we will use a 3-axis accelerometer (MMA7361L) to recognise the hand gestures and one more spark core. This extra spark core will remain with the user which will take analog readings through its ADC input pins from the accelerometer and then analyse them by using the inbuilt ADC (analog to digital conversion) of the STM32F103CBT6 microcontroller. This second spark core will be programmed in such a pattern that after the analysis from the ADC it will decide the direction in which the robot has to move. Let us say that the accelerometer is kept on the palm of our hand. So we want that if we tilt our palm in the direction to the right then robot should move right. Hence the readings of one of the axis of the accelerometer will increase in a particular pattern. This can be analysed by the ADC and then suitable functions can be called from the program of the spark core which will in turn send commands to the spark cloud and thus the other spark core (one on the robot) will receive its instructions accordingly. Improvements can be made by using a gyro sensor which provides 9 degrees of freedom and thus we can have more functions being carried out by detecting more number of gestures.

4. Conclusion & Applications

In this paper, a system is presented in which can very efficiently navigate and explore various areas where human reach is not possible. Here we have also explained various different methods to control our robot. Hence, we

may choose the system according to our needs and limitations.

This system can have a wide range of applications wherein with some safety modifications we can even send this robot in inhumane conditions. For houses and offices this can be used as a basic navigation and security purposes and it can also do more dedicated specific work if a robotic arm is added to our setup.

In military services, the mainframe of the body can be protected by using strong materials to construct it. If more number of peripherals are added to our robot, the system can be used for advanced operations, taking readings of the environment around the robot and many such things. Hence, we can see that this system can be conveniently developed into a high class drone. It can also act as a spy

robot and then sent into the enemy territory to give us information and data regarding their defence forces.

Also during disaster management (forest fires, earthquakes and floods), search and rescue operations can be very quickly done using such systems. Such systems can be used to move and navigate through the debris caused due to earthquakes and floods.

In industries too, such systems are very helpful. Small robots can be sent inside the enormous machines to locate the problems and issues inside them and also to repair the faulty parts inside the machine itself.

We have thus developed a fantastic system which is highly cost effective as compared to the substantial amount of operations it can perform.

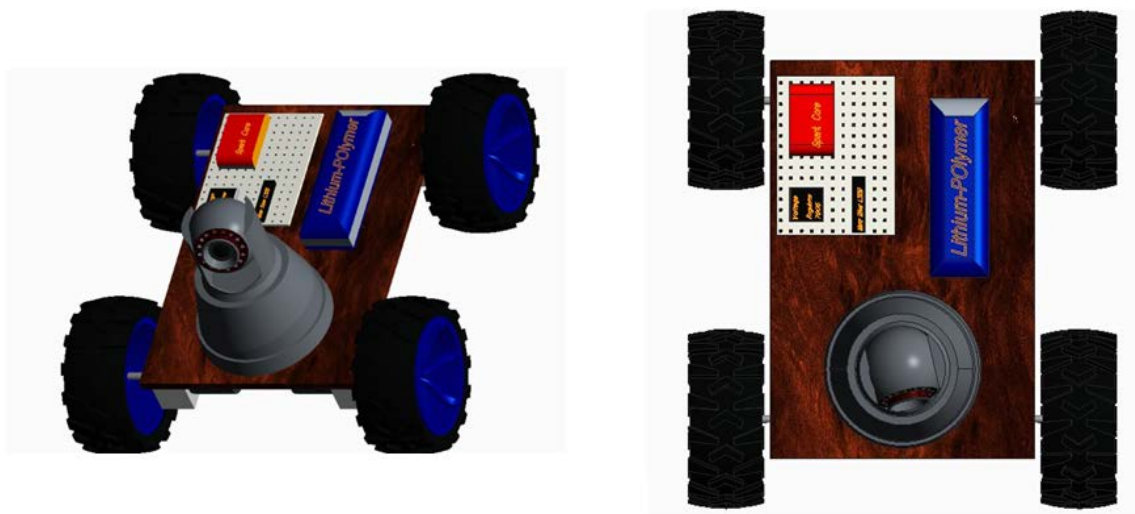


Figure 5. This is the design of the robot showing the front and top view. It was designed using the software PTC Creo. Hence, the whole 3-D model of the robot can be regenerated

Acknowledgments

The authors would like to express their sincere gratitude to the learned referees and Editors for their valuable and effective suggestions and supports which were essential for realizing and improvement of this research article.

The authors were assisted by Mr Samir Merchant and Mr Arun Kumaran in developing our website and also Mr Amol Shinde for making the design for the robot.

This work was supported by funds from a company Glowlogic Media Pvt Ltd, Mumbai. All the authors are thankful to all the editorial board members and reviewers of esteemed journal i.e. Journal of Computer Networks (JCN).

References

- [1] Nader Bargshady, Nayef A. Alsindi, Kaveh Pahlavan, Yunxing Ye, Ferit Ozan Akgul, "Bounds on Performance of Hybrid Wi-Fi-UWB Cooperative RF Localization for Robotic Applications".
- [2] Wenbin Wei, Yaodong Pan, Katsuhisa Furuta, "Internet-based Tele-control System for Wheeled Mobile Robot", Proceedings of the IEEE International Conference on Mechatronics & Automation Niagara Falls, Canada July 2005
- [3] R. Want, A. Hopper, V. Falco and J. Gibbons, "The Active Badge location system". ACM Transactions on Information Systems, 10, 91-102, January 1992.
- [4] M. Ocaña, L. M. Bergasa, M. A. Sotelo R. Flores, "Indoor Robot Navigation using a POMDP based on Wi-Fi and Ultrasound observations"
- [5] Hou-Tsan Lee, Wei-Chuan Lin, Ching-Hsiang Huang, Yu-Jih Huang, "Wireless Indoor Surveillance Robot", SICE Annual Conference 2011 September 13-18, 2011, Waseda University, Tokyo, Japan.
- [6] Wang Shaokun , Xiao Xiao , Zhao Hongwei, "The Wireless Remote Control Car System Based On ARM9", 2011 International Conference on Instrumentation, Measurement, Computer, Communication and Control.
- [7] Guoliang Zhong, Yukinori Kobayashi, Yohei Hoshino, Takanori Emaru, "Intuitive Teleoperation of NonHolonomic Mobile Robot with a manipulator based on virtual reality and Wi-Fi".
- [8] I. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle". IEEE Transactions on Robotics and Automation, 7(2): 193-204, 1991.
- [9] P. Bahl, and V. N. Padmanabhan, "RADAR: A, In-building RF-based User Location and Tracking System". Proceedings of the IEEE Infocom 2000, vol.2, pp. 775-784, Tel Aviv, Israel, March 2000.
- [10] M. Ocaña, L.M. Bergasa, M.A. Sotelo, "Robust Navigation Indoor Using Wi-Fi Localization", Proceedings of the 10th IEEE International Conference on Methods and Models in Automation and Robotics, pp. 851-856, Miedzyzdroje, Poland, 30 August-2 September 2004.
- [11] T. Fitzpatrick, Live remote control of a robot via the Internet, IEEE Robotics & Automation Magazine, Volume 6, Issue 3, 1999, pp. 7-8
- [12] <http://arduino.cc/en/Tutorial/HomePage>
- [13] Massimo Banzi – "Getting Started with Arduino"
- [14] <http://docs.spark.io/start/>
- [15] <https://github.com/spark/core/>