

WG11 Android based Surveillance Robot Control System Using Socket Programming with Implementation

Muhammad Gulfam¹ and Mirza Waleed Iftikhar Baig²

^{1,2}Department of Computer Science and Engineering, University of Engineering & Technology, Lahore, (RCET, Gujranwala Campus) Pakistan
¹2011csgulfam163@gmail.com, ²waleed08@yahoo.com

Abstract– This paper presents the mechanism of controlling a surveillance robot using android mobile devices through socket programming. User gives commands to the robot from his android device. The android mobile communicates with robot via local Wi-Fi network. The key for sending commands to the robot is the IP addressing. Overall system consists of a local Wi-Fi network, two android mobile devices and a robot. One of the two android devices is permanently associated with robot.

Keywords– WG11, Android, Robot and Wi-Fi Network

I. INTRODUCTION

Robots are becoming a very useful part of our lives. They are being used in almost every field of life and are working quite efficiently. One of the major reason for using robots is their accuracy in the job. They do the job with more precision than humans as they only do what they are programmed to do. As they don't feel anything and also they do not get tired, they do more amount of work than a normal human.

One of the very interesting areas of robotics is their use in surveillance. Practically surveillance is a dangerous area for humans and for this job robots are more suitable. Some places are not suitable for humans to go and get information from there. There might be life threatening creatures present or that place may be a war zone etc. So in such places robots do the job with more efficiency for us.

Smartphones are becoming more and more popular these days. Almost everyone is holding a smartphone now a days. They have changed our life style and have become the essential part of our lives. Due to improvements in operating systems these smartphones are very easy to use even for a common person. Android is the biggest player of this smartphone industry. In 2014 more than 80% of the smartphone market was in hold of android. Being open source, it provides a good facility to the developers to develop applications and even enhance its operating system.

As android is such a flexible smartphone we can use it in lots of our jobs. In this paper we have implemented surveillance robot control through android mobile over the local Wi-Fi network. We have selected Wi-Fi over the

Bluetooth due to relatively wider range of Wi-Fi network as compared to Bluetooth. Besides in case of Bluetooth, we have low range of bandwidth. While in case of Wi-Fi, we have comparatively higher bandwidth which provides us more speed in data communication and ultimately more efficiency. Moreover using IP addressing will ensure our data to reach at the required address. The reason is in the network, each host is given a unique IP address and no other host has the same IP address at the same network. It makes the identification of required object on the network lot easier. After receiving each command robot send an acknowledgment back to the android mobile which ensures the successful arrival of the data at the destination and this functionality makes the control system more reliable.

II. EXISTING WORK IN CONTROL SYSTEMS

Basically control systems are used to control physical processes. Normally a control system is composed of a few components like sensors, control units and networks etc. and the whole setup is completely used to do the job. Usually industrial control systems are found in hierarchies. In the hierarchy, at each stage there are different types of components used to do the job assigned to that stage. So in short each stage of the hierarchy do its job and summing up all makes the complete control system [1].

Nowadays control has become easier due to the use of technology as compared to the old days when the technology has not yet evolved enough. Now sending and receiving data from one place to another is lot easier and this has been achieved by adding some technical components to the control systems. Right now due to networking, technology has become more mature but wireless communication still has to be improved a lot as it has some problems to be dealt with, like noise, fading and jitters etc [2].

Control of multi-vehicle system became very interesting research area back in 1980s.especially introduction of wireless technology and then its advancements made this area more interesting for researchers, even new areas were explored like unmanned aerial vehicles (UAV) [3].

Nowadays it is possible to control your desktop computers through your smartphones geographically, independently through internet. For this, one just has to make his desktop pc

as remote desktop client. This remote desktop client will be running a server and will also be connected to the internet for communication purpose. Behind the scene this is achieved using virtual network computing VNC. VNC is platform independent and communicate to any operating system running java. This is how you can remotely connect to your desktop PC. This is similar as screen sharing between two desktop computers [4].

Smartphones have become an essential part of life nowadays. They provide almost all types of functionalities that were provided by desktop computers back in time. In smartphones, android has become the biggest player and is playing a big role in technical field. It provides almost all types of functionalities that we need to do our technical jobs. One of the hot areas in which android is providing its applications are networking and remotely controlling of devices. In this area android can be used to even remotely control our desktop computers and to perform many tasks from the desktop computer by not directly interacting with those computers [5].

In these days, most of the appliances are coming with the embedded computers and remote controlled functionalities. For a common person it is hard to remember and be familiar with the interface of all these appliances as the interface is becoming more wide and complex due to more functionalities in these devices. At the same time smartphones are becoming more popular and everyone is becoming familiar with them due to their regular use in daily life and easy interfaces. Almost everyone has smartphone and using it in his daily routine. So it is more intuitive to provide the remote control functionalities of the home appliances on smartphones [6].

Disability is a quite serious issue in the world. A significant fraction of the world's population is facing different forms of disabilities. Disable people are unable to do so many jobs due to their disabilities. They need to communicate with technical devices but they can't. To sort out these problems brain machine interfacing considered as a solution. But this system has some problems yet. We can add smartphones in these brain machine systems and it would be quite helping and problem solving for these systems as smartphones can provide computational facilities and storage facilities too [7].

Bluetooth was invented by telecom vendor Ericsson in 1994. Later it was introduced in smartphones and proved to be an impressive technology for short range communication. A single Bluetooth host can communicate with almost 7 other Bluetooth hosts at the same time within approximately 8 meter radius. Adding Bluetooth and other such technologies to smartphones, made them quite useful and handy. And now controlling robots through smartphones have become very interesting area of research [8].

III. ANDROID-CONTROLLED SURVEILLANCE ROBOT SYSTEM

This system explains how we can control a surveillance robot using an android device. The system includes Wi-Fi router, robot and two android mobiles, one mobile for user with which he will send commands to the robot and the other

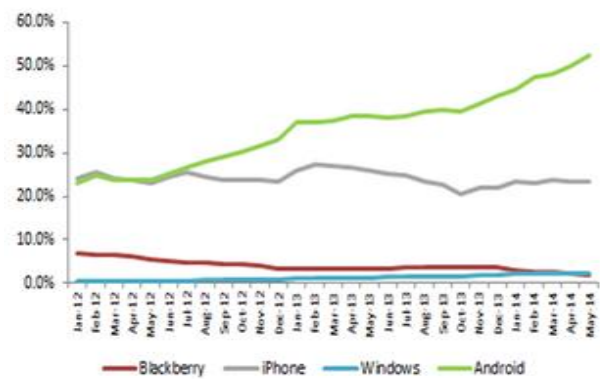


Fig. 1: Worldwide Smartphone Usage Share

android for robot to receive the commands from user's android and translates it for robot to operate it accordingly. Both the android mobiles are connected to a Wi-Fi router.

It allows them to communicate each other over the network. The reason for selecting Wi-Fi for this job is its range and bandwidth. The range of Wi-Fi is much greater than Bluetooth. The maximum range of Bluetooth is almost 32 feet. Whereas in case of Wi-Fi network the maximum range goes to almost 300 feet, which is almost 10 times greater than Bluetooth range. Besides, Wi-Fi technology provides more bandwidth than Bluetooth. That's why Wi-Fi is way faster than Bluetooth.

Whenever we connect to some network our computer or host is allocated a unique IP address. This IP address is the key for communication over the network. Another thing is port. The communication is based on both the ports and IP addresses. IP address tells the data about the destination host whereas on the host machines, these are the ports where application fetches data from.

As shown in the Fig. 2, IP address prevents data from getting lost on the network and to reach the required destination address and at that address, port number guides the data to reach at the destination application. Collectively IP address and Port makes Socket. Socket is like a pipeline and acts as medium for the data to travel.

In this control system first both androids are connected to network (Wi-Fi router) and then connection is established between both androids through sockets. The initiation of the process is done by the user's android as it is sort of master host in the communication. Once the connection is established system is ready for communication. User's device sends commands to receiver android where the receiver android receives commands and after translating it for robot it sends back the acknowledgment to the user's android device. The main purpose of the acknowledgment is to make the communication reliable. So user may be satisfied that commands are being received by receiver's android device and communication is carried out without any data loss.

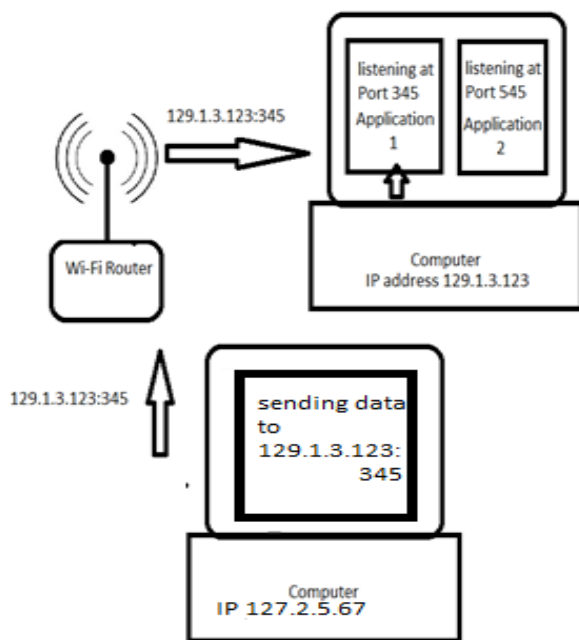


Fig. 2: Data Communication between Two Hosts over the Network

Here is the set of commands along with their semantics:

Semantic: command

1. Move forward : movF
2. Move backward : movB
3. Move Rightward : movR
4. Move leftward : movL
5. Capture image : captImg
6. Start recording : sRec
7. Strop recording : stpRec
8. Reset : rst

First four commands are for motion of the robot, as their name indicates. Move forward will tell the robot to take one step in forward direction. To make robot to move in forward direction more than one steps, user will have to send as much move forward commands as the number of steps he want the robot to move in forward direction. Same are the rules for other three move commands. Next is capture image command which will tell the robot to take the picture of the scene and store it in its own memory. Start recording command will tell the robot to start capturing the video. While the robot is capturing video it will store it in its own memory. While for the stop recording command robot will stop the video recording and will save the video in its internal memory which later on can be retrieved by the user when he gets the robot back. Otherwise during the job all the data that robot collects will be stored in the internal memory of robot. As the robot is storing all the data in its internal memory therefor it is a bit risky. Because during the surveillance, if the robot gets caught by the enemy, the enemy will get the information about us. At least enemy will know about which type of information about our enemy we are interested in.

This can be risky for us. So as solution for such problem, we have set a reset command. Reset command will not only shut the robot down but before shutting it down it will also erase all the data from the internal memory of robot. In this way we will be able to save our secrets.



Fig. 3: System diagram

IV. IMLEMENTATIONS

We have implemented this idea to test all pros and cons of this system. All the communication is between both android devices, one with robot and the other in user's hand.

Here is the code for user android application. The job for user android's application is to take input from user and send commands to robot's android according to the user's input. For user input, we have provided buttons for each command in the android application. Pressing any button will send the relevant command to the robot android. For start and stop recording, we have only one button because at a time only one of these two tasks can be performed.

```
private Socket socket;
private String ackMessage;
private PrintWriter outputSocket;
private InputStreamReader inputStreamReader;
private BufferedReader bufferedReader;
```

```
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
    new Thread(new ClientThread()).start();
```

```

Thread.sleep(1000);
outputSocket = new PrintWriter(new
BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()))
,true);
new Thread(new AckListener()).start();
}

class ClientThread implements Runnable {

    @Override
    public void run() {
        try {
            InetAddress ROBOT_ANDROID_ADDRESS =
InetAddress.getByName(ROBOT_ANDROID_IP);
            socket = new
Socket(ROBOT_ANDROID_ADDRESS,4444 );
        }
        catch (Exception e1) {
            e1.printStackTrace();
        }
    }
    ///////////////////////////////////////////////////
class AckListener implements Runnable {

    @Override
    public void run() {
try {
        inputStreamReader = new
InputStreamReader(clientSocket.getInputStream
());
        bufferedReader = new
BufferedReader(inputStreamReader);
        while (true) {
            if(socket.isConnected()){
                ackMessage =
bufferedReader.readLine();
            }
        }
        catch (Exception e1) {
            e1.printStackTrace();
        }
    }

    ///////////////////////////////////////////////////
public void btn_forwardCommand(View view) {
        try {
            outputSocket.println("movF");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }

    ///////////////////////////////////////////////////
public void btn_backwardCommand(View view) {
        try {
            outputSocket.println("movB");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    ///////////////////////////////////////////////////
public void btn_rightwardCommand(View view) {
        try {
            outputSocket.println("movR");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    ///////////////////////////////////////////////////
public void btn_leftwardCommand(View view) {
        try {
            outputSocket.println("movL");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    ///////////////////////////////////////////////////
public void btn_captureImgCommand(View view)
{
        try {
            outputSocket.println("captImg");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    ///////////////////////////////////////////////////
public void btn_startVideoRecCommand(View
view) {
        try {
            outputSocket.println("sRec");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    ///////////////////////////////////////////////////
public void btn_stopVideoRecCommand(View
view) {
        try {
            outputSocket.println("stpRec");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    ///////////////////////////////////////////////////
public void btn_resetCommand(View view) {
        try {
            outputSocket.println("rst");
        }
    }
}

```

```

catch (Exception e) {
    e.printStackTrace();
}
}

```



Fig. 4: User Android Application

When the user's android will start the application the first function that will execute is onCreate. The job for this function is to establish a connection with robot android and make the system ready to communicate for both sending and listening. All other button function will be executed every time when the relevant button will be pressed.

Now here is a part of code for robot android application. What this code part doing is to keep waiting for a client to connect at port 4444 and when it connects, then keep listening on that port until the connection is disconnected.

```

String message="";
ServerSocket serverSocket;
Socket clientSocket;
InputStreamReader inputStreamReader;
BufferedReader bufferedReader;
PrintWriter outBuffer;

Thread serverThread=new Thread(new
Runnable(){
    public void run(){

try {
    serverSocket = new ServerSocket(4444);

    text.setText("Server 4444");
    clientSocket = serverSocket.accept();

```

```

        inputStreamReader = new
        InputStreamReader(clientSocket.getInputStream()
        ());
        bufferedReader = new
        BufferedReader(inputStreamReader);
        outBuffer = new PrintWriter(new
        BufferedWriter(new
        OutputStreamWriter(socket.getOutputStream())
        ,true));
    }
    catch (IOException e) {
        e.printStackTrace();
    }

while (true) {
    try {
        if(clientSocket.isConnected()){
            message = bufferedReader.readLine();

outBuffer.println("ack:"+ message+"\r\n");

if(message.equals("movF"))
    {
        //Move robot forward one step
    }
else if(message.equals("movB"))
    {
        //Move robot backward one step
    }
else if(message.equals("movR"))
    {
        //Move robot rightward one step
    }
else if(message.equals("movL"))
    {
        //Move robot leftward one step
    }
else if(message.equals("captImg"))
    {
        //capture image.
    }
else if(message.equals("sRec"))
    {
        //start video recording
    }
else if(message.equals("stpRec"))
    {
        //Stop video recording
    }
else if(message.equals("rst"))
    {
        //Reset the robot.
    }
else
    {
        //Error : Unrecognized command
    }
}
}
}

```

```

}}
catch (IOException ex) {
    ex.printStackTrace();
}
}
}
});
serverThread.start();

```



Fig. 5: Robot Android Application

Pre requisite for running this application is to connect the robot's android to a local Wi-Fi network so that it will be available to user's android for making connection and to start communication.

In both codes, we have used threads to execute different tasks simultaneously. Threads are most efficient and key tool for simultaneous execution of multiple jobs. Besides, using try catch will prevent our system from being crashed.

V. WHAT CAN BE DONE IN FUTURE

Right now we have implemented the idea for local network that is established using a single Wi-Fi router. Therefore the access is limited to the area that is in range of that local network. So right now we are bound to the limited use of this system. But in future we intend to replace the local network by internet. In that case we will not be limited for the use of the system. We will be able to control the robot from anywhere in the world.

VI. CONCLUSION

The surveillance is always has been a quite sensitive task. And it includes so many risks. So it's better to use robot for this job instead of people. And if you are able to control the robots with efficiency and accuracy then you can guarantee

yourself with good results and success. This system is a good step for secure surveillance using robots.

REFERENCES

- [1]. Alvaro A. C'ardenas, Saurabh Amin, Shankar Sastry "Research Challenges for Security and Control System", In proceeding HOTSEC'08 Proceedings of the 3rd conference on hot topics in security Article no. 6. University of California, Berkeley.
- [2]. ByJohn Baillieul, Fellow IEEE, and Panos J. Antsaklis, Fellow IEEE, "Control and Communication Challenges in Networked Real-Time Systems", Vol. 95, No. 1, January 2007.
- [3]. Richard M. Murray "Recent Research in Cooperative Control of Multi-Vehicle Systems" Submitted, 2007 International Conference on Advances in Control and Optimization of Dynamical Systems, September 2006.
- [4]. Ajit Kotkar, Alok Nalawade, Siddhesh Gawas, Aniket Patwardhan, Snehal Mangale "Android Based Remote Desktop Client", in International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Issue 2, April 2013.
- [5]. Anjumara Inamdar, Heena Aggarwal, Sayali Kadam, Mayuri Kadhane "COMPDROID -Remote Desktop Access through Android Mobile Phone", International Journal of Science and Modern Engineering (IJISME) ISSN: 2319-6386, Vol. 2, Issue 1, December 2013.
- [6]. Jeffrey Nichols, Brad A. Myers, "Controlling Home and Office Appliances with Smartphones", Carnegie Mellon University Research Showcase @ CMU 2006.
- [7]. Prashanth Shyamkumar, Sechang Oh, Nilanjan Banerjee, Vijay K. Varadan, "A Wearable Remote Brain Machine Interface using Smartphones and the Mobile Network", In Advances in Science and Technology (Vol. 85) chapter1: Smart Fabrics and Wearables.
- [8]. Ritika Pahuja, Narender Kumar, "Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller", International Journal of Scientific Engineering and Research (IJSER) ISSN (Online): 2347-3878, Vol. 2, Issue 7, July 2014.