Research in Communication Mechanism between PC's Bluetooth and Android Bluetooth

Santhosh B K [1], Mrs. Krithika K [2], Mrs. Sunitha K [3]
[1] Student, M. Tech in Computer Engineering, SJBIT, Bengaluru, India
[2]Technical Specialist, RBEI, Bengaluru, India
[3]Associate Professor, Dept. of CSE, SJBIT, Bengaluru, India
[1]Email-id: santhosh92ckm@gmail.com Contact no: +919036745822

Abstract— Cellphones and computers have become integral part of our life. Today, mobile phone is more than a communication tool, but an intelligent terminal which is used for various purposes. New technologies are arriving every year to make the operations much faster and simpler. In this regard new communication mechanism between Android and PC has been proposed. This paper describes the new Bluetooth communication mechanism between PC and Android and gives the detailed explanation about the implementation process through workflows. In other words, through .NET serial port communication realizing serial port communication from PC to an Android App and Android mobile delivers bytes of data to the PC's serial port through socket communication using RFCOMM.

Keywords— Cellphones; serial port; Socket; communication; Android; RFCOMM; Personal Computer; .NET; Bluetooth;

I. INTRODUCTION

In the recent years, with the development of mobile communication and mobile terminal, especially with the release of Android smart phone platform has enclosed or added the new vitality to the mobile space. Android is a open source mobile operating system based on LINUX which is completely open and interconnected platform for mobile devices. Android platform consists of user interface, middleware and application software. Bluetooth technology is a modified short range wireless communication technology. Around the globe the working frequency band of Bluetooth do not need a license. The benefits of Bluetooth are reflected in low price, easy to control, non-visual distance limitations. Bluetooth is the important feature of the smart phone, which is embedded in android platform, as android mobile network communication module. Bluetooth technology supports short range wireless transmission in universal way, by using unlicensed 2.4GHz short-range radio frequency bandwidth. Using Bluetooth, users can form the clusters of 8 maximum devices that form a star shaped network called piconet. Master is the main device and all other are slaves. Data between two Bluetooth devices can exchange data with the maximum speed of 2.1Mbps. Bluetooth uses radio technology called frequency hopping spread spectrum (FHSS), in order to lessen the interference between different piconet structures.

Bluetooth technology is used for number of purposes for example file transferring form one Bluetooth enabled device to another. Bluetooth chatting between two androids has also existing. This paper introduces a new concept for cross domain communication using Bluetooth where different devices that work on different platforms will be able to communicate with each other in real time. The two domains implemented here are Android JAVA and .NET. The main intention of this paper is to create channel for communication between Android and PC through Bluetooth, where Android phone will be able to communicate with PC in real time using simplest mechanism. The Physical structure of the proposed communication mechanism is as shown in the Figure 1, there are 2 applications in this mechanism one is running on the PC i.e., .NET application which uses virtual serial port of PC for communication and other is Android application running on Smartphone which will convey the bytes of data to the PC through Socket communication using RFCOMM protocol.

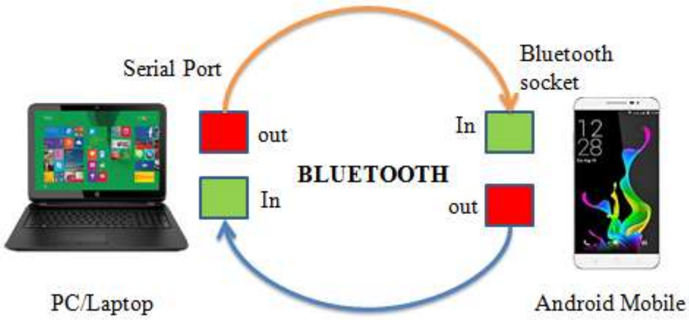


Fig 1. Physical structure of the communication mechanism

II. ANDROID BLUETOOTH[1]

Android can communicate with Bluetooth devices using Bluetooth protocol. If Bluetooth API were used, a variety of tasks like other Bluetooth device searching, local Bluetooth adaptor query for paired Bluetooth devices, setting RFCOMM channel, etc. could be performed.

The Android platform incorporates support for the Bluetooth network stack, which permits a device to remotely exchange data with other Bluetooth devices. The application framework offers access to the Bluetooth functionality through the Android Bluetooth APIs. These APIs let applications to remotely connect to other Bluetooth devices, empowering point-to-point and multi-point remote features.

Utilizing the Bluetooth APIs, an Android application can perform the following:

- Scan for the Bluetooth devices.
- Query the nearby Bluetooth adapter for paired Bluetooth devices.
- Establish Radio Frequency Communication channels.
- Connect to other devices through service discovery.
- Transfer data to and from devices.

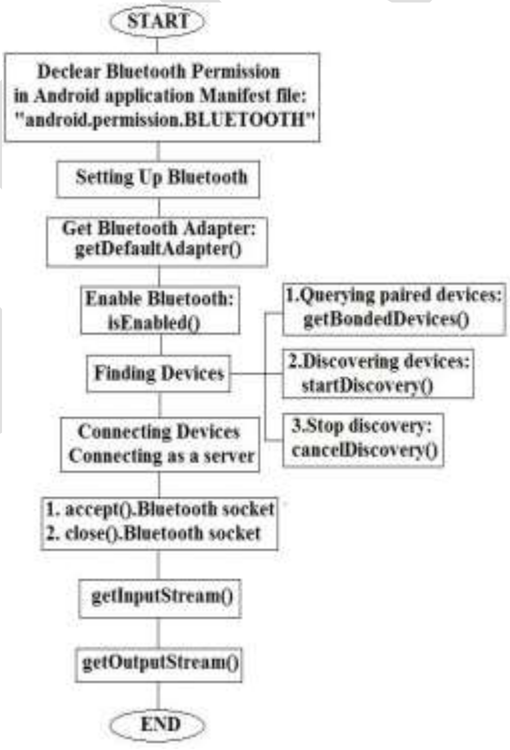
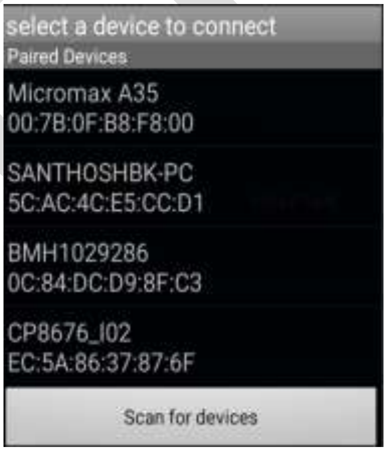

START
Declear Bluetooth Permission in Android application Manifest file: "android.permission.BLUETOOTH"
Setting Up Bluetooth
Get Bluetooth Adapter: getDefaultAdapter()
Enable Bluetooth: isEnabled()
1.Querying paired devices: getBondedDevices()
Finding Devices
2.Discovering devices: startDiscovery()
Connecting Devices Connecting as a server
3.Stop discovery: cancelDiscovery()
1. accept().Bluetooth socket
2. close().Bluetooth socket
getInputStream()
getOutputStream()
END

Fig 2. Program Flowchart for establishing Bluetooth Connection

The program flowchart that illustrates discovering for Bluetooth devices, establish a connection and then sends data to Bluetooth enabled PC is shown in the figure 2. As shown in the flowchart, we first get hold of the local Bluetooth adapter installed in the Android phone and establish a connection with PC's Bluetooth module. Once the connection is successfully established we then send data from Android to the PC.

To begin communication with Bluetooth, Android first looks for a permission which is required for requesting a connection, accepting a connection, and exchanging information. To discover local Bluetooth devices this permission is required. The Bluetooth Adapter is essential for all Bluetooth activity and it is called by the static getDefaultAdapter() method. This returns a Bluetooth Adapter that signifies the device's own Bluetooth adapter (the Bluetooth radio). If getDefaultAdapter() results null, then the Bluetooth is not upheld by the device. Now isEnabled () is called to check whether Bluetooth is right now enabled. The false output originating from this method makes the Bluetooth disable and to enable the Bluetooth once more, startActivityForResult() is called with the ACTION_REQUEST_ENABLE action intent which will sends a command request to enable Bluetooth in the system settings. Although the upcoming operation is to perform nearby device discovery, before that it is better to query the set of paired devices to see if the desired device is already known by calling getBondedDevices(). startDiscovery() method is used to start searching devices in the surrounding. This asynchronous method and can immediately return with a Boolean value, indicating whether discovering has successfully initiated. The window for selecting the paired devices to connect is illustrated in the figure 3.

Performing device searching is a power consuming procedure. With cancelDiscovery() method the search for another device can be terminated immediately, once a device is identified to connect. For connection establishment between two devices (PC and Android), one device must act as a server by holding an open BluetoothServerSocket. In this case PC acts as a server by creating virtual COM port. The goal of the server socket is to listen for incoming connection requests and when one is granted the request, give a connected BluetoothSocket. Now connection requests can be granted by calling accept() method which is a blocking call that will return when a connection has been established or an error has occurred. The approval of a connection request is depended on a correct UUID (Universally Unique Identifier) matching, the one which is registered with this listening server socket and for a successful matching, accept() will return a connected BluetoothSocket. Now for preventing other connections, call close() method which detaches the server socket and all its resources, but does not terminate the connected BluetoothSocket that's been returned by accept(). As the accept() method is a blocking, it should not be running in the main activity UI thread because it will prevent any other communication with the application. Using the BluetoothSocket, sending arbitrary data can be established by calling the InputStream and OutputStream which controls the transmissions via the socket, through getInputStream() and getOutputStream(), respectively.



select a device to connect

Paired Devices

Micromax A35
00:7B:0F:B8:F8:00

SANTHOSHBK-PC
5C:AC:4C:E5:CC:D1

BMH1029286
0C:84:DC:D9:8F:C3

CP8676_I02
EC:5A:86:37:87:6F

Fig 3. Window for selected device

Bluetooth Communication is based on unique MAC [6]. For the security before any Bluetooth communication, the devices must be paired. In order to create a connection between the two devices, both the server-side and client-side mechanisms must be implemented, because one device must open a server socket and the other one must initiate the connection (using the server device's MAC address to initiate a connection). The server and client are considered connected to each other when they each have a connected Bluetooth Socket on the same channel. At this point, each device can obtain input and out-put streams and data transfer can begin.

III. COMMUNICATION VIA BLUETOOTH

A. .NET SERIAL PORT[2]:

.NET platform is a powerful, collection of a wide assortment of programming environments, high productive application improvement platform outlined by Microsoft Corporation. Creators effortlessly understand the particular application advancement as per their own well known programming language by calling rich class libraries of .NET platform. To acknowledge serial port communication in between PC and gear in .NET platform, we require just use communication control gave by .NET. Through programming for the relating control's properties and events, we can execute serial port communications without much complexity.

B. ANDROID MOBILE SENDS BYTES OF DATA TO THE PC SERIAL PORT THROUGH SOCKET COMMUNICATION WAYS[3]

Socket is a sort of communication mechanism taking into account TCP/IP convention in PC in computer network communication. The application sends and gets information through an socket, similar to open a file handle for the application, to read and compose information to stable storage. By socket applications can be added to the network, and speaks with alternate applications which are in the same network. The information composed by the client program on android to the socket is fit for being perused by service application on PC.

Android is a mobile system based on Java platform, full support to socket communication mechanisms depends upon TCP and UDP for JDK environments. Since .NET platform not just with advantages in serial port communications not accessible in the Java platform, but also with intense features on the socket communication mechanism. We select Android as a socket client, and .NET platform as a socket server to build a communication between the Android mobile and the PC, accomplishing Android mobile to send bytes of data to PC serial port.

C. INITIALIZING BLUETOOTH COMMUNICATION BETWEEN ANDROID AND PC[2]

For Bluetooth communication between android and PC, it is advantageous to make serial Bluetooth connections between an android phone and a .NET application on a PC. This section depicts how to connect from an Android mobile to a .NET application running on Windows 7. The methodology of connecting an Android app to PC is as shown in Figure 4.

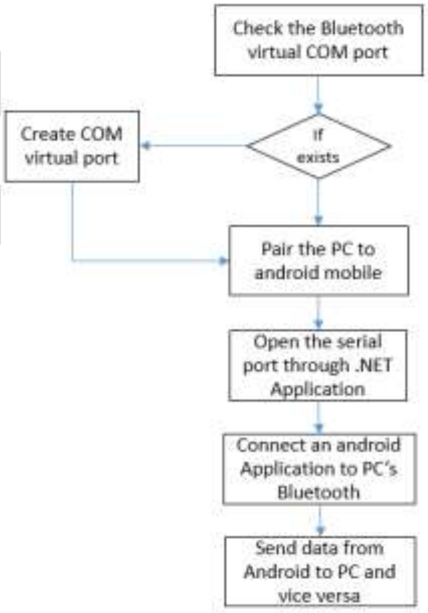
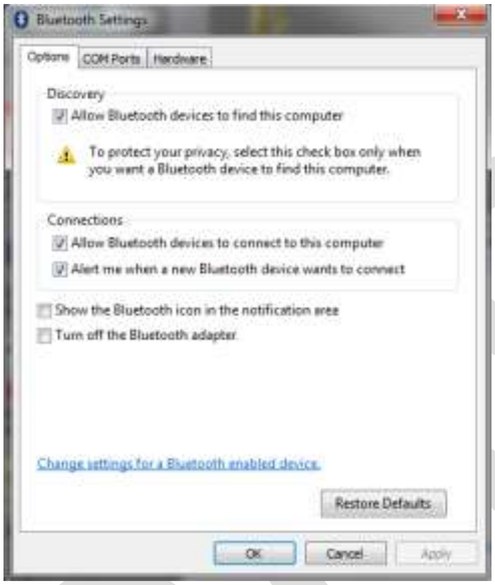

Check the Bluetooth virtual COM port
If exists
Create COM virtual port
Pair the PC to android mobile
Open the serial port through .NET Application
Connect an android Application to PC's Bluetooth
Send data from Android to PC and vice versa

Fig 4. Initializing Bluetooth Communication between Android and PC

Below are the steps to connect Android to PC

1) Enable Bluetooth on PC
2) Find Assigned COM Port
3) Check the COM port Baud Rate

4) Pair the Mobile Bluetooth to PC
5) Run the .NET application on PC
6) Connecting Android App to .NET Application
7) Send data from Android to PC over Bluetooth.



Discovery

☑ Allow Bluetooth devices to find this computer

To protect your privacy, select this check box only when you want a Bluetooth device to find this computer.

Connections

☑ Allow Bluetooth devices to connect to this computer

☑ Alert me when a new Bluetooth device wants to connect

□ Show the Bluetooth icon in the notification area

□ Turn off the Bluetooth adapter

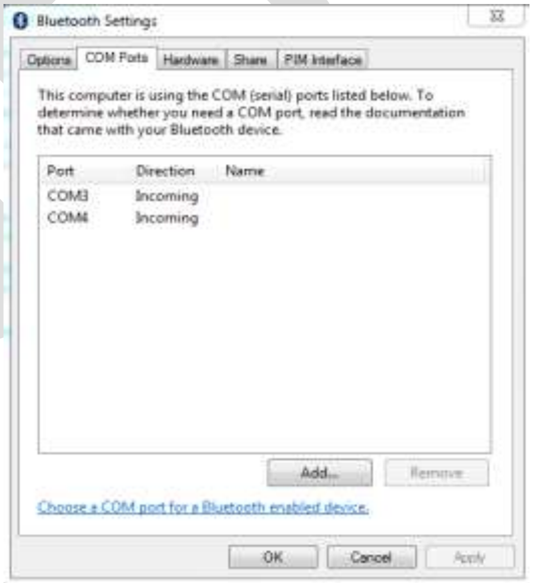
Change settings for a Bluetooth enabled device.

Fig 5. Bluetooth settings in PC

- **How to Find Assigned COM Port:**

a. To find out which COM port has been assigned to the Bluetooth serial link, click on the Windows Start button and search for “Bluetooth settings” and select “Change Bluetooth settings”. The window opened for Bluetooth settings is shown in Figure 5.

b. Click on the “COM Ports” tab to see which port has been assigned to a Bluetooth serial link. If No COM port displayed, then add new incoming COM port to the Bluetooth serial link which is shown in Figure 6.



This computer is using the COM (serial) ports listed below. To determine whether you need a COM port, read the documentation that came with your Bluetooth device.

Port	Direction	Name
COM3	Incoming	
COM4	Incoming	

Choose a COM port for a Bluetooth enabled device.

Fig 6: Adding new COM ports for Bluetooth

D. RFCOMM [4]

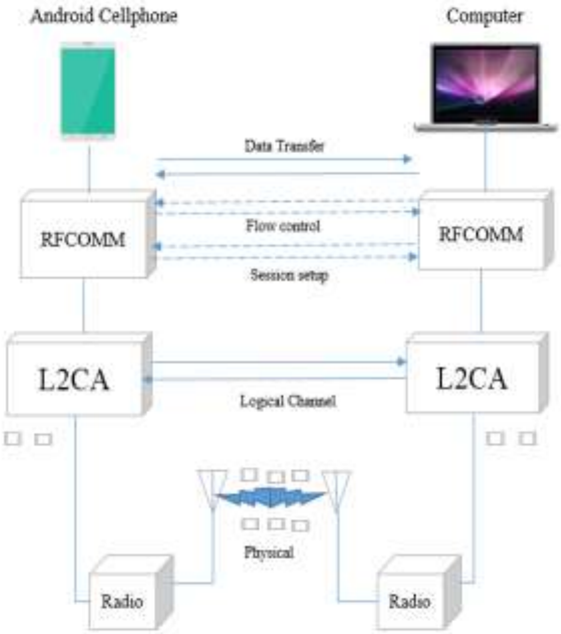


Fig 7. RFCOMM Communication

The Bluetooth protocol called RFCOMM is a simple set of transport protocols, made on top of the L2CAP protocol, giving emulated serial ports (up to sixty concurrent connections to a Bluetooth device at a time). The protocol is depends on the ETSI standard TS 07.10. RFCOMM is commonly called serial port emulation. The Bluetooth serial port profile is based on this protocol. Figure 7 showing the process to process communication using RFCOMM.

E. .NET CONSOLE APPLICATION TO READ BYTES OF DATA FROM SERIAL PORT [5]:

a. Serial Port class:

This class is used for controlling a serial port file resource. This class gives event-driven and synchronous I/O, and break states, and access to serial driver properties. Moreover, the functionality of this class can be hided in an internal Stream object, accessible through the BaseStream property, and send to classes that wrap or use streams. Figure 8 shows the data flow of .Net Console Application which reads the data from serial port.

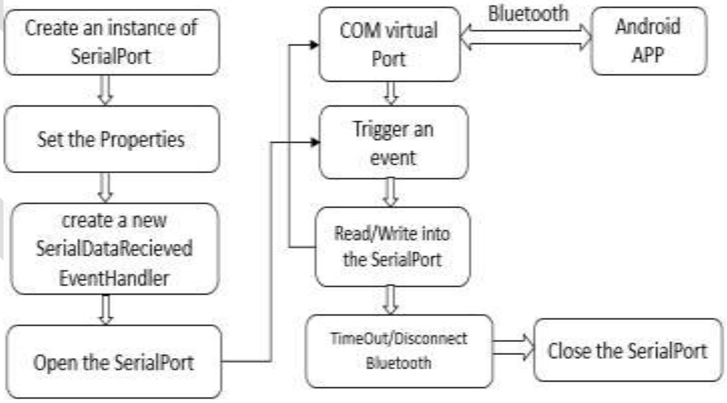

Create an instance of SerialPort
Set the Properties
create a new SerialDataRecieved EventHandler
Open the SerialPort
COM virtual Port
Bluetooth
Android APP
Trigger an event
Read/Write into the SerialPort
TimeOut/Disconnect Bluetooth
Close the SerialPort

Fig 8. Data Flow Diagram for .NET Application

- **Algorithm for .NET Application:**

STEP1: As shown in Figure 8, first create an object of the serialPort.
STEP2: Set the properties of serial port like Baud rate, Parity,Data Bits, StopBits etc.,
STEP3: Create a new SerialDataRecieved EventHandler and assign it for the created SerialPort.
STEP4: Open the Serial Port and connect the Android App to PC through Bluetooth.
STEP5: When the App sends the bytes of data to PC's Serial port, it triggers the SerialDataRecieved Handler event and Reads the data.

Below is the code for reading Bytes of data from a serial port:

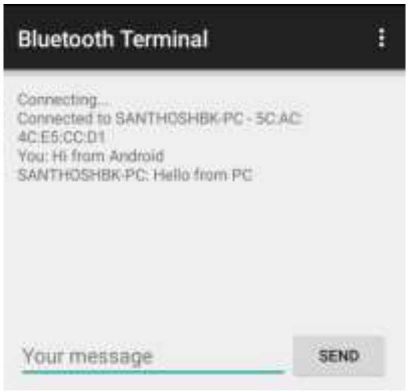
```
        // PortName = "COM3", Baud Rate = 9600, Parity = None,
        // Data Bits = 8, Stop Bits = One, Handshake = None

          SerialPort serialobj = new SerialPort("COM8", 9600, Parity.None, 8, StopBits.One);
          //To receive data, create an EventHandler for the "Serial Port"
          serialobj.DataReceived += new SerialDataReceivedEventHandler(DataReceivedHandler);
          serialobj.Handshake = Handshake.None;
          serialobj.Open();    //Opens serial port
          //logic for write data;
          Serialobj.Write("Hello from PC");//Read the data from console and send to Android
          serialobj.close();    //Closes the serial port
          private static void DataReceivedHandler(object sender, SerialDataReceivedEventArgs e)
          {
          SerialPort obj = (SerialPort)sender;
          string buf = obj.ReadExisting();  //Read Serial Port Data
          }
```

The DataReceived event is raised on a optional thread when data is gotten from the SerialPort object. Since this event is called on a secondary thread, and not the main thread, attempting to change some components in the main thread, such as UI components, could raise a threading exception. When the Android App sends data to PC's Serial port, it triggers an SerialDataEventHandler Event and reads the serial port data.

IV. BENEFITS AND USE CASES

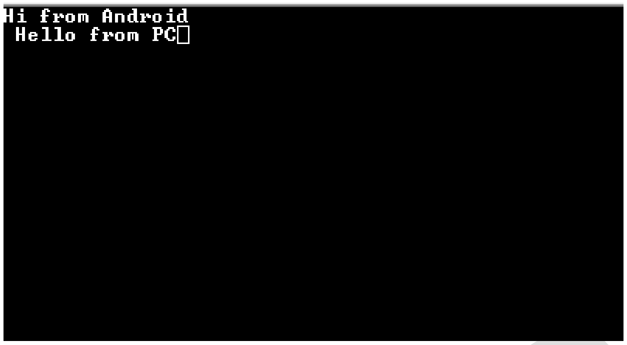
1. This mechanism does not need any extra hardware like Arduino board or HC-05 Bluetooth module, TTL serial converter which are used in the existing mechanisms.
2. As we are using Bluetooth as a transmission medium, it's very much secured and easy to connect and operate.
3. This mechanism is more reliable, costless.
4. Two way communications is possible over a single port.
5. It allows multiple connections simultaneously.

V. RESULTS



```
Connecting...
Connected to SANTHOSHBK-PC - 5C:AC:
4C:E5:CC:D1
You: Hi from Android
SANTHOSHBK-PC: Hello from PC
```

Fig 9. Screenshot of Android App sending data to PC and showing received data from PC



```
Hi from Android
 Hello from PC
```

Fig 10. Screenshot of PC sending data to Android and showing received data from Android

VI. CONCLUSION

The implementation of communication mechanism between 2 different platforms i.e., Android and .NET is achieved successfully. Further two ways communication is also proposed and completely tested with different Baud rates. For efficiency of the power and other resource usages, use the default Baud rate (9600). Ultimately this technology is useful for people across various industries. Requirements of people will be met efficiently.

VII. FUTURE WORK

1. In future, this mechanism can be extended to develop remote desktop connection from Android mobile using Bluetooth.
2. This mechanism can be enhanced by using advanced technologies for faster data transmission, like Wi-Fi, NFC or data network.
3. We can use this mechanism for developing Offline Chatting Applications using Bluetooth.

REFERENCES:

[1] Zhengrong Lei, Xiaoli Long,“ Research in Serial Port Communication Mechanism Based on Android Mobile” Proceedings of the 2012 2nd International Conference on Computer and Information Application (ICCIA 2012).

[2] http://www.forward.com.au/pfod/serialBluetoothConnections/Window7SerialBluetoothConnection/index.html.

[3] http://androidsrc.net/android-client-server-using-sockets-client-implementation/

[4] https://developer.bluetooth.org/TechnologyOverview/Pages/RFCOMM.aspx

[5] http://www.codeproject.com/Articles/340714/Android-How-to-communicate-with-NET-application.

[6] Ms Prerna Hingle, Ms Shubhangi Giripunje “ Real Time Multiple Cross Platform Communication through Bluetooth”, International Journal of Scientific & Engineering Research, Volume 6, Issue 2, February-2015

[7] Sadeque Reza Khan, and Farzana Sultana Dristy , “ ANDROID BASED SECURITY AND HOME AUTOMATION SYSTEM”, International Journal of Ambient Systems and Applications (IJASA) Vol.3, No.1, March 2015

[8] Gurpal Singh, Inderpal Singh, “ANDROID OS BASED WIRELESS DATA ACQUISITION SYSTEM VIA BLUETOOTH”, IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163 | pISSN: 2321-7308

[9] Li Ma1, Yu Zhou1, KuoLiao, Shen Liu, Jie Qiao, Zhenyu Han and Jin Wang, “Design and Implementation of Controlling PC Wirelessly by Android Mobile Based on C/S Mode International Journal of Control and Automation Vol.7, No.7 (2014), pp. 157-166 http://dx.doi.org/10.14257/ijca.2014.7.7.13\

[10] NISHAD WARHEKAR, APURVA KHANDARE, “RAPID COMMUNICATION WITH ANDROID BLUETOOTH CHAT” *International Journal of Research in Computer & Information Technology (IJRCIT) Vol. 1, Issue 2, March-2016*

[11] http://www.makeuseof.com/tag/how-to-set-up-bluetooth-in-windows-7-make-pc-discoverable-and-add-devices/

[12] Bluetooth Connectivity, http://developer.android.com/guide/topics/connectivity/bluetooth.html

[13] http://windows.microsoft.com/en-us/windows/choose-com-port-bluetooth