

CONNECTING A NOKIA® 7650 TO BAYANIHAN LINUX V2.0 USING COMMERCIAL BLUETOOTH ENABLERS

Emmanuel Balintec Janice Ballesteros Mabeth Borres
Anne Margrette Caccam Billy Pucyutan Francis Noel Reyes
Mobile Systems and Applications Group
Advanced Science and Technology Institute
UP Campus, Diliman, Quezon City

Abstract: Bluetooth™ is a wireless technology standard for the interconnection of electronic devices in the personal space. It was originally developed as a cable replacement technology and later on evolved into a Wireless Personal Area Network (WPAN). This paper discusses the implementation, setup, and testing processes involved in connecting a Nokia 7650 mobile phone to a computer running ASTI's Bayanihan Linux v2.0 (BLv2.0), which has built-in BlueZ (the official Linux Bluetooth protocol stack). The application used to interface these devices was developed over one of the layers of the Bluetooth™ protocol stack, the Radio Frequency Communication (RFCOMM) layer. RFCOMM is a simple transport protocol, which provides emulation of the RS232 serial ports over the Logical Link Controller and Adaptation Protocol (L2CAP). The setup was tested with four different commercial Bluetooth™ enablers.

Key Words: Bluetooth™, Bayanihan Linux, RFCOMM

1 INTRODUCTION

The advanced researches and developments in the field of wireless mobile technologies have caused the emergence of wireless devices in the local and international markets. In order to take advantage of this scenario, the Mobile Systems and Applications Group (MSAG) of the Advanced Science and Technology Institute (ASTI)¹ implemented a test plan and experiment to investigate the functionality and connectivity issues of these devices and one specific wireless mobile technology called Bluetooth™². The team would like to share the results of their tests with the Bluetooth application in the Nokia® 7650 mobile phone and four different commercial Bluetooth™ enablers run using the ASTI Bayanihan Linux (BL) v2.0 operating system. Generally, in this paper, four things will be highlighted: the ASTI Bayanihan Linux v2.0, the Bluetooth technology, the Nokia® 7650 mobile phone, and the four different Bluetooth™ enablers.

¹ ASTI is the research and development arm of the Department of Science and Technology of the Philippine government mandated to pursue R&D in the advanced fields of Microelectronics and Information Communications technologies.

² Bluetooth™ is a trademark owned by Telefonatibolaget L. M. Ericsson, Sweden.

2 LISTING AND CHARACTERIZING THE TOOLS USED

2.1 The ASTI Bayanihan Linux v2.0

The ASTI Bayanihan Linux is an easy-to-install Linux distribution primarily for desktop use. It is a customized distribution based on the Red Hat Linux 8.0 Psyche release. BL is a project of the Open Source Group of ASTI.

The name Bayanihan Linux was derived from the word *Bayanihan*, a Filipino tradition which signifies working together for the common good. The Open Source movement exhibits the spirit of Bayanihan, a virtual community working together to develop, produce, and maintain software that everyone can freely use.

Some features of BL version 2.0 include:

- Linux Kernel 2.4.18
- Easy to install with user-friendly interface
- KDE 3.0
- Office Suite applications
- Support for audio, video, and multimedia applications
- BlueZ Bluetooth Protocol Stack and Tools for Bluetooth applications

development (The included kernel has been recompiled and customized to particularly support Bluetooth technology, the BlueZ stack, and the applications development tools.)

2.2 Bluetooth™

Bluetooth™ wireless technology is a low cost, low power, short-range radio technology utilizing the 2.4 GHz ISM (Industrial, Scientific and Medical) band. It was originally developed as a cable replacement technology and evolved into a wireless personal area network (WPAN) technology. It provides a universal link to connect different devices without the hassle of connecting and disconnecting different kinds of wires and cables[1].

The Bluetooth specification defines a whole protocol stack necessary for a standardized interface of different Bluetooth™ devices. The Bluetooth™ protocol stack is defined as a structured software characterizing a complete system from the radio right up to the application level[2]. An example is *BlueZ*, which, as mentioned, is the official Bluetooth™ protocol stack of Linux. Figure 1 shows the Bluetooth™ Specification protocol stack.

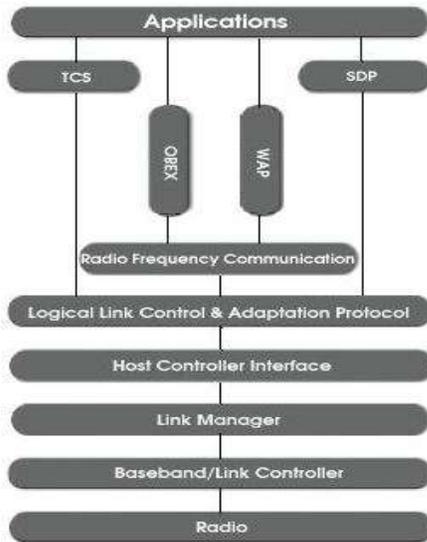


Figure 1 - The Bluetooth™ Protocol Stack

Applications can be developed over several layers of the protocol stack. In this paper, we shall focus only on one layer, which is the Radio Frequency Communications (RFCOMM).

2.3 The RFCOMM Layer

RFCOMM is a simple transport protocol that provides emulation of serial ports over the Logical Link Control and Adaptation Protocol (L2CAP). The RFCOMM protocol is based on the ETSI standard TS 07.10 and supports up to 60 simultaneous connections between two Bluetooth devices. The number of connections that can be used simultaneously in a Bluetooth device is implementation-specific[3].

A complete communication path is defined as a process that involves two applications running on different devices (the communication endpoints) with a communication segment between them[3]. In our case, the endpoints are the BLv2.0 box and the Nokia® 7650. Figure 2 shows the complete communication path.

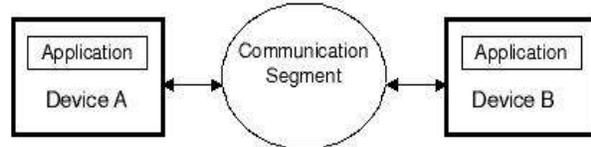


Figure 2 - RFCOMM Communication Segment

In our system, RFCOMM is implemented as a loadable kernel module. The RFCOMM special files, listed as `rfcomm0`, `rfcomm1`, ..., and so on, are located in the `/dev` directory. These special files can generate and receive data and usually correspond to a physical piece of hardware[4]. In our case, the special piece of hardware is represented by the different Bluetooth enablers, which will be discussed in the following section. Figure 3 shows the RFCOMM reference model, a representation of how RFCOMM fits into a typical system.

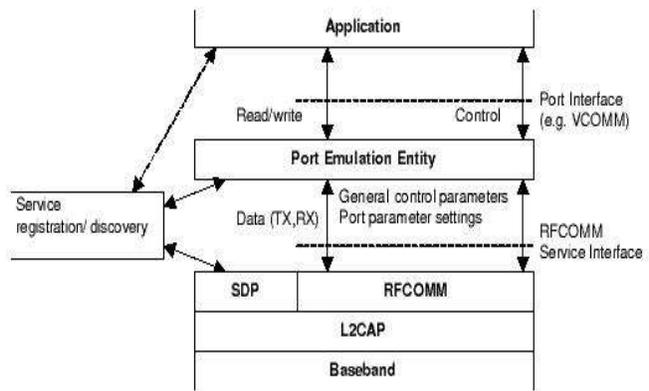


Figure 3 - RFCOMM Reference Model

The elements of the RFCOMM reference model includes the application, port emulation entity, RFCOMM, Service Registration and Discovery, L2CAP, and Baseband. The applications utilize the serial port communication interface. The port emulation entity, on the other hand, maps a system-specific communication interface or the Applications Programming Interface (API) to the RFCOMM services. The port emulation entity plus the RFCOMM make up a port driver. The RFCOMM provides a transparent data stream and control channel over an L2CAP channel. It multiplexes several emulated serial ports. In the service registration and discovery, the server applications are registered on the local devices for the client applications to discover, and thus be able to reach and utilize them on other devices. L2CAP is for protocol multiplexing and the baseband is the baseband protocol defined by Bluetooth specification[3].

At any time, there must be at most one RFCOMM session between any pair of devices. Setting up an RFCOMM connection in BLv2.0 is discussed in the succeeding sections.

2.4 The Nokia® 7650

The Nokia® 7650 mobile phone is Nokia's first mobile phone that comes with an integrated digital camera and one of Nokia's series 60 phones. It has a lot of features including it's 3.6Mb dynamic memory, full color display, Multimedia Messaging Service (MMS), General Packet Radio Service (GPRS), and others. It uses the Symbian Operating System with support for MIDPJava™. It also supports Infrared and Bluetooth™ Technology for connectivity with other devices[5].

Generally, applications do not have access to the hardware layer of the protocol stack. However, the Bluetooth APIs of the Symbian Operating System (OS) do give applications access to the RFCOMM, L2CAP, SDP and, to a limited extent, the HCI layers of the Bluetooth™ protocol stack.

Profiles are used to ensure interoperability between bluetooth devices. A profile is a description of how to use a specification to implement a given end-user function[6]. The profiles supported by the Nokia 7650® and all of

Nokia's series 60 phones include the Generic Access Profile, Serial Port Profile (DevA and DevB), Generic Object Exchange Profile, and the Object Push Profile (as a client and server). Third-party applications can take advantage of the these profiles through C++ APIs. In addition, certain profiles can be used from an external Bluetooth device even without an application on the phone. These include the File Transfer Profile (as a server), Dial-Up Networking Profile (as a gateway), and Fax Profile (as a gateway)[7].

Figure 4 shows an image of the Nokia® 7650 mobile phone.



Figure 4 - Nokia® 7650

2.5 Bluetooth Enablers

In performing the test/experiment, the team used four different samples of commercial Bluetooth enablers, namely, the 3COM Bluetooth USB Adapter, 3COM Bluetooth PC Card, the Mitsumi and MSI Bluetooth USB Dongles. These apparently made the ordinary PC's Bluetooth-enabled.

The 3COM Bluetooth USB Adapter 3CREB96-EU and 3COM Bluetooth PC Card 3CRWB6096 v2 both use the 3Com stack. They also have support for Serial Port/Dial-up Networking/ profiles. The Mitsumi Bluetooth USB adapter is compliant with the Bluetooth 1.1 Standard. It runs with the WidComm Bluetooth protocol stack. It has Generic Access/Service Discovery Application/Serial Port/Dial-up Networking/ LAN Access/FAX/Generic Object Exchange/Object Push. The MSI Bluetooth USB Dongle provides wireless technology that revolutionizes personal connectivity, with the use of its Widcomm stack. It is the solution for the seamless integration of Bluetooth technology

into personal computers, enabling short-range wireless connections between desktop/ laptop computers, Bluetooth-enabled peripherals (printers, faxes, etc.), portable handheld devices, and connectivity to the Internet. It is Bluetooth Specification version 1.1 compliant. Its network capabilities support piconet point-to-point and point-to-multipoint connections. Among its profiles are: Generic Access, Service Discovery, Serial Port, Dial-Up Networking, Fax Profile, LAN Access Profile, Object Exchange, File Transfer, Object Push, Synchronization, Personal Area Network, Hard Cable, and Replacement Profile.

Figures 5 a-d show images of these four different Bluetooth enablers.



Figure 5a - 3COM USB Adapter



Figure 5b - 3COM PC Card



Figure 5c - Mitsumi USB Dongle



Figure 5d - MSI USB Dongle

3 GETTING YOUR DEVICES UP AND RUNNING

BlueZ as a whole consists of a set of kernel modules which implements the Bluetooth protocol stack and some programs to control the stack and to use the stack to communicate[8]. BlueZ has been setup in BLv2.0 so all we have to do is setup the application profile OBEX (Object Exchange) and the server. OBEX is a session protocol and can best be described as a binary HTTP protocol[9]. An implementation of OBEX in BlueZ is OpenObex. In addition to the core software libraries, there is also a collection of simple applications built on top. Using the files included in the OpenObex software package, we will be integrating the server file which provides a handy way to receive files from the Nokia® 7650. This server is the OpenObex file server which accepts files transferred using OBEX from other devices[10]. In the setup, the OBEX application runs on top of RFCOMM.

Setting up BLv2.0 to communicate with the Nokia® 7650 is a straight forward matter. Initially, the procedure calls for enabling of Bluetooth and then running the appropriate software in the PC to communicate with the phone.

3.1 Setting up your BL Box

To enable the BL box to connect to the Nokia® 7650, it is necessary to first obtain the OpenObex packages, openobex and openobex-apps. The next thing to do is to un-tar the files, compile and then install with the following commands:

```
[jan@Asgard jan]$./configure &&
make install prefix=/usr
```

Next, get the source of the obex server we are going to use from <http://www.frasunek.com/sources/unix/obexserver.c>. Copy this file to the openobex-apps source directory and compile using the following command:

```
[jan@Asgard jan]$gcc -o obexserver
obexserver.c libmisc.a -lopenobex
```

After setting up OpenObex, we can now proceed with interfacing the two devices. Simply plug the Bluetooth enabler to your BLv2.0 box. Initialize the device by using the command

```
[jan@Asgard jan]$ hciconfig hci0 up
```

At this point, if you run `hciconfig` you should see an interface showing that your device is up and running:

```
[jan@Asgard jan]$ hciconfig -a hci0:
Type: USB
BD Address: 00:A0:96:1F:D4:FE ACL
MTU: 128:8 SCO MTU: 64:8
UP RUNNING PSCAN ISCAN
RX bytes:20202 acl:182 sco:0
events:73 errors:0
TX bytes:1009 acl:39 sco:0
commands:28 errors:0
Features: 0xff 0xff 0x05 0x00
Packet type: DM1 DM3 DM5 DH1 DH3
DH5 HV1 HV2 HV3
Link policy: HOLD SNIFF PARK
Link mode: SLAVE ACCEPT
Name: 'Sleipnir'
Class: 0x000100
Service Classes: Unspecified
Device Class: Computer,
Uncategorized
HCI Ver:1.1(0x1) HCI Rev: 0x72 LMP
Ver:1.1(0x1) LMP Subver: 0x72
Manufacturer:Cambridge Silicon
Radio(10)
```

Run the HCI³ and the SDP⁴ daemons. The HCI daemon runs as a process while the SDP daemon listens to the L2CAP and Unix sockets. It responds to incoming SDP queries, maintains the SDP database, allows record registration via unix socket, etc.

```
[jan@Asgard jan]$ hcid
[jan@Asgard jan]$ sdpd
```

After successfully running the daemons, register the SDP service for OpenObex Object Push(OPUSH).

```
[jan@Asgard jan]$ sdpdtool add
channel=10 OPUSH
```

The RFCOMM channel is dependent of the manufacturer of the device. Nokia[®] uses RFCOMM channel 10. If the service is properly registered, the message “OBEX Object Push service registered” will be displayed.

Now, we are ready to test our obexserver. Run the OBEX server.

```
[jan@Asgard jan]$ ./obexserver
```

The message “Waiting for connection...” will be displayed if the system is properly setup. You can now upload files from the Nokia[®] 7650 to the BLv2.0 box.

3.2 Sending Files From the Nokia[®] 7650

Now that the server is working, you can now send files from your mobile phone. The Nokia[®] 7650 communicates using the Object Exchange (OBEX) protocol.

Turn on Bluetooth connectivity in your device. From the main menu of your mobile phone go to *Images*. Select an image, click *Options* and select *Send -> via Bluetooth*. The list of devices found will be displayed. The default name of the Bluetooth enabler in your BLv2.0 box is Bluez(0). Select this and wait until the image is sent. While sending, a notification message will be displayed in the Nokia[®] 7650. If there is no error in the transfer process, the file is then saved in the `directory / tmp`. You may send as many files as you want.

4 RESULTS

The team was able to implement and test the connection setup of a Nokia[®] 7650 mobile phone to a computer running ASTI's Bayanihan Linux version 2.0 carrying a commercial Bluetooth enabler.

3 The Host Controller Interface (HCI) provides a command interface to the Baseband Link Controller and Link Manager, and access to hardware status and control registers[11].

4 The Service Discovery Protocol (SDP) provides a means for applications to discover which services are provided by or available through a Bluetooth device. It also allows applications to determine the characteristics of those available services[11].

From the previous discussions, it can be derived that the commands have been executed via the console or terminal. To make the obexserver application user friendly, the team developed a Graphical User Interface (GUI)-based application, which is called the *PCStorage*. Figure 6 shows a screenshot of *PCStorage*. It was developed using Glade 0.6.4. Glade is an application for creating graphical user interfaces that use the *Gtk+* and *GNOME* libraries. Glade allows you to rapidly develop these interfaces, and it can create source codes in a variety of languages that will construct the interfaces for you. Glade can also be used in conjunction with *libglade* to dynamically create user interfaces from the XML description file that Glade creates.



Figure 6 - PCStorage

PCStorage simply runs obexserver and waits for connections from the Nokia® 7650. The application works by first selecting the port where the Bluetooth enabler is connected. Clicking the OK button will run obexserver. All files sent from the Nokia® 7650 will be saved in the /tmp directory and will be previewed using an image viewer application.

In the implementation, the use of the different commercial Bluetooth enablers have been significant for the test of interoperability. Bluetooth technology is a standard that claims interoperability among devices from different manufacturers.

The observed speed of the file transfer is distance dependent. For close transfers like one foot away, the transfer of a 30KB file takes approximately 3 seconds.

The speed is almost the same when transferring files using each of the four Bluetooth enablers.

5 CONCLUSION

Connecting a Nokia® 7650 to the ASTI Bayanihan Linux v2.0 using commercial Bluetooth enablers is possible. Basically, the tools needed are the following: the Bayanihan Linux v2.0, which has built-in BlueZ protocol stack and kernel supporting Bluetooth, a Nokia® 7650 mobile phone with built-in Bluetooth tools and applications, and several Bluetooth enablers.

In each of the devices (the Bluetooth-enabled PC and the Nokia® 7650), the Bluetooth modules have already been organized and are ready to use. From the BLv2.0 box, the procedures are simply to setup the OpenObex and subsequently issue the commands from the BlueZ console or run a GUI-based application if available. From the Nokia® 7650, it is only necessary to turn on the Bluetooth module and as an end-user application start transmitting or receiving the data to or from the computer.

6 RECOMMENDATIONS

6.1 Hardware

Currently, the personal computer set-up is fully tested with a Nokia® 7650 mobile phone. It would be better if it can support other bluetooth-equipped mobile phones from other manufacturers. The difficulty with diverse mobile phone model clients will be the discovery in which channel the software application works. This will be a subject of further research in this field of Bluetooth interoperability.

Another issue is the mobility of the server. Sure, we can set-up PCs in malls and other areas where the application can be deployed but it would be easier if we wouldn't have to bring a PC set-up, instead we have a dedicated smaller hardware as a server. The Linux operating system, as well as the applications can be embedded into a single board computer and we can use the raw Bluetooth modules instead of Bluetooth USB dongles. This will greatly reduce the size of the server.

6.2 Software

The *PCStorage* software application can be enhanced by making it a native KDE application which is the desktop environment of Bayanihan Linux version 2.0. It will also be better if we

invoke it from the KDE menu or it can be an applet that is running once we start the PC and it will wait for incoming data. This will create the impression that the technology is easy to use.

References

- [1] Bray, Jennifer and Charles F. Sturman. BLUETOOTH: *Connect Without Cables*
- [2] Ballesteros, Borres, Botardo, Caccam, Galang, Pucyutan. "*Real-Time Systems Development Using Object-Oriented Real-Time Techniques (OORT)*": A Paper Presented at The First Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) International Conference: March 29-31, 2003, Manila, Philippines.
- [3] *Specification of the Bluetooth System*. Volume 1, Version 1.1, February 22, 2001.
- [4] mkndod man pages, Blv2.0
- [5] <http://www.nokia-asia.com/nokia/0,8764,27445,00.html>
- [6] http://www.informit.com/isapi/product_id_2E2FB42B4C64/content/index.asp~7C5F4B65-5C59-4269-A385-

- [7] http://www.forum.nokia.com/main/1,6566,1_43_50_10,00.html
- [8] <http://www.unrooted.net/hacking/bluez-rfcomm-obex.html>
- [9] <http://openobex.sourceforge.net/index.html>
- [10] <http://www.iterationx.org/essays/2002/10/21/>
- [11] <http://www.palowireless.com/infotooth/tutorial.asp>

The images were taken from the following websites|:

[www.str.spb.ru/images/ BluetoothMSI.jpg](http://www.str.spb.ru/images/BluetoothMSI.jpg)

www.blueunplugged.com/Shop/detail.asp?productgroupID=60

www.blueunplugged.com/shop/detail.asp?productgroupID=92

www.blueunplugged.com/shop/detail.asp?productGroupID=63