

Operate Computer from Android Device via Bluetooth

W.A. Tiroshan Madushanka

Department of Computer Science & Engineering University of Moratuwa, Katubedda

Sri Lanka

tiroshanm@gmail.com

out.

Abstract – This paper presents the main steps followed during the development of AndroMouse, a brief discussion on each step, justification for the choices made and most importantly, special/different methodologies used in the application.

AndroMouse application can be classified as a combination of three separate subsystems as follows.

1. AndroMouse Android Application
2. Communication Server
3. Driver

AndroMouse Android Application can be considered as the main user interface that contains functions for controlling a computer device remotely.

AndroMouse Communication Server is a Bluetooth based communication handling server which functions using universally unique identifiers (UUIDs). Message encryption and message handling components are implemented along with the Android application and message decryption and session handling components are implemented along with the Windows based application. Driver module controls hardware components and it functions according to the decrypted message.

Index Terms – Android Platform, Bluetooth stack

I. INTRODUCTION

AndroMouse is a standalone Android based application which provides the functionality of controlling a computer device via Bluetooth. The system consists of two components. The first component is an Android application and other component is a Windows application.

AndroMouse 1.0 provides following functionalities of controlling a computer device (desktop computer, laptop etc) connected remotely over Bluetooth with a user friendly interface.

1. Mouse/ Touchpad functions
2. Keyboard functions
3. Selection of power mode
4. Control of the media
5. Game play functions
6. Presentation functions

The Android application is used as an interface for the system and the application can be operated in different modes such as mouse mode, keyboard mode, media player controller mode, PowerPoint presenter mode and game mode. Bluetooth server and driver software is embedded in the Windows based application. The following sections will provide an overview of the main activities carried

II. SYSTEM ARCHITECTURE

A. Layered Architecture

The AndroMouse system was implemented with a layered architecture pattern using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system. The AndroMouse system can be interpreted using four layers.

1) Presentation Layer-

The Android application is implemented using the Android 2.2 API. The AndroMouse user interface first asks to enable Bluetooth service if the device Bluetooth service is disabled and then provides a list of paired history. Then the application switches to the main user interface which contains the selection of controlling modes.

The Windows based server and driver software is implemented using the JAVA platform. Java swing library^[2] is used for the implementation of user interface. User interface is implemented as a tray icon and the tray icon is changed according to the server condition.

2).Communication Layer-

Bluetooth communication between the mobile device and the host server is represented in the communication layer. The system is implemented as three subsystems namely Communication Handler subsystem, Paired History Manager subsystem and Data Encryption subsystem.

The Bluetooth adapter facility of the Android platform and the standard Bluetooth stack is used to implement the communication layer. Communication is established using a pre-initialized universally unique identifier (UUID). A byte buffer is used in the server side in order to buffer the input byte data. A session is established between the

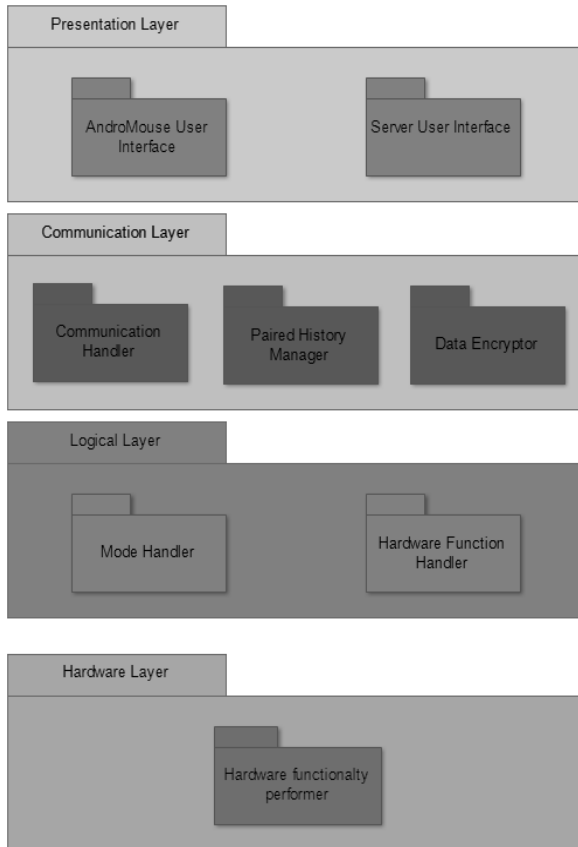


Fig 1 - Layered architecture of the system

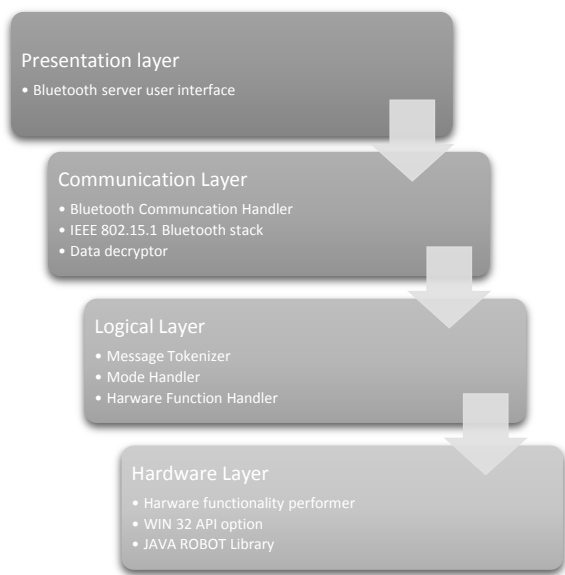


Fig 2 - Layered architecture of the Windows Application

Fig 3 - Layered architecture of the Android Application

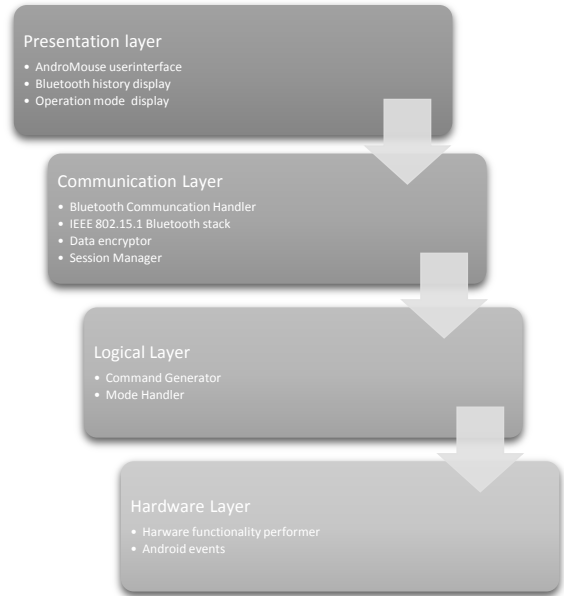


Fig 4 – Server status indication tray icon
 I. Connected to a device
 II. Server idle
 III. Server disconnected

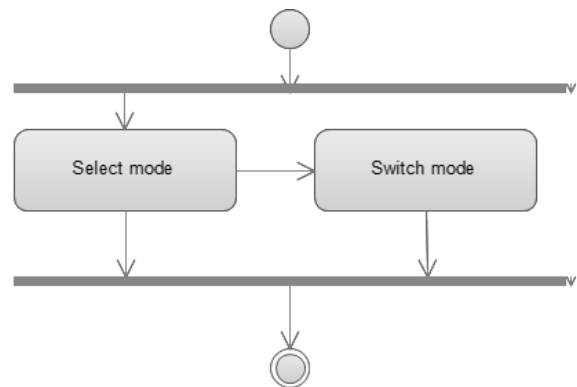


Fig 5 - Switch mode process view in presentation layer

Android application and the server when the application is launched and the session is terminated

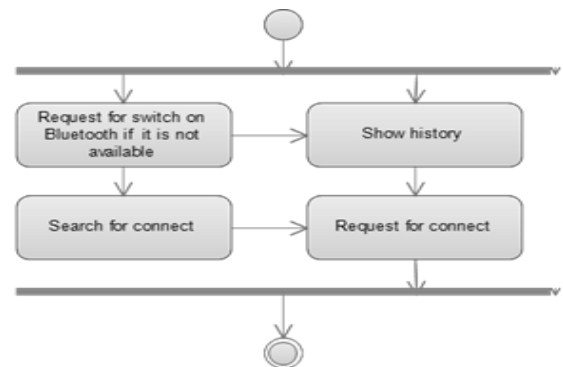


Fig 6- Connection establishment process view in communication layer

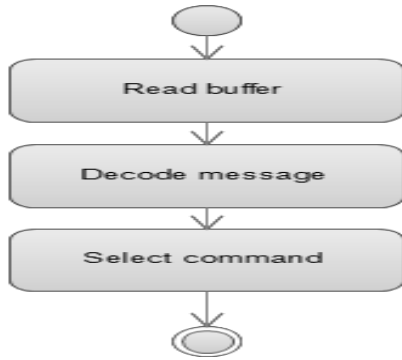


Fig 7 - Data encryption in server process view

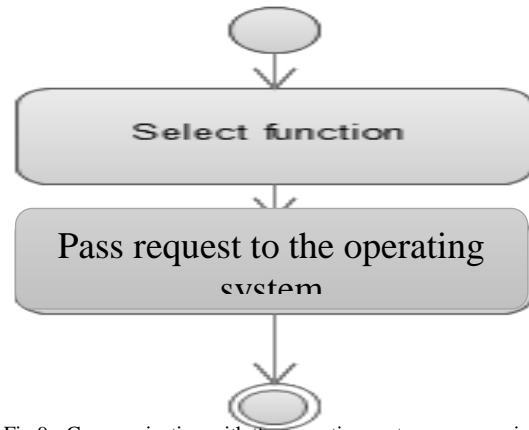


Fig 9 - Communication with the operating system process view in hardware layer

when the Android application is closed.

3) Logical Layer-

The functional modes of the Android application are implemented in the logical layer and the AndroMouse system can function under five modes. The operating mode can be selected from the given list of modes. AndroMouse 1.0 application provides following modes.

1. Mouse mode
2. Key board mode
3. Presentation mode
4. Media mode
5. Power Option mode

Functionalities of the modes of operation are implemented using the Android 2.2. API. Logical layer of the Android application processes and encrypts the control commands and transfer the data to the communication handler.

A.4 Hardware Layer

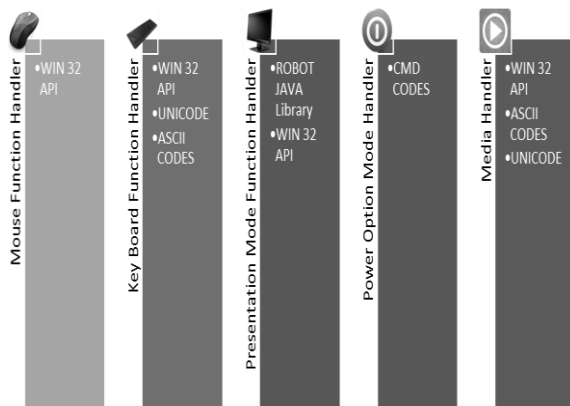


Fig 8 – Hardware Layer subsystems

III. CONNECTING TO THE SERVER

[1] Run the server

Initially, the user has to run the Windows based Bluetooth server in order to run the system. AndroMouse 1.0 is only capable of connecting over Bluetooth.

The server keeps listening to its input stream until it receives an encrypted “CONNECT” request. (All input data is decrypted). If the UUID of the server is matched with the request UUID, the session is established. Otherwise it rejects the request.

[2] Connect to the server

User has to launch the Android application in order to connect to the server. Application provides the Bluetooth paired history and search functionality. When a device is selected, the application captures the media access controller (MAC) address of the device and sends the encrypted “CONNECT” request to the server.

[3] Select modes

After the session is established between the server and the application, the user interface is switched to the mode interface.

III. COMMUNICATION WITH HARDWARE

The hardest part of the system was the implementation of the driver software. There are number of different approaches that can be used to implement the driver software. Since the server is implemented using the JAVA platform, the driver software was implemented using the JAVA platform as a subsystem of a Windows based

application by integrating the hardware controlling module. In order to achieve the required functionalities the system is implemented using three basic approaches.

- Used ROBOT JAVA library
- Extended the W32APIOptions
- Implemented functions using ASCII codes and command line commands

A. *ROBOT JAVA library* [2]

ROBOT JAVA library is used to generate native system input events for the purposes of test automation, self-running demos, and other applications where control of the mouse and keyboard is needed.

B. *WIN 32 API Option* [1]

W32API Option provides flexibility in communication with the hardware components of a computer. W32APIOption is implemented under the Java Native Access project which has started in year 2011. This library provides flexibility in work in both ASCII values and the UNICODE values.

REFERENCES

- [1] Emmanuel Bourg. Win32API [Online]. Available
FTP:<https://github.com/twall/jna/tree/master/src/com/sun/jna/win32>
- [2] Java Native Access [Online]. Available:
<https://jna.java.net/javadoc/com/sun/jna/win32/W32APIOptions.html>