

Android-based Java Programming for Mobile Phone LED Control

Yi-Jen Mon

Department of Computer Science and Information Engineering, Taoyuan Innovation Institute of Technology
Chung-Li, Taoyuan, 320, Taiwan, R. O. C.

Abstract

The smart phone is quite common with closed market in the past. The mobile phone manufacturers started researching for mobile phone system since Google Ltd. started to develop a completely free open mobile phone platform called Android operation system (OS). The world's major manufacturers and scholars began to study the Android OS to the smart phone on the mobile phone market. This has unlimited possibilities and opportunities. Android applications are developed by using the Java programming language. By using Android OS, such as desktop, digital phone, phonebook, web browser and other useful applications have been provided to engineers to develop many other applications. In this paper, the Android development kit called DMA 6410 is used to do empirical test of LED control. The software is developed by Eclipse which is based on Linux and Android. Then the peripheral of LEDs are controlled to be turned on or off by the buttons appeared in the touch pad screen. The experimental results reveal that the control performances are possessed.

Keywords: *Android, LED, Eclipse, Java.*

1. Introduction

The Google Ltd. launched the open Android operating system platform in November 2007 and announced the establishment of a union organization called the Open Handset Alliance (OHA). Google Inc. also promotes the Android mobile phone operating system and related applications software. In addition to the Google Inc. U. S., the alliance organization has includes many manufacturers of mobile phone developments, telecommunications companies and application software development companies. Android operating system in addition to be sent to the phone, meanwhile, it can be developed as embedded systems. Android is the world's most popular mobile platform. With Android, all the Google apps can be used plus there are more than 600,000 apps and games available on Google Play to keep them entertained, alongside millions of songs and books, and thousands of movies. Android devices are already smart, and will only get smarter, with new features be won't found on any other platform, letting us focus on what's important and putting methods in control of mobile experience [1].

Android has become the most popular smart phone platform, recently. It generally lags way behind as far as tab-

lets go the iPad retains something of a stranglehold on that market but significant numbers of Android tablets are being sold nonetheless. From May 2012 give Android 59% of the worldwide market for smart phones against 23% for iOS. And a number of online advertising companies have now positioned Android as the popular smart phone platform worldwide [2]. Eclipse which is an open source multi-language software development environment, offers a user interface framework into which any company can add functionality though plug-ins, thus creating an overall product which serves a broad group of software developers [3]. Many research results have been developed recently. For example, an Android-based 3G phone is developed to establish enterprise applications of ground monitoring of aircraft loading and production activities at the airfield, and use the event mechanism to coordinate the production work of System Operation Control Department (SOC), cargo and aircraft maintenance factory [4]. A general process for data collection of Android devices is developed by exploring special device boot modes and Android's partitioning schema [5]. To support collaboration of teams of mobile users by enabling anytime and anywhere access to shared contact data [6].

In this paper, the Android development kit embedded of ARM-11 based Samsung 6410 CPU [7] called DMA 6410 manufactured by *DMATEK Ltd, Taiwan* [8] is used to do empirical test of LED control. The experimental results reveal that the control performances are possessed.

2. The Software Development of Android and Eclipse

Android is a powerful Operating System supporting a large number of applications in Smart Phones. These applications make life more comfortable and advanced for the users. Hardware's that support Android is mainly based on ARM architecture platform [9]. Each Android project includes three main files as: (1) Main.XML that manages interface components, (2) Android Manifest.XML that contains the set of references used to execute Android application, (3) Active.java that is used to start Android [10]. Android is Linux based operating system. Android OS consist of various components which include a Linux ker-

nel, Android runtime, Set of various libraries, Application framework and Applications. Android uses Linux kernel version 2.6 for providing its essential services, such as process management, memory management, security etc. Android runtime includes Dalvik virtual machine. Dalvik Virtual Machine is conceptually quite similar to Java Virtual machine but instead it uses a different file format. Android applications are written in java language and compiles into dex files. These dex files are packaged into a single file called Android Package (.apk). The dex files are optimized for consuming minimal memory [11]. Android Application is developed by software development kits (SDK) issued by android web [1]. It is combined of the following tools:

- JDK(Java Development Kit) –This is an application tool used to develop such as some maps applications.
- Eclipse- This is an integrated development environment (IDE) which includes many application software of Java such as Editor, Debugger, etc. The Eclipse plug-in invokes the Android resource compiler, to adapt, and to preprocess the XML into a compressed binary format [10].
- ADT(Android Development Tools)– ADT is provided by Google to be used in the plug-in environment for Eclipse to design the applications of Android project.
- Android SDK –This Android SDK can transfer the Java class to the files of .dex file types and pass this .dex files to Dalvik virtual machine (VM) to execute the functions. Besides, it includes some other useful tools such as Android emulator and debugger of LogCat.
- Linux kernel –This is necessary for Android applications. All the developed Android applications must based on this operation system (OS) of Linux. In this paper, the Ubuntu Linux OS is used.

The installation steps are described as follows. At first, to install Java JDK-> install Eclipse-> install Android SDK-> install ADT. After this procedures of installation are completed, the file folder of 'android-sdk-windows' will be appeared in folder of disk 'C'. This path of file folder must be set appropriate to be pointed by other applications. The setting steps are from clicking on the icon of 'My Computer' at first, then click right mouse button, select 'Contents' and click the right of mouse button to enter the screen of 'system'-> setting 'system Properties'-> 'Advanced'-> 'environment settings'-> 'environment variables'-> 'system variables'-> press 'edit' to find a variable field of 'Path'-> paste file folder of "JDK" and "SDK" into this field of variables. Finally, the installation is completed and the ADT can be run smoothly. It is an important step which diagram of setting result is shown in Fig. 1.

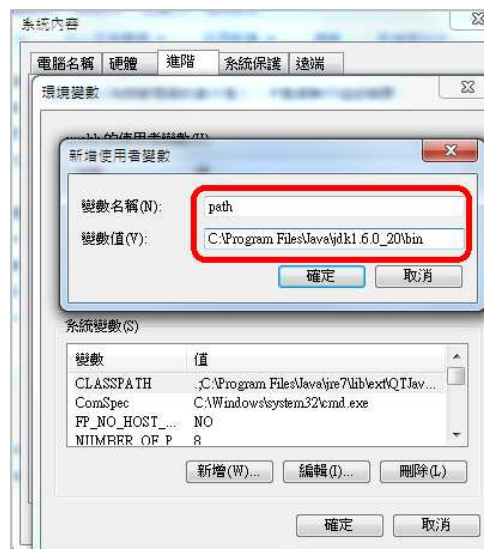


Fig. 1 The diagram of 'path' setting result

The Java software development kit is necessary for compiling the library files of Android applications. At first, the tool called 'Putty' is used to communicate the Java machine. In the environment of Putty, we can compile the head file and C language file of Android by using the command of 'javah -classpath'. The command is executed as the Fig. 2. This command will generate the necessary file of type of '.h'. The compiled head file is shown in Fig. 3.



Fig. 2 The command of 'javah'

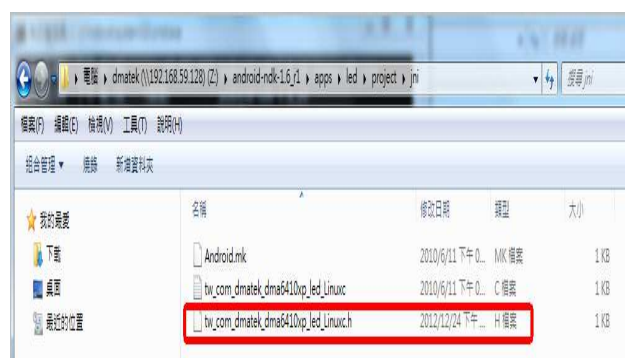


Fig. 3 The diagram of compiled head file of Android

In the development kit, the schematic diagram of LED is shown in Fig. 4. The numbers of LEDs are D608 and D609 which are set appropriate in the head file of this project. The Java programming will be suitably matched with these identified devices.

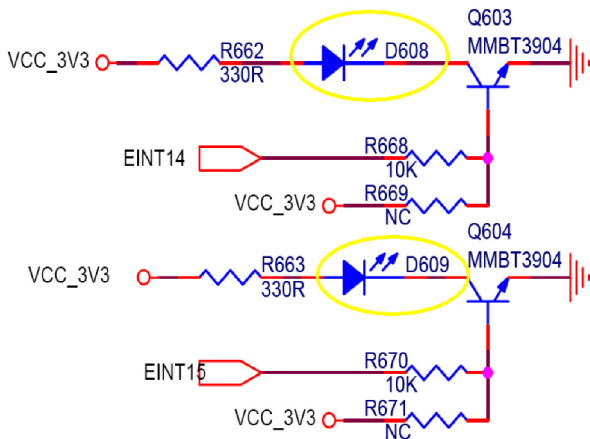


Fig. 4 The schematic diagram of LEDs of DMA 6410 kit

The Eclipse environment setting diagram is shown in Fig. 5. The Android class file which is set as 'Led_Control' in this paper is generated by 'Linuxc' and Android 1.6 which is shown in Fig. 6. All the compiled files must put into Eclipse to generate the Android application program which file type is '.apk'. The finally simulated layout of LED control in this paper is shown in Fig. 7 which can control two LEDs to be turned on or off. The main program of Java is developed as shown in Fig. 8. The file of 'Led_Control.apk' can be generated after executing the 'clean' command of Eclipse which file is shown in Fig. 9. This file can be burned into flash of Android development kit named DMA 6410 manufactured by *DMATEK, Ltd, Taiwan* to do real experimental test [7]. The main parts of Java program of 'Led_Control' is designed as follows.

```

/* Define the controlled number of LEDs */
public int state1 = 0, state2 = 0;
/* The number of 1 implies to be light */
public int led_on = 1;
/* The number of 2 implies to be dark */
    public int led_off = 2;
    public int fd = 0;
    private Button.OnClickListener button1_listener=
new Button.OnClickListener()
    {
        public void onClick(View v)
        {
            Linuxc.send(2, led_on);
        }
    };

    private Button.OnClickListener button4_listener=
new Button.OnClickListener()
    {
        public void onClick(View v)
    {

```

```

Linuxc.send(2, led_off);
    }
};
    private Button.OnClickListener button5_listener=
new Button.OnClickListener()
    {
        public void onClick(View v)
        {
            Linuxc.send(1, led_off);
        }
    };
    private Button.OnClickListener button2_listener=
new Button.OnClickListener()
    {
        public void onClick(View v)
        {
            Linuxc.send(1, led_on);
        }
    };
    private Button.OnClickListener button3_listener=
new Button.OnClickListener()
    {
        public void onClick(View v)
        {
            /* Close all documents of equipments*/
            Linuxc.closeled();
            /* Quit the running process*/
            finish();
        }
    };
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}

/*Use the structure of &findViewById * to employ &
main.xml * and &string.xml */
    mButton1 =(Button) findViewById(R.id.myButton1);
    mButton2 =(Button) findViewById(R.id.myButton2);
    mButton3 =(Button) findViewById(R.id.myButton3);
    mButton4      =(Button)          findViewById(R.id.myButton1_1);
    mButton5      =(Button)          findViewById(R.id.myButton2_1);
    /* Open the document of led to get the feedback value
of &fd */
    fd = Linuxc.openled();
    if (fd < 0){
        setTitle("open device false!");
        /* If the open process is fail then quit*/
        finish();
    }
    else {
        setTitle("open device success!");
    }
}

```

```

/*Use &setOnClickListener * to listen event*/
mButton1.setOnClickListener(button1_listener);
mButton2.setOnClickListener(button2_listener);
mButton3.setOnClickListener(button3_listener);
mButton4.setOnClickListener(button4_listener);
mButton5.setOnClickListener(button5_listener);
    }
}
    
```

The main part of source program of Linuxc.java is designed as follows.

```

public class Linuxc {
    static {
        try {
            Log.i("JNI", "Trying to load libledtest.so");
            /* To make library of & libled.so * */
            System.loadLibrary("ledtest");
        }
        catch (UnsatisfiedLinkError ule) {
            Log.e("JNI", "WARNING: Could not load libled-
            test.so");
        }
        public static native int opened();
        public static native int closed();
        public static native int send(int led_num, int
        on_off);
    }
}
    
```

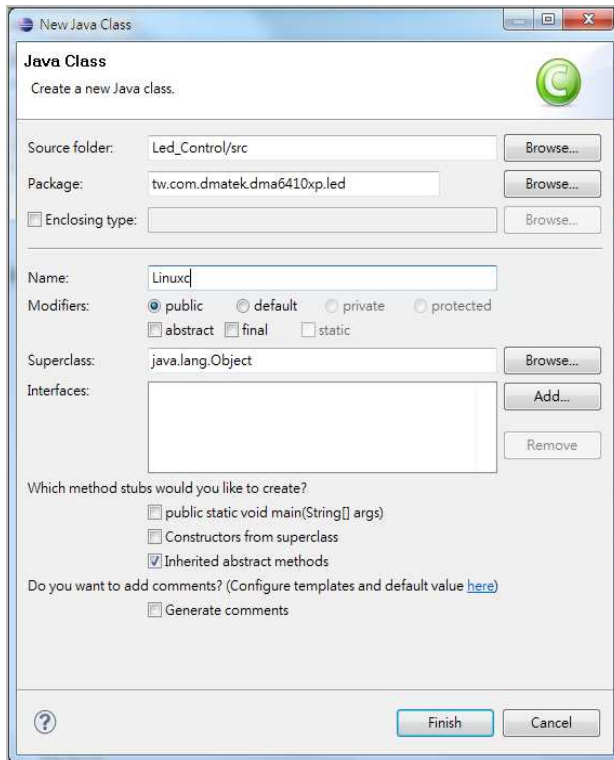


Fig. 6 The diagram of Android class file setting

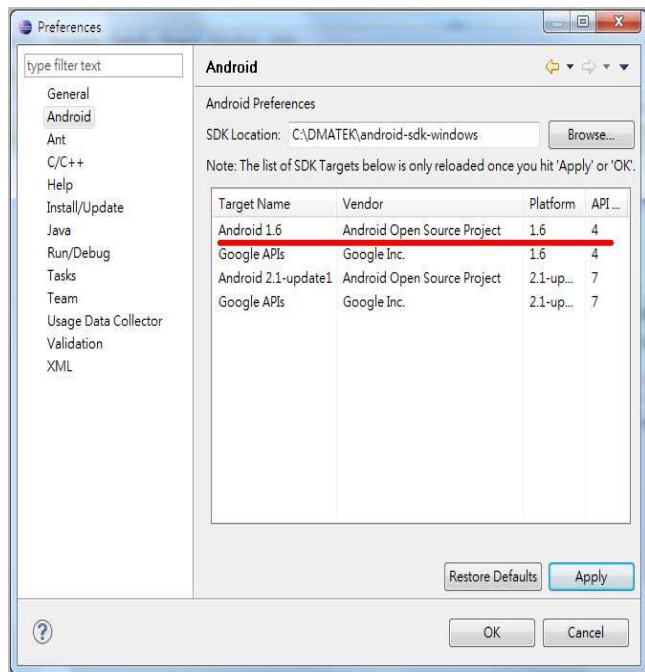


Fig. 5 The Eclipse environment setting diagram

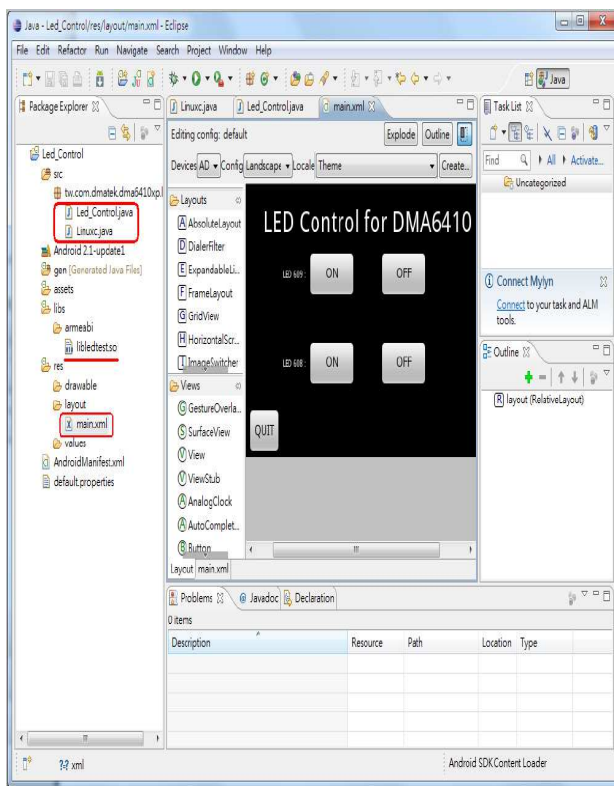


Fig. 7 The diagram of simulated layout of LED control

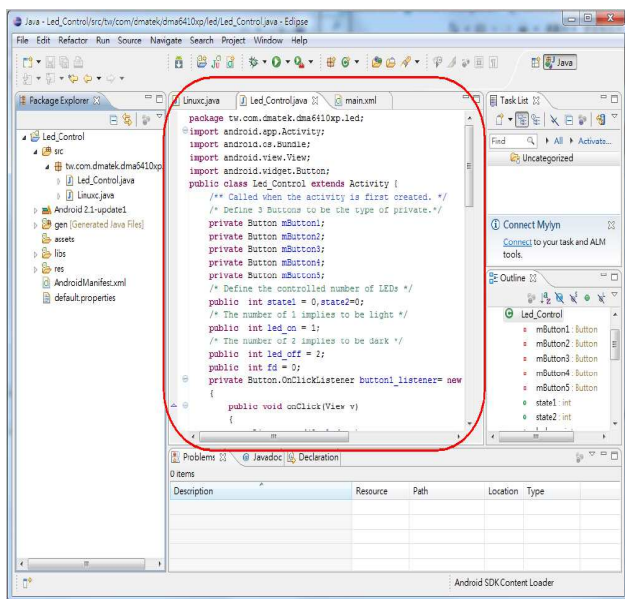


Fig. 8 The diagram of main program of Led_Control.java



Fig. 9 The diagram of generated program of Led_Control.apk

3. Experimental Results

The Samsung S3C6410 mobile processor gives designers an unbeatable combination of 3D performance and low power in a cost-effective package. This 32-bit ARM11 RISC microprocessor with AXI 64-bit bus delivers up to 667MHz of processing performance. With its 3D hardware accelerator handling 4 million polygons/second. It can power next-generation handhelds, such as mobile Internet devices and 3D UI-enhanced multimedia phones, as well as Personal Navigation Devices that display detailed images like buildings and landmarks. Memory support includes dual DRAM and flash/ROM external memory ports for parallel access. The DRAM port can support mobile DDR, standard SDRAM memory while the flash/ROM port supports NAND flash, NOR flash and ROM [7]. In this paper, the LED control is proceeded on the DMA 6410 kit which is shown in Fig. 10. The file of 'LED_control.apk' can be copied into SD card then go into the page of installation which is shown in Fig. 11. After

installation is completed, the experimental results are shown in Fig. 12 which show the light of LEDs and off of LEDs are successfully controlled by the buttons on the touch pad screen. One is shown with the all LEDs on and the other is shown with all LEDs off.



Fig. 10 The diagram of DMA 6410 kit



Fig. 11 The installation page of Android



Fig. 12 (a) The experimental result of all LEDs on



Fig. 12 (b) The experimental result of all LEDs off

4. Conclusions

In this paper, the Android and Eclipse have been installed successfully to develop LED control applications. This is verified by the ARM-11 based DMA 6410 kit. For

software, many tools are necessary such as Android JDK, SDK, Eclipse, ADT, etc. The development of LED control applications can be used to develop other different peripheral applications. The experimental results also reveal that the good LED control performances are possessed.

Acknowledgments

This research is partially funded by the teachers' research and cooperation with company research project of the Taoyuan Innovation Institute of Technology and supported by the SOC project of the Ministry of Education, Taiwan, R. O. C.

References

- [1]. Android user guide, <http://www.android.com>
- [2]. S. Mansfield-Devine, Paranoid Android: just how insecure is the most popular mobile platform, *Network Security*, Vol. 2012, Issue 9, September 2012, pp. 5-10.
- [3]. H. Hartmann, T. Trew and J. Bosch, The changing industry structure of software development for consumer electronics and its consequences for software architectures, *Journal of Systems and Software*, Vol. 85, Issue 1, 2012, pp. 178-192.
- [4]. G. Zhou, C. Li and Y. Hou, The Design of Ground Operation Control System Based on Red5 & Flex Original, *Physica Procedia*, Volume 33, 2012, pp. 521-527.
- [5]. T. Vidas, C. Zhang and N. Christin, Toward a general collection methodology for Android devices, *Digital Investigation*, Vol. 8, 2011, pp. S14-S24.
- [6]. V. S. Pascual and F. Xhafa, Evaluation of contact synchronization algorithms for the Android platform, *Mathematical and Computer Modelling*, 2012, doi:10.1016/j.mcm.2011.12.039.
- [7]. Samsung 6410 CPU data sheet, <http://www.samsung.com>.
- [8]. DMA 6410 user guide, <http://www.dmatek.com.tw>.
- [9]. J. Bharathi and S. Rao, "Remote computer access through Android mobiles," *International Journal of Computer Science Issues*, Vol. 9, Issue 5, No 3, September 2012, pp. 363-369.
- [10]. V. Monfort and S. Cherif, "Bridging the Gap between Technical Heterogeneity of Context-Aware Platforms: Experimenting a Service Based Connectivity between Adaptable Android, WComp and OpenORB," *International Journal of Computer Science Issues*, Vol. 8, Issue 4, No 2, July 2011, pp. 1-12.
- [11]. Y. Hasan, M. Zaidi, N. Haider, W. U. Hasan and I. Amin, "Smart Phones Application development using HTML5 and related technologies: A tradeoff between cost and quality," *International Journal of Computer Science Issues*, Vol. 9, Issue 3, No 3, May 2012, pp. 455-461.

Yi-Jen Mon is currently an Associate Professor of Department of Computer Science and Information Engineering, Taoyuan Innovation Institute of Technology, Taiwan, R. O. C. His research interests include fuzzy neural network control, intelligent control, automotive safety systems, wireless sensor network systems and embedded systems design.