## Content

1. Overview ................................................................................................................................. 1
2. Setup Demonstration ............................................................................................................... 3
3. Linux Application Project on DE10-Standard ................................................................. 6
4. Android Application Project .................................................................................................. 8
5. Appendix ................................................................................................................................ 13
1. Overview

This document describes how to an Android Smart Phone can remotely control the LEDs on the DE10-Standard board. The Bluetooth SPP (Serial Port Profile) is used as communication protocol between Smart Phone and DE10-Standard. Classical USB Bluetooth dongle is used to expand DE10-Standard Bluetooth capability.

- System Block Diagram

Figure 1 System Block Diagram shows the system block diagram of this demonstration. In the demonstration, we implement two Bluetooth application software. One is running on Android and one is running on DE10-Standard. These two application communicate with each other based on a proprietary command set.

In DE10-Standard, the Bluetooth application is running on the Linux. A classical USB Bluetooth dongle is used to expand DE10-Standard Bluetooth capability. The SPP (Serial Port Profile) protocol is used as communication protocol. The BlueZ Bluetooth protocol software stack is used by the application to handle the Bluetooth protocol. The Linux BSP build in the BlueZ and Bluetooth USB dongle driver, so user don’t need install to install any library or driver in this demonstration.

![System Block Diagram](image-url)
### Command Set

There two application communicates based on the following command set defined by Terasic. The commands are send by from Android Smart Phone to the DE10-Standard.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“ATLED0\n”</td>
<td>Turn one LED0 only</td>
</tr>
<tr>
<td>“ATLED1\n”</td>
<td>Turn one LED1 only</td>
</tr>
<tr>
<td>“ATLED2\n”</td>
<td>Turn one LED2 only</td>
</tr>
<tr>
<td>“ATLED3\n”</td>
<td>Turn one LED3 only</td>
</tr>
<tr>
<td>“ATON\n”</td>
<td>Turn on all four LEDs.</td>
</tr>
<tr>
<td>“ATOFF\n”</td>
<td>Turn off all four LEDs.</td>
</tr>
</tbody>
</table>
2. Setup Demonstration

This section describe how to setup the Bluetooth remote control demonstrations.

- **Hardware Requirements**

  The following hardware items are required to perform this demonstration:
  - DE10-Standard Board with shipped microSD card
  - Classical Bluetooth USB dongle
  - USB Mouse
  - USB hub with two ports at least.
  - Android Smart Phone with Bluetooth supporting
  - A microSD card writer is required if reprogramming the microSD card is necessary.

  The microSD card came with this kit already include the Linux application in the /home/root directory. If the microSD content is changed, please program the microSD card with the Linux Image File locate at [http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1046&PartNo=5](http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1046&PartNo=5).

- **Execute Demonstration**

  Here are show the procedure to setup the demonstration. First, the Bluetooth Application should be launched on DE10-Standard. Then, launch the Bluetooth Application on the Android phone. In the first time, the Android need to discovery the DE10-Standard and pair with it. When Android phone connects with the DE10-Standard, users can controls the four LEDs on DE10-Standard.

  Here are the procedure to launch the Bluetooth Application on the DE10-Standard:

1. Connect the HDMI port of DE10-Standard to a monitor as shown in [Figure 2](#).
2. Connect a USB hub into the micro USB port of DE10-Standard.
3. Plug the USB mouse to the USB hub.
4. Plug a Bluetooth USB dongle into USB hub.
5. Insert the microSD card, came with this kit, into the DE10-Standard.
6. Make sure MSEL[4:0] is 01010.
8. When the LXDE desktop appears on the monitor, use a mouse to double click the BT_LED_AP icon to launch the Bluetooth Application as shown in [Figure 3](#).
Here are the procedure to setup the demonstration on Android Smart Phone:

1. Download the TerasicBluetooth App from the QR code shown in Figure 4.
2. Install TerasicBluetooth.
3. Launch TerasicBluetooth.
4. For first time to connect DE10-Standard, click the ZOOM icon to discovery nearby devices, select DE10_Nano device, and pair it with password “1234” as shown in Figure 5.
5. Click the ZOOM icon and connect the paired DE10_Nano device.
6. In the TerasicBluetooth App GUI, click the LED0/LED1/LED2/LED3 and ON/OFF icons to control the on the DE10-Standard.
Note, the QR code link to:
http://www.terasic.com/downloads/demo/de10-nano/TerasicBluetooth.apk

Figure 4 BT_LED_APP QR-Code

Figure 5 Search Bluetooth Device
3. Linux Application Project on DE10-Standard

The Linux BT_LED_AP is a C++ project. Figure 6 show the main objects in the system. The CBtSppCommand object is used to receive the led control command from the Android smart phone. This object is running in a separated thread. The thread checks whether there is a coming command in polling method. The received commands are pushed into the CQueueCommand object.

The main program and the other objects are running in main thread. It will check the CQueueCommand whether there any queue command. If there are queue command, it will retrieve the command from the CQueueCommand object, and control CPIO_LED object to perform associated action.

![Figure 6 Major objects in the system](image)

### C++ Class

The major class used in this project as shown in the table below.

<table>
<thead>
<tr>
<th>Class</th>
<th>Function Description</th>
<th>Implementation Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>BtSppCommand</td>
<td>Parse the raw data coming from Bluetooth. Derive from the CBtSpp class.</td>
<td>BtSppCommand.cpp/h</td>
</tr>
<tr>
<td>BtSpp</td>
<td>Provide Bluetooth SPP service based on the RFCOMM Bluetooth stack in Linux kernel</td>
<td>BtSpp.cpp/h</td>
</tr>
<tr>
<td>CQueueCommand</td>
<td>Queue received Bluetooth</td>
<td>QueueCommand.cpp/h</td>
</tr>
<tr>
<td>command</td>
<td>Provide queue function</td>
<td>Queue.cpp/h</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CQueue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPIO_LED</td>
<td>Control the LEDs from HPS side of DE10-Standard</td>
<td>PIO_LED.cpp/h</td>
</tr>
</tbody>
</table>

### Build Project

Altera SoC EDS (Embedded Design Suite) is required to compile this C++ project. The C++ project is located at:

CD/Demonstrations/SoC_Advanced/Bluetooth_Spp/Linux_BT_App

Please follow the steps below to compile the project.

1. Make sure Altera SoC EDS v16.0 is installed on the host PC.
2. Copy the Linux_BT_App folder into the local hard disk of your host PC.
3. Launch Altera “SoC EDS Command Shell”.
4. In the shell, type “cd” command in the command shell to change the current directory to folder Linux_BT_App.
5. Type “make” to build the project, as shown in Figure 4.
6. The “BT_LED_AP” binary file will be generated in the project directory if compile is successful.

![Figure 7 Screenshot of build all](image)

### Test BT_LED_AP Binary File

Copy the generated BT_LED_AP binary file into the /home/root directory of DE10-Standard Linux. Then following the demonstration setup procedure to perform the test.
4. Android Application Project

The Android TerasicBluetooth project is Java-based project built by Eclipse. The main function of the Android TerasicBluetooth project is to receive user’s input from the GUI and send propriety commands to the spider robot through Bluetooth. The Android device should equity with Classical Bluetooth capacity for running this Bluetooth-based application. To open this project, Android Eclipse IDE is required. For installation details, please refer to the Appendix section in this document.

- **Android UI Layout and Components**

  **Figure 8** shows the User Interface of this application software. Two kinds of build-in components are used in this application software. They are **ImageView** and **ListView**. The **ImageView** is used to implement button functions. When users click a button, the associated command will be sent to DE10-Standard through Bluetooth. When DE10-Standard receives the command, it will change the LED status. **ListView** is used to log the command translated between Phone and DE10-Standard.

  ![Android User Interface layout](image)

  **Figure 8 Android User Interface layout**

- **Java Class**

  There are three Java Classes are used in this project. They are **Bluetooth**, **BluetoothService** and **deviceListActivity**. **Bluetooth** is the main Activity that handle the GUI event. **BluetoothService** provides all the work for setting up,
managing Bluetooth connector with other devices, and data transmission. DeviceListActivity appears as a dialog. It lists any paired devices and devices detected in the area after discovery.

If developers want to change GUI setting, they can modify the main.xml file under the layout folder in this application project.

### Build Project

Both Android SDK and Eclipse ADT Plugin must be installed to complete the installation for this project prior to the development of Android. For installation details, please refer to the Appendix section in this document.

Beside, developers need to create a new Android Device for their Android Smart Phones. In the Eclipse GUI, select the menu item “windows->Android Virtual Devices Manager->Device Definitions->Create Device” to create a new device as shown in Figure 9.

![Figure 9 New and Edit Android Device](image)

The Bluetooth project has to be imported prior to the start of building the project. In the Eclipsed, select the menu item “File->Import->Android->Existing Android Code into Workspace” to select the imported project type as shown in Figure 10.
In this Import Projects dialog, specify Bluetooth project location in the Root Directory edit box as shown in Figure 11. Then, click the Finish button.

After project is imported successfully, open the main gui file main.xml as shown in Figure 12. In this demonstration, Android Device is 5”1280x720 LCD, and API 19: Android 4.2.2 is used. Developers can change these setting according to their Android
Smart Phone. The binary file TerasicBluetooth.apk will be generated automatically when it will be download to the Android Smart Phone.

Figure 12 Project Main GUI

- **Download Binary File the Android Device**

Note, before the TerasicBluetooth.apk binary file can be downloaded to Android mobile through Eclipse tool, the Debug mode of the users’ Android Device must be enabled to allow the installation from an unknown source. The phone must also have the developer option turned on. The corresponding driver for users’ Android Device also needs to be installed on the host PC.

Connect your host PC and Android Smart Phone with a USB cable. To update binary file, right click on the project folder to pop up a menu, then select the menu item “Run As → Android Application” to download the demo file to Android mobile, as shown in Figure 13.
Run and Test Android Application

First, please launch the demo code on DE10-Standard. For details, please refer to Chapter 2. The following procedures below to the just download binary file.

1. Launch TerasicBluetooth application.
2. For first time to connect DE10-Standard, click ZOOM icon to discovery the DE10-Standard and pair it with pin-code “1234” as shown in Figure 14.
3. Click the ZOOM icon to connect the paired DE10-Standard.
4. In the TerasicBluetooth App GUI, click the LED0/LED1/LED2/LED3 and ON/OFF icons to control the on the DE10-Standard.
5. Appendix

The following items should be install on your host PC for developing Android Applications software. This section will describe how to download and install these items.

- Java JDK
- Eclipse
- Android SDK
- Android ADT

Install Java JDK

Go to web link below. In the web page, select Java Download icon as shown in Figure 15.


![Java SE Downloads](image)

Figure 15 JAVA SE Download Web

In the download web as shown in Figure 16, select proper installer for your Host PC.

![Java SE Development Kit 8u111](image)

Figure 16 JAVA SE Development Kit
- **Install Eclipse**

  Go to the web below. Select proper installer for your Host PC in the download page as shown in **Figure 17**. Execute the installer to install the Eclipse.

  http://www.eclipse.org/downloads/eclipse-packages/

  ![Figure 17 Eclipse Download Web](image)

  After Eclipse is installed successfully, launch the Eclipse. A the Workspace Launcher dialog will appear as shown in **Figure 18**, input project folder and then click OK.

  ![Figure 18 Workspace directory for Eclipse](image)

- **Install ADT (Android Development Tools)**
In the Eclipse, select the menu item “Help->Install New Software …” as shown in Figure 19.

![Figure 19 Install New Software](image)

The Install dialog will appear as shown in Figure 20. In the dialog, click the “Add…” button to popup an Add Repository dialog. In the Add Repository dialog, type in the following information for Name and Location edit box:

Name: Android
Location: [https://dl-ssl.google.com/android/eclipse/](https://dl-ssl.google.com/android/eclipse/)

![Figure 20 Install ADT](image)

In the Install Dialog, check the Developer Tools check box, then click the Next button to start downloading Android ADT as shown in Figure 21.
After the installations completed, restart the Eclipse.

- **Install Android SDK**

  Go to the web link below to download the Android SDK as shown in **Figure 22**. Then uncompressed the downloaded zip file.


  ![Figure 22 Android SDK Download](image)

  In the Eclipse, select the menu item “Windows > Preference” to open the Preferences dialog as shown in **Figure 23**. In the dialog, select Android tab. In the SDK Location edit box, input the folder location where the uncompressed Android SDK is located.
Then, we need to select the desired API for your Android Smart Phones. In Eclipse, select the menu “Windows → Android SDK Manager” to popup the Android SDK Manager dialog as shown in Figure 24. In the dialog, select the desired API package. In this demonstration, Android 4.2.2 (API 19) is used.