# DE10-Nano

# **Bluetooth SPP**





Copyright © 2003-2017 Terasic Inc. All Rights Reserved.

# Content

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



# 1. Overview

This documents describe how to an Android Smart Phone can remotely control the LEDs on the DE10-Nano board. The Bluetooth SPP (Serial Port Profile) is used as communication protocol between Smart Phone and DE10-Nano. Classical USB Bluetooth dongle is used to expand DE10-Nano 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-Nano. These two application communicate with each other based on a propriety command set.

In DE10-Nano, the Bluetooth application is running on the Linux. A classical USB Bluetooth dongle is used to expand DE10-Nano 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.



Figure 1 System Block Diagram



# 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-Nano.

Command	Description
"ATLED0\n"	Turn one LED0 only
"ATLED1\n"	Turn one LED1 only
"ATLED2\n"	Turn one LED2 only
"ATLED3\n"	Turn one LED3 only
"ATON\n"	Turn on all four LEDs.
"ATOFF\n"	Turn off all four LEDs.



# 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-Nano 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 <u>http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1046&PartNo=5</u>.

#### Execute Demonstration

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

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

- 1. Connect the HDMI port of DE10-Nano to a monitor as shown in Figure 2.
- 2. Connect a USB hub into the micro USB port of DE10-Nano.
- 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-Nano.
- 6. Make sure MSEL[4:0] is 01010.
- 7. Power on DE10-Nano.
- 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**.





Figure 2 DE10-Nano Demo Setup



Figure 3 Launch BT\_LED\_AP on LXDE

Here are the procedure to setup the demonstration on Android Smart Phone:

- 1. Download the TerasicBluetooth App from the QR code shown Figure 4.
- 2. Install TerasicBluetooth.
- 3. Launch TerasicBluetooth.
- For first time to connect DE10-Nano, 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 TerasicBlueooth App GUI, click the LED0/LED1/LED2/LED3 and ON/OFF icons to control the on the DE10-Nano.



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-Nano

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

# C++ Class

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

Class	Function Description	Implementation Files
BtSppCommand	Parse the raw data coming from Bluetooth. Derive from the CBtSpp class.	BtSppCommand.cpp/h
BtSpp	Provide Bluetooth SPP service based on the RFCOMM Bluetooth stack in Linux kernel	BtSpp.cpp/h
CQueueCommand	Queue received Bluetooth	QueueCommand.cpp/h



	command	
CQueue	Provide queue function	Queue.cpp/h
CPIO_LED	Control the LEDs from HPS side of DE10-Nano	PIO_LED.cpp/h

## 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
- The "BT\_LED\_AP" binary file will be generated in the project directory if compile is successful.



#### Figure 7 Screenshot of build all

# ■ Test BT\_LED\_AP Binary File

Copy the generated BT\_LED\_AP binary file into the /home/root directory of DE10-Nano 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-Nano through Bluetooth. When DE10-Nano receives the command, it will change the LED status. List View is used to log the command translated between Phone and DE10-Nano.



#### Figure 8 Android User Interface layout

#### Java Class

There are three Java Classes are used in this project. They are Bluetooth, BluetoothService and eviceListActivity. Blueotooth is the main Activity that handle .the GUI event. BluetoothService provides all the work for setting up,



DE10-Nano Bluetooth SPP www.terasic.com February 24, 2017 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**.

Edit Device			
Name:	zenfone	Size:	large 🗸
Screen Size (in):	5.0	Screen Ratio:	long -
Resolution (px):	1280 × 720	Density:	xhdpi 🗸
Sensors:	✓ Accelerometer  Gyroscope	Buttons:	Hardware -
Cameras	GPS Proximity Sensor		Portrait: Enabled Vavigation Landscape:
Input:	✓ Keyboard ○ No Nav  ● DPad  ○ Trackball	Device States:	Enabled Vavigation     Portrait with keyboard:     Enabled Vavigation     Landscape with keyboard:
RAM:	512 MiB -		✓ Enabled ✓ Navigation
		Override th	e existing device with the same name
			Edit Device Cancel

#### Figure 9 New and Edit Android Device

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**.





Figure 10 Select Imported Project Type

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.

•		- • •		
Import Projects Select a directory to search for existing Android projects				
Coot Directory: C:\TerasicBlueto	ooth	Browse		
Projects:				
Project to Import	New Project Name	Select All		
C:\TerasicBluetooth	TerasicBluetooth	Deselect All		
		Kefresh		
Copy projects into workspace	$\geq$			
Working sets				
Add project to working sets				
Working sets:		v Select		
? < Back	Next > Finish	Cancel		

Figure 11 Root Directory for Imported Project

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 $\rightarrow$ Android Application" to download the demo file to Android mobile, as shown in **Figure 13** 





#### Figure 13 Select menu item "Run As→Android Application"

# Run and Test Android Application

First, please launch the demo code on DE10-Nano. 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-Nano, click ZOOM icon to discovery the DE10-Nano and pair it with pin-code "1234" as shown in Figure 14.
- 3. Click the ZOOM icon to connect the paired DE10-Nano.
- 4. In the TerasicBlueooth App GUI, click the LED0/LED1/LED2/LED3 and ON/OFF icons to control the on the DE10-Nano.







# 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** 

http://www.oracle.com/technetwork/java/javase/downloads/index.html



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						
ary Code Licen	se Agreement for Java SE to download this					
softwa	re.					
se Agreement	Decline License Agreement					
File Size	Download					
77.78 MB	jdk-8u111-linux-arm32-vfp-hflt.tar.gz					
74.73 MB	jdk-8u111-linux-arm64-vfp-hflt.tar.gz					
160.35 MB	🚽 jdk-8u111-linux-i586.rpm					
175.04 MB	jdk-8u111-linux-i586.tar.gz					
158.35 MB	jdk-8u111-linux-x64.rpm					
173.04 MB	jdk-8u111-linux-x64.tar.gz					
227.39 MB	jdk-8u111-macosx-x64.dmg					
131.92 MB	jdk-8u111-solaris-sparcv9.tar.Z					
93.02 MB	jdk-8u111-solaris-sparcv9.tar.gz					
140.38 MB	jdk-8u111-solaris-x64.tar.Z					
96.82 MB	jdk-8u111-solaris-x64.tar.gz					
189.22 MB	jdk-8u111-windows-i586.exe					
194.64 MB	jdk-8u111-windows-x64.exe					
	E Develop ry Code Licen softwa se Agreement File Size 77.78 MB 74.73 MB 160.35 MB 175.04 MB 158.35 MB 173.04 MB 227.39 MB 131.92 MB 93.02 MB 140.38 MB 96.82 MB 189.22 MB 194.64 MB					

#### 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

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

# 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

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/

🖨 Install					
Available S	oftware				
Check the i	tems that you wish to	o install.			
Work with:	TEST - https://dl-ss	l.google.com/android/eclipse 🛛 🔶			- Add
			Find more software by wo	rking with the <u>"Availab</u>	le Software Sites" preferences.
type filter te	ext				
Name			Version		
> 🗐 000 C	Developer Tools				
		C Add Repository		<b>×</b>	
		Name: Android		Local	
		Location: https://dl-ssl.google.com	m/android/eclipse/	Archive	
		?	ОК	Cancel	
Select Al	Deselect All				
Details					
					*
					•
Show only	y the latest versions o	of available software	Hide items that are a	already installed	
Group ite	ms by category		What is already insta	lled?	
Show only	y software applicable	e to target environment			
Contact a	ll update sites during	install to find required software			
E contacta	in operate sites during				
0			< Back	Next >	Finish Cancel

#### Figure 20 Install ADT

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**.



🕽 Install			
Available Check the i	Software tems that you wish to install.		
Work with:	Android - https://dl-ssl.google.com/andro	oid/eclipse/site.xml 👻	Add
	Find mo	ore software by working with the <u>"Available Software Si</u>	tes" preferences
type filter te	xt		
Name		Version	
🕑 🔽 000 E	Developer Tools		
Select Al	Deselect All 3 items selected		
Details			
Show only	y the latest versions of available software	Hide items that are already installed	1
Group ite	ms by category	What is <u>already installed</u> ?	
Contact a	Il update sites during install to find required	l software	
?		< Back Next > Finish	Cancel

Figure 21 ADT Development Kit

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.

Platform	SDK tools package		SHA-1 checksum
Windows	tools_r25.2.3-windows.zip	292 MB (306745639 bytes)	b965decb234ed793eb9574bad8791c50ca574173
Mac	tools_r25.2.3-macosx.zip	191 MB (200496727 bytes)	0e88c0bdb8f8ee85cce248580173e033a1bbc9cb
Linux	tools_r25.2.3-linux.zip	264 MB (277861433 bytes)	aafe7f28ac51549784efc2f3bdfc620be8a08213

https://developer.android.com/studio/index.html

#### Figure 22 Android SDK Download

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.Figure 23



	Android		<	• 🗘 •
General Android Ant Help	Android Preferences SDK Location C:\Andr Note: The list of SDK Tar	oid\android-sdk-windows	Apply' or 'OK'.	Browse
Install/Update Java	Target Name	Vendor	Platform	API Le.
Java Plug-in Development Run/Debug Team XML	-	No target available		
)			Restore Defaults	Apply

Figure 23 Android SDK installation location

Then, we need to selected 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.

<ul> <li>Java - SuperCar/src/com/example/android/B</li> <li>File Edit Refactor Source Navigate Sear</li> </ul>	luetoothChat/B rch Project F	etoothChatjava - Eclipse in Window Help	
i 📬 🕶 🔚 🐁 🗁 🔍 🚺 🚔 i 🗹 🕶 i	🗟 i 🧐 🏄 📚	■ <b>Π</b>   ☆ ▼ <b>O</b> ▼ <b>Q</b> ▼   ⊕ <b>Ø</b> ▼   ⊕ <b>Ø</b> ▼   ⊕ ▼ <del>Ø</del> ▼ <b>♥</b> ↓ ▼ → ▼ Quick	Access 🔡 😰 🐉 Java 🚳 DDMS
😫 Package E 😒 🦹 Type Hier 🖳 🗖	🔝 BluetoothC	at.java 🛛 💭 Bluetooth.java 🔄 🔐 BluetoothChat.java 🔍 🗖 🗐	Task List 🔀 📃 🗖
E 😫   🗊 マ ▷ 🖉 appcompat_v7 ▷ 🚰 BluetoothTMD	76 pi 79 // 80 pi 81 //	Android SDK Manger Colors	
<ul> <li>S<sup>D</sup> DE1SOC</li> <li>S<sup>D</sup> DEMO SPIDER</li> <li>MttpRequestExample</li> </ul>	82 pr 83 84	SDK Pathr. C.\Users\User\android-sdks Packages	Monday - Today
<ul> <li>p) gill Seekläär</li> <li>p) gill SenaturGarlemote</li> <li>p) gill SenaturGarlemote</li> <li>p) gill TensicBluetooth</li> <li>p) gill Android 4.4.2</li> <li>g) gill second semaple android Bluetoothi</li> <li>p) gill Sluetooth java</li> <li>p) gill Sluetooth Java</li> <li>p) gill Sluetooth Archivin java</li> <li>p) gill Sluetooth Archivin java</li> <li>g) gill Sluetooth Archivin java</li> <li>g</li></ul>	85⊖ @( 86 pr 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 }	Name         API         Rev.         Status           >         C         Android 5.11 (API 22)            >         C         Android 4.07 (API 23)            >         C         Android 4.21 (API 13)            >         C         Android 4.22 (API 16)            >         C         Android 4.03 (API 13)            >         C         Android 4.03 (API 13)            >         C         Android 4.03 (API 13)            >         C         Android 4.03 (API 14)            >         C         Android 4.03 (API 13)            >         C         Android 4.0 (API 14)            >         C         Android 3.1 (API 12)	Wednesday       Thunday       Finday       Saturday       This Week       Dancet My/m       Saturday       This Week       Cast ask       dime 33       Image 1/2, Saturday       a       mchatSenrice: Bluetoothr       a       onCreate(Bundle): void
a) Androidhiantetmi ■ c jauncherweb.png project.properties ≥ Ø Widgets	104⊕ @C 105 pt 106 107 108 109 110 111 112 113 114	Show 🗹 Updates/New 🗹 Installed Select New or Updates Install 24 packages.  O Obvolete Desident All Delete 21 packages.  Done loading packages.  Intent enableIntent = new Intent(Bluetooth/dapter.ACTION_REQUEST_ENABLE_oT); startActivityForResult(enableIntent, REQUEST_ENABLE_OT); () Otherwise, section the chart section	G onResume(): void     setup(Cht(): void     or, onStop(): void     or, onStop(): void     or, onStop(): void     or, onStop(): void     ensureDiscoverable(): voi     sendMessage(Sking): void     of setStatus(On2: void     of setStatus(CharSequence):

Figure 24 Android SDK Manager