

Java HID Demo App for MSP430™ MCUs

MSP430 Applications

1 Introduction

The Java HID Demo Application is a host side application that provides a GUI for communicating with a HID device. It simplifies the creation of a general purpose USB HID device. It uses the MSP430's HID-Datapipe format, but it can also be used for implementing any custom HID devices not directly driven by the OS. (A mouse or a keyboard are examples of HID devices driven directly by the OS.)

The Java HID Demo App is part of the MSP430 USB Developers Package and can be downloaded from <http://www.ti.com/tool/msp430usbdevpack>.

2 System Requirements

See the 'release_notes.html' file included in the Java_Hid_Demo/Windows or /Linux project folder for system requirements.

3 HID Demo Project

The HID Demo App's project is composed of a Java GUI and C drivers. The communication between Java and the C drivers is based on Java Native Interface (JNI). This means there is an accompanying DLL (Windows) or SO (Linux) in the same path as the *.jar file that contains the native calls.

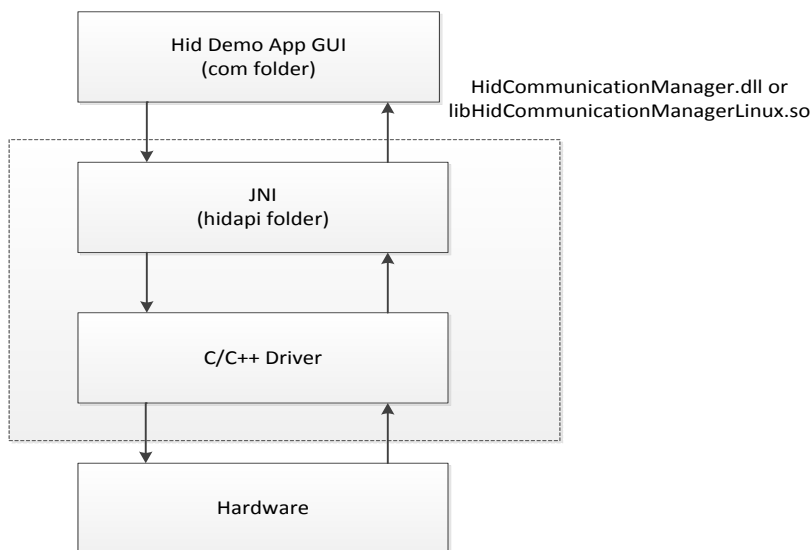


Figure 1. Hid Demo App Architecture

The Windows project folder contains the following items:

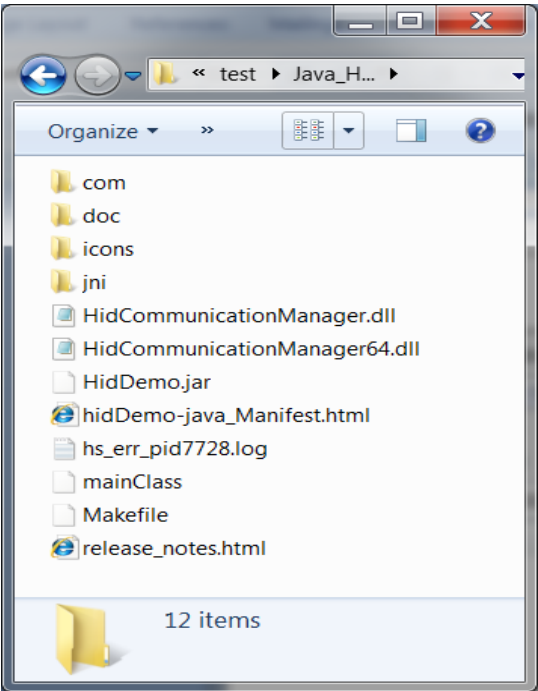


Figure 2. Windows Project Structure

The Linux project folder contains the following items:

File System	com	1 item	folder	Mon 29 Sep 2014 1
Network	doc	10 items	folder	Mon 29 Sep 2014 1
DVD+R Disc	icons	12 items	folder	Mon 29 Sep 2014 1
Floppy Drive	jni	2 items	folder	Mon 29 Sep 2014 1
4.2 GB Filesystem	lib	1 item	folder	Mon 29 Sep 2014 1
Trash	HidDemo.jar	1.6 MB	Java archive	Mon 29 Sep 2014 1
Documents	hidDemo-java_Manifest.html	25.0 KB	HTML document	Mon 29 Sep 2014 1
Music	libHidCommunicationManagerLinux.so	32.9 KB	shared library	Mon 29 Sep 2014 1
Pictures	mainClass	48 bytes	plain text document	Mon 29 Sep 2014 1
Videos	Makefile	2.2 KB	Makefile	Mon 29 Sep 2014 1
Downloads	release_notes.html	8.8 KB	HTML document	Mon 29 Sep 2014 1

Figure 3. Linux Project Structure

The main Java files are contained in the ‘com’ directory, whereas all the JNI code are located in the ‘jni’ folder. The ‘jni’ folder contains the open source ‘hidapi’ software downloaded from <https://github.com/signal11/hidapi/downloads>.

In the Windows version of the tool, the project folder also contains two DLLs – one for 32-bit OS and the other for 64-bit OS. Depending on the version (32-bit or 64-bit) of the JDK or JRE used to bring up the GUI, the tool automatically selects the appropriate DLL.

In the Linux version of the tool, the project folder contains two SOs – one for 32-bit OS and the other for 64-bit OS. Both the 32-bit and 64-bit SOs can be created on either a 32-bit or 64-bit Linux box by running the Makefile included in the Linux project folder. When the Makefile is run, both the .jar file and the .so file for the selected OS bit version are created. Depending on the version (32-bit or 64-bit) JDK installed on the OS, the tool automatically selects the appropriate SO.

4 Running the Demo

See the 'release_notes.html' file included in the Java_Hid_Demo folder for how to run the HID Demo App.

4.1 Using HID Demo App

This section gives some tips on using the HID Demo App. For additional information in the context of using the USB examples, see the Examples Guide within the USB Developers Package.

If no USB device with the default VID/PID is connected to the host, the HID Demo App will display the following screen to indicate an error:

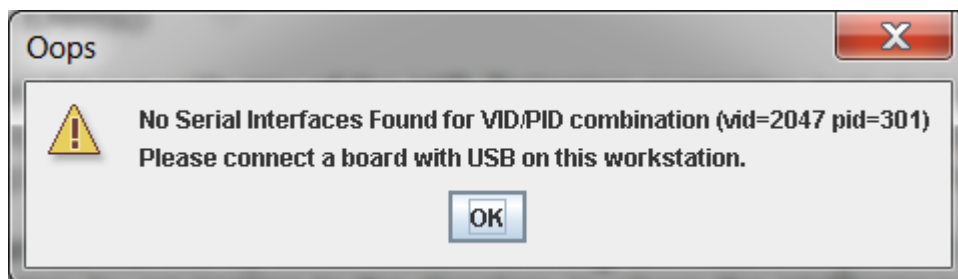


Figure 4. No Device Found, for the Selected VID/PID

When a USB device with the selected VID/PID is present on the USB host, the HID Demo App displays the main screen:

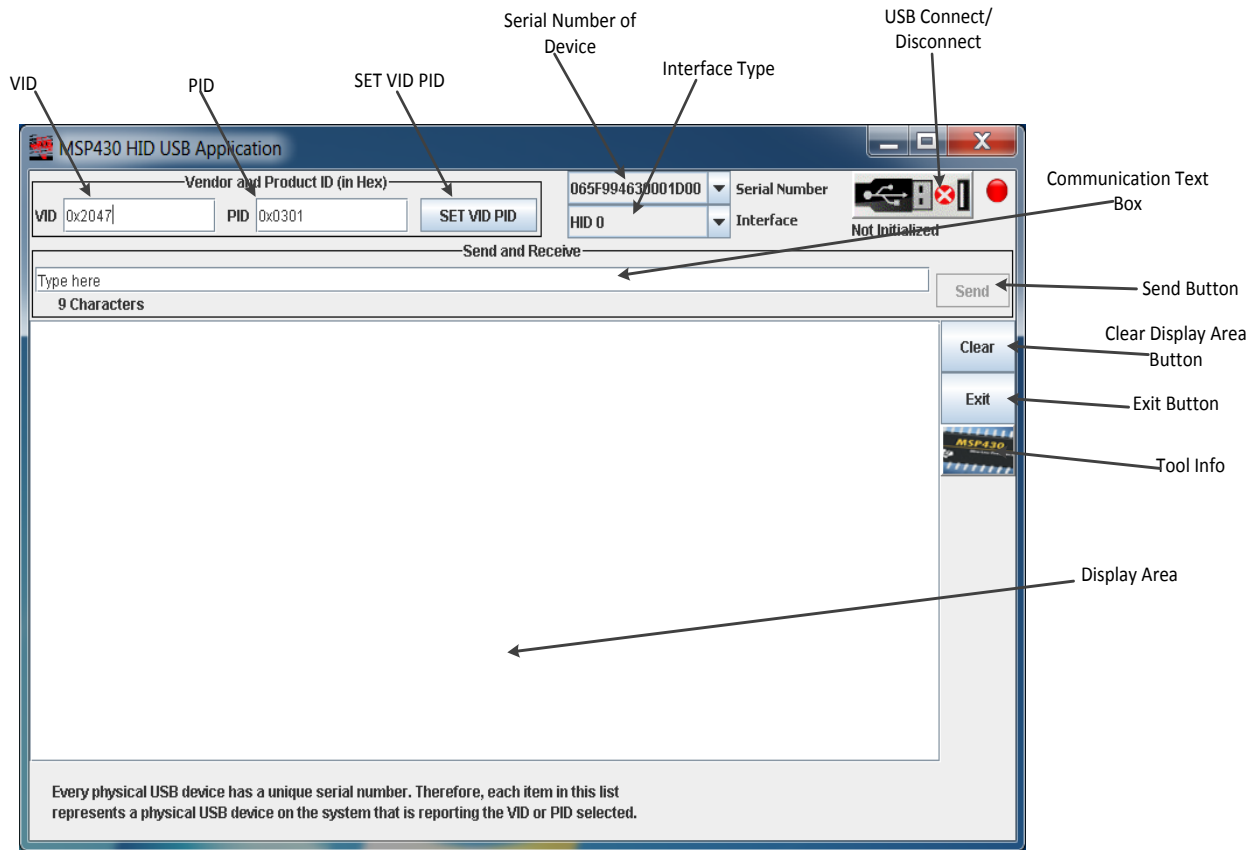


Figure 5. HID Demo Initial Screen

Enter the appropriate VID and PID for the device connected to the computer, and click on 'SET VID PID' for the tool to display the correct Serial Number and Interface. Once the USB Connect/Disconnect button is clicked, the GUI is connected to the device and the following screen is displayed:

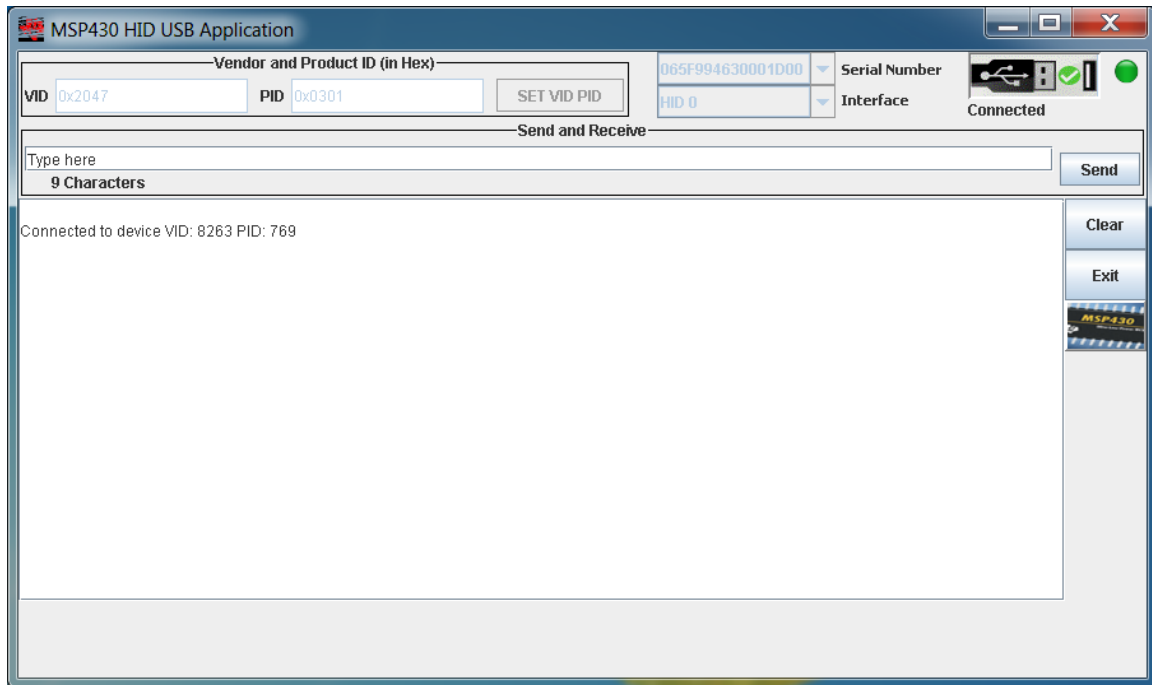


Figure 6. HID Demo App GUI Connected to a USB HID-Datapipe Device

NOTE: The VID and PID displayed in the display area are in decimal format.

The user is now able to communicate with the device by typing the text in the communication text box. See the Examples Guide (Examples_Guide_MSP430_USB.pdf, in the USB Developers Package) for information on using the HID Demo App with the USB examples.

An example is shown below, of the HID Demo App communicating with the MSP430F5529 LaunchPad, loaded with the USB example #H1_LedOnOff.

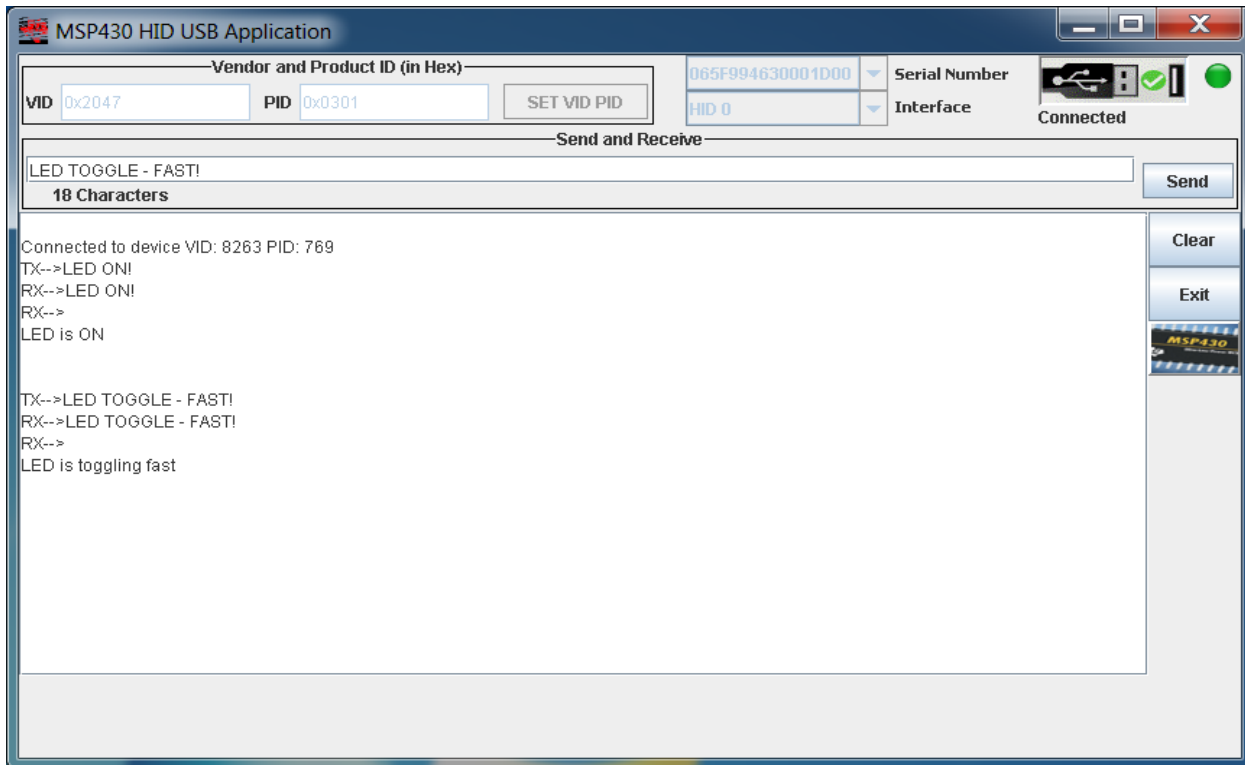


Figure 7. Communication Example with HID Device

5 Rebuilding the Demo

On Windows the HID Demo App should be built using cygwin. On Linux, a terminal window will suffice. The Makefile that is in the project folder allows for selecting either to build the 32-bit or 64-bit versions of the *.jar file and DLL/SO file of the tool. After the system requirements for re-building the tool are met, the user can run the Makefile in one of two ways:

- By typing 'make' at the command line to build the 32-bit version
- By typing 'make JDK64=1 or make Linux=1' at the command line to build the 64-bit version

If 32-bit version is selected, verify that the JDK pointed to in the Makefile is for 32 bit version of java installed on the host computer. If 64-bit version is selected, verify that the JDK pointed to in the Makefile is for 64 bit version installed on the computer.

On Windows 7, the 64-bit version of Java is installed in the default folder 'Program Files', and the 32-bit version of Java is installed in the default folder 'Program Files (x86)'.

For additional information on re-building the Hid Demo, see the 'release_notes.html' file included in the Java_Hid_Demo project folder.

6 References

- *MSP430F5xx Family User's Guide (SLAU208)*
- <http://www.ti.com/430usb>