February 19, 2018

Java HID Demo App for MSP432™ MCUs

MSP432 Applications

1 Introduction

The Java HID Demo Application is a host side application that provides a GUI for communicating HID Temperature Sensor device. It simplifies the creation of a general purpose USB HID device.

2 System Requirements

See the 'release_notes.html' file included in the 'usb_dev_hid_sensor' 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) , SO (Linux) or Mac(dylib) in the same path as the *.jar file that contains the native calls.

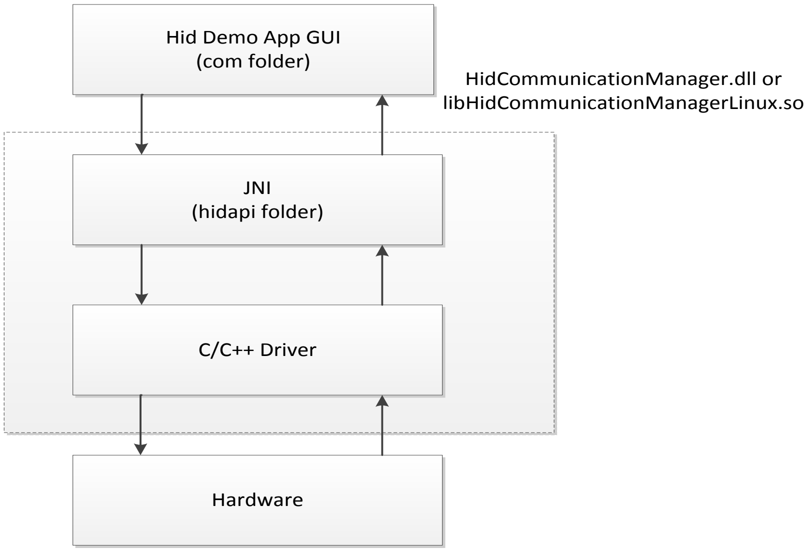


Figure 1. Hid Demo App Architecture

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 contains a 64-bit .dll.

In the Linux version of the tool, the project folder contains a 64-bit SO file.

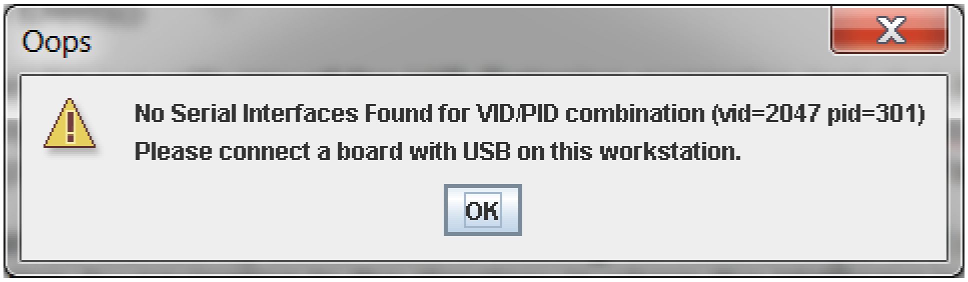
4 Running the Demo

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.

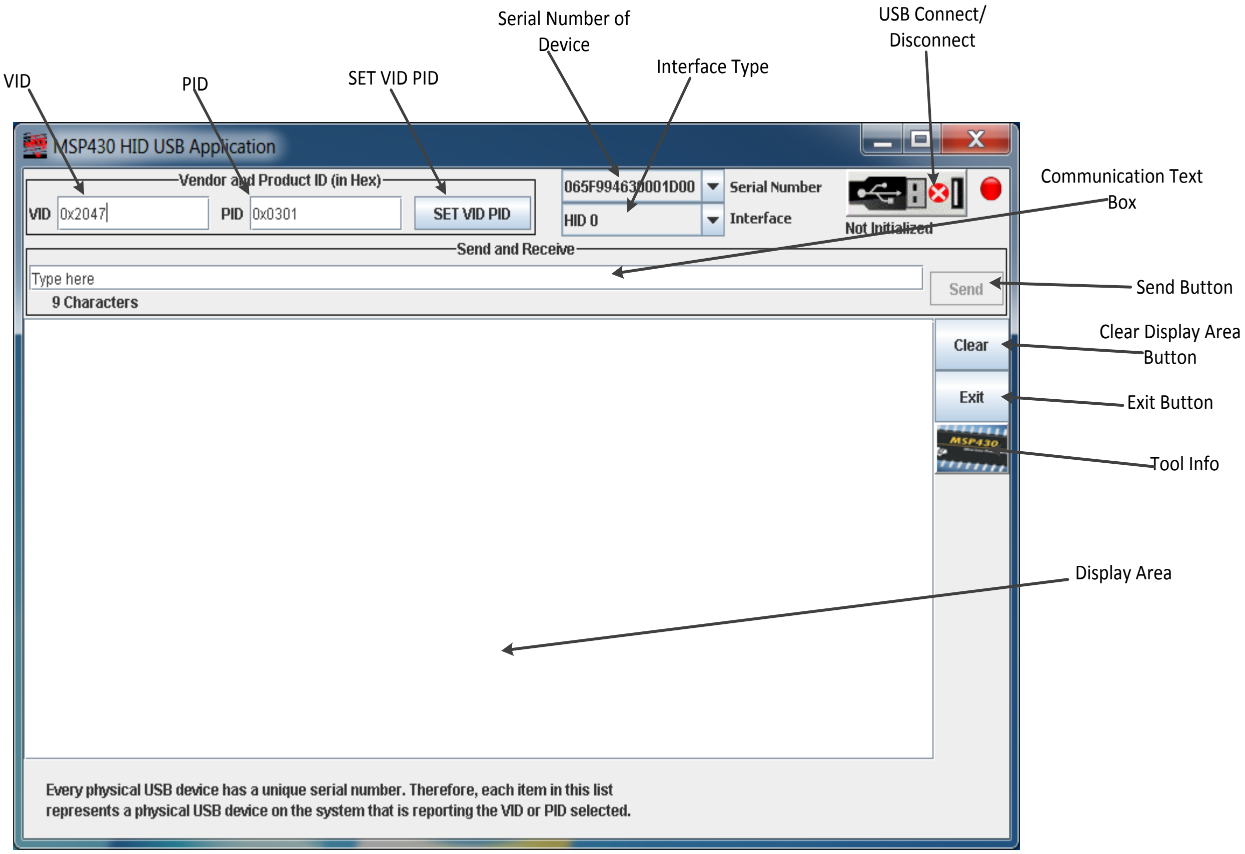
If no USB device with the default VID/PID is connected to the host, the HID Demo App will display the following screen to indicate an error:



No Serial Interfaces Found for VID/PID combination (vid=2047 pid=301)
Please connect a board with USB on this workstation.

Figure 2. 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:



VID 0x2047 PID 0x0301

065F994630001D00 Serial Number

HID 0 Interface

Not Initialized

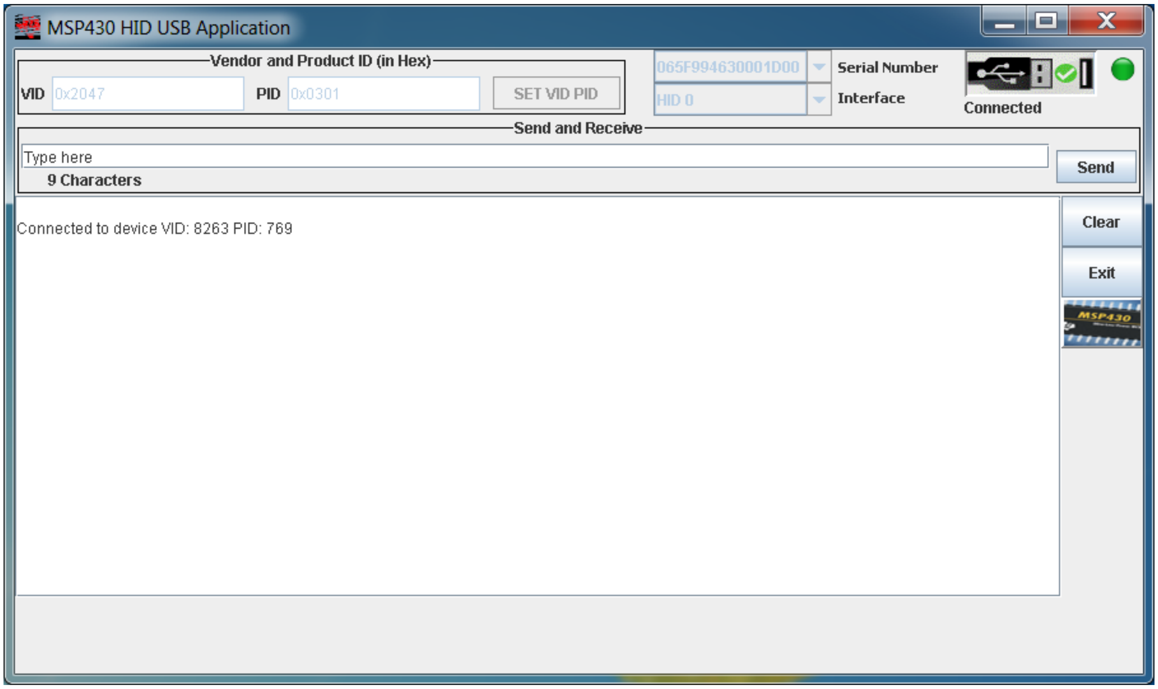
Type here

9 Characters

Every physical USB device has a unique serial number. Therefore, each item in this list represents a physical USB device on the system that is reporting the VID or PID selected.

Figure 3. 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:



VID 0x2047 PID 0x0301

Serial Number 065F994630001D00

Interface HID 0

Type here

9 Characters

Connected to device VID: 8263 PID: 769

Figure 4. 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.

5 Rebuilding the Demo

On Windows the HID Demo App should be built using cygwin. On Linux, a terminal window will suffice. The Makefile (see release_notes.html for makefile script) is run by 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 for Windows and Linux 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 usb_dev_hid_sensor project folder.