March 21, 2017

Java HID Demo App for MSP430™ MCUs with AES-128 Encryption and Decryption

MSP430 Applications

1 Introduction

The Java HID Demo Application is a host side application that provides a GUI for communicating with a HID device that performs encryption and decryption of its HID data. 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_AES128 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). The JNI layer is also where data is encrypted and decrypted using TI's AES-128 APIs. This means there is an accompanying DLL (Windows) in the same path as the *.jar file that contains the native calls.

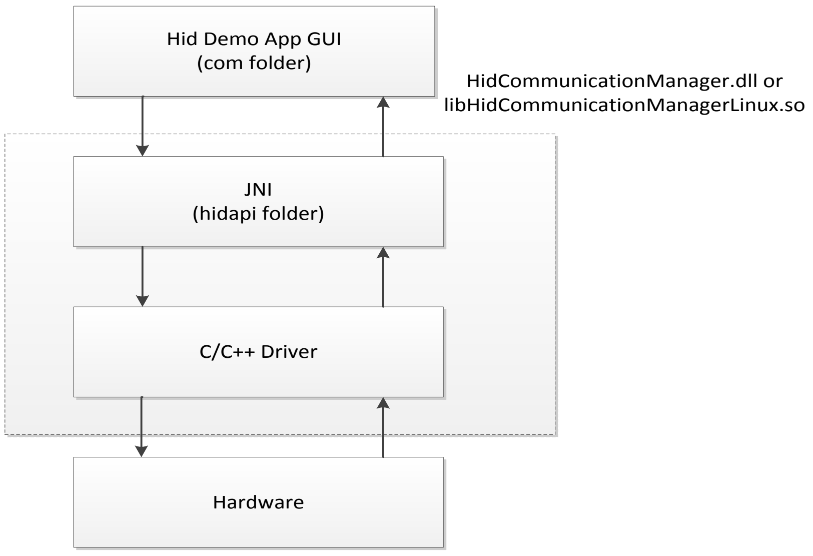
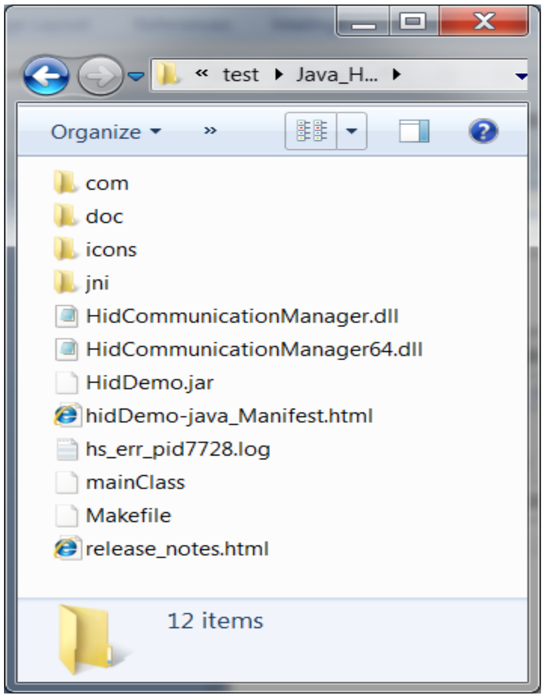


Figure 1. Hid Demo App Architecture

The Windows project folder contains the following items:



com
doc
icons
jni
HidCommunicationManager.dll
HidCommunicationManager64.dll
HidDemo.jar
hidDemo-java_Manifest.html
hs_err_pid7728.log
mainClass
Makefile
release_notes.html

12 items

Figure 2. Windows 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.

4 AES128 Encryption, Decryption and Keys

Encryption or decryption of HID data is performed in the JNI file, '*com_ti_msp430_usb_hiddemo_management_HidcommunicationManager.c'.*

Encryption of data is implemented in the JNI call *Java_com_ti_msp430_usb_hiddemo_management_HidCommunicationManager_sendDataNative()* and decryption of data is implemented in '*Java_com_ti_msp430_usb_hiddemo_management_HidCommunicationManager_receiveDataNative()*'

Since the AES128 APIs require a key of 16 bytes in length, the HidDemo application has been updated to encrypt or decrypt data in 16 byte increments up to 48 bytes at a time.

The 16 byte Key (that can be personalized by the user) should be the same for the device application as well as for the HidDemo tool. The Key is defined in the JNI file. If the Keys are different for either device or HidDemo tool, encryption and decryption of data will not happen.

Currently, the HidDemo tool is able to only encrypt and decrypt data lengths of up to 48 bytes.

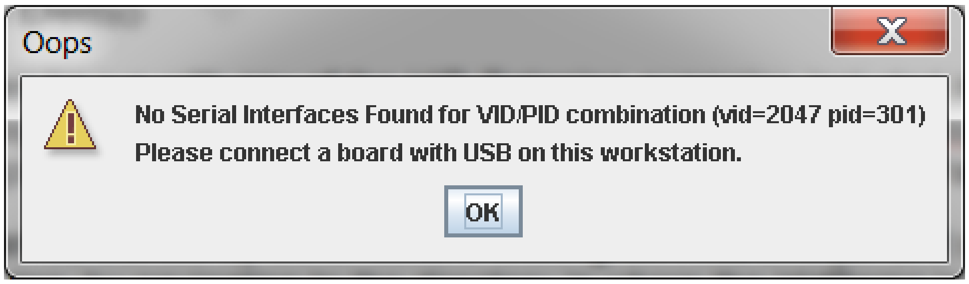
5 Running the Demo

See the 'release_notes.html' file included in the Java_Hid_Demo/Windows_AES128 folder on how to run the HID Demo App.

5.1 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:

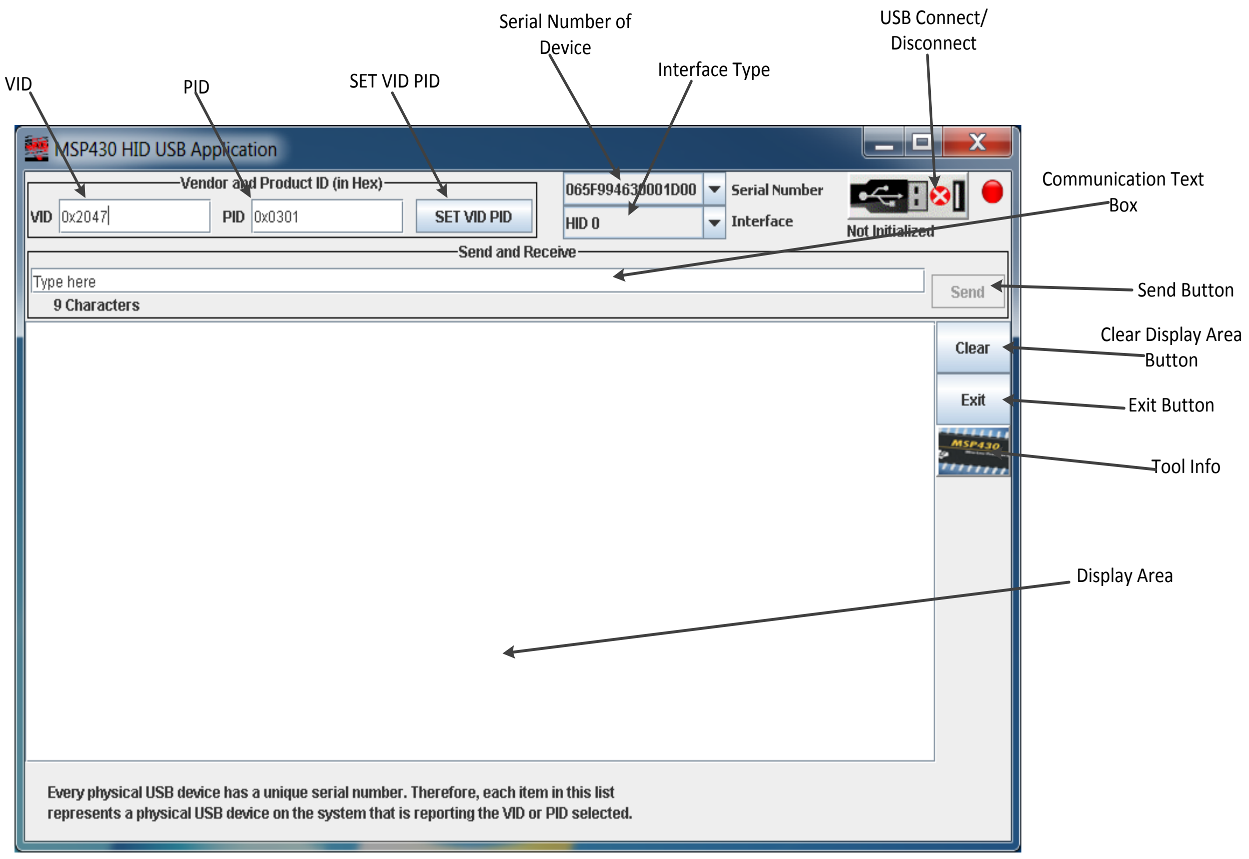


Oops

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

Figure 3. 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

HID 0

Not Initialized

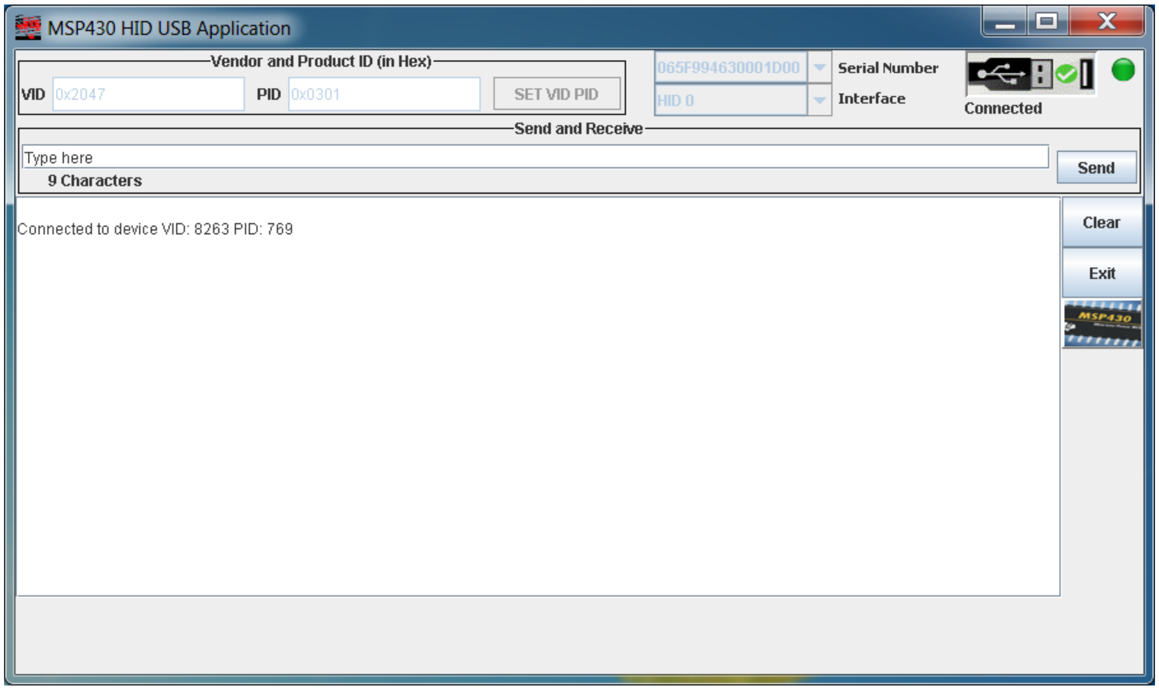
Type here

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 4. 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:



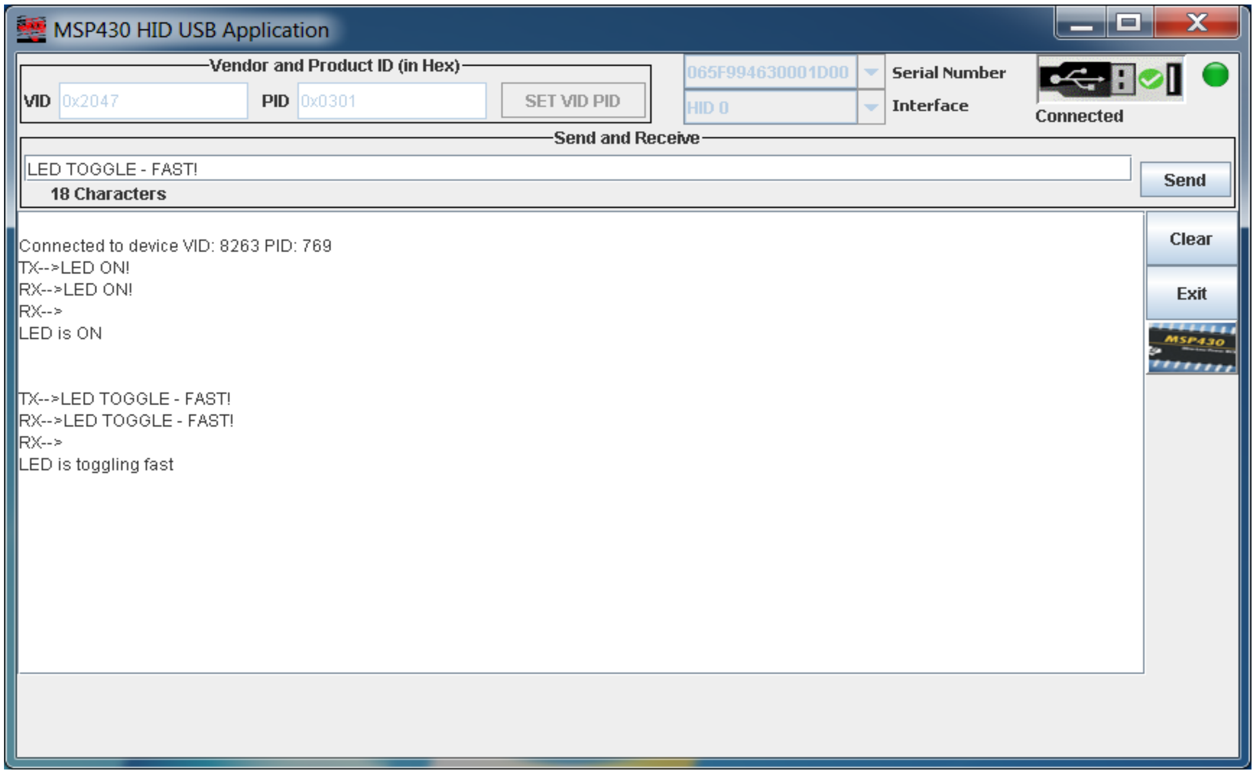
Connected to device VID: 8263 PID: 769

Figure 5. 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, *H10_ReceiveData_EncryptDecrypt* and *H11_LedOnOff_EncryptDecrypt* only.

An example is shown below, of the HID Demo App communicating with the MSP430F5529 LaunchPad, loaded with the USB example *#H11_LedOnOff_EncryptDecrypt*:



```
Connected to device VID: 8263 PID: 769
TX-->LED ON!
RX-->LED ON!
RX-->
LED is ON


TX-->LED TOGGLE - FAST!
RX-->LED TOGGLE - FAST!
RX-->
LED is toggling fast
```

Figure 6. Communication Example with HID Device

6 Rebuilding the Demo

On Windows the HID Demo App should be built using cygwin. 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 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/Windows_AES128 project folder.

7 References

- *MSP430F5xx Family User's Guide (SLAU208)*
- http://www.ti.com/430usb
- *AES128 - A C Implementation for Encryption and Decryption (SLAA397A)*