Overview

This Host HID example can suport mouse device and keyboard device.
The application will print the mouse operation when mouse device is attached. The application will print the pressed keyboard key when keyboard is attached. As a simple demo, some special function keys beyond regular and long press function are not supported.

The application support three types of device:

- single mouse device or single keyboard device.
- composite device that contain one mouse and one keyboard.
- hubs connected with one mouse device and one keyboard device.

System Requirement

Hardware requirements

- J-Link ARM
- P&E Micro Multi-link universal
- Mini/micro USB cable
- USB A to micro AB cable
- Hardware (tower/base board, ...) for specific device
- Personal Computer(PC)

Software requirements

- The project path is:
  <SDK_Install>/boards/<board>/usb/usb_host_hid_mouse_keyboard/<RTOS>/<toolchain>.

> **Note**
> The RTOS is BM or FreeRTOS.

Getting Started

Hardware Settings

- The Jumper settings:
  JP12 connected .

Prepare the example

1. Download the program to the target board.
2. Power off the target board. And then power on again.
3. Connect devices to the board.

Note

For detailed instructions, see the appropriate board User's Guide.

Run the example

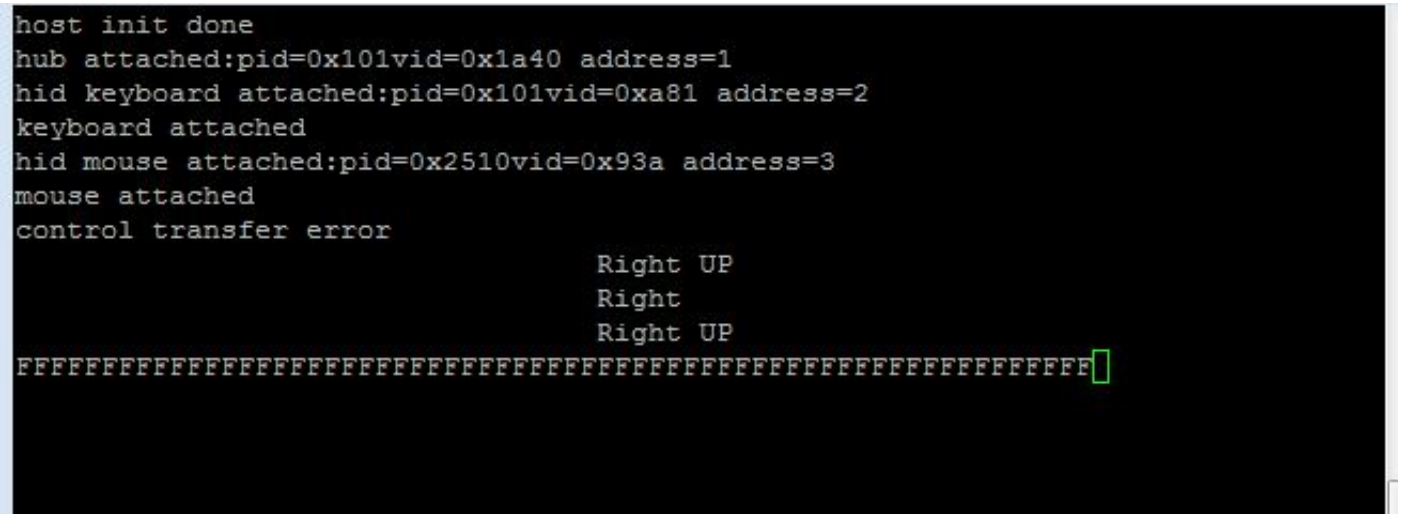
1. Connect board uart to PC and open the COM port in a terminal tool.
2. plug in hub, mouse or keyboard deivce to the board, the attach information print out in the terminal.
3. if one mouse is pluged, the mouse operation information will print in the terminal when you operate the mouse.
   Application print mouse operation informations in one line. Each line contain the following sequential string: "Left Click", "Middle Click", "Right Click", "Right"/"Left" movement, "UP"/"Down" movement and "Wheel Down"/"Wheel Up" movement. Whitespace will replace the above string if mouse don't have the corresponding operation.

   ```
   for example: when mouse move right and up, "
      Right UP             " print in the terminal.
   ```

4. if one keyboard is pluged, the keyboard pressed key information will print in the terminal when you operate the keyboard.

   ```
   for example: when F key is pressed, the 'F' print in the terminal.
   ```

The follow picture is an example for attaching one hub, one mouse and one keyboard.



Note

Host keyboard don't support long press key. If you want to implement the long press function, you can do as follow:

```
1.  Implement one timer and the interval is 10ms, for examle PIT.
2.  Define one 6 bytes' array for counting time.
3.  Update time array:
    (1) When timer time out: For every key in the lastPressData, add the ti
me. For example: if the second key is valid in lastPressData, the second va
lue of time array add 1.
    (2) When lastPressData are updated: Update the time array in order to m
```

ap the time with the lastPressData. For example: the second value of time a
rray is the time for the second of lastPressData.
4. When one time is greater than 5, the key is long pressed. Then print th
e key and reset the time value for next long press.