## **Overview**

The host CDC project is a simple demonstration program that uses the KSDK software. It enumerates a COM port and echoes back the data from the UART as follows:

- 1. The host CDC receives data from the UART, which is plugged in the PC. Type characters in the terminal tool, such as Tera Term, and the characters are sent to the host example.
- 2. After the host example receives data from the UART, it sends the data to the device virtual COM.
- 3. After the data has been sent to the device virtual COM successfully, it receives the same data from the device virtual COM.
- 4. If the host has received the data, it sends data to the UART using the UART driver API. The UART echoes back data to the PC.

## **System Requirement**

#### Hardware requirements

- Mini/micro USB cable
- USB A to micro AB cable
- Hardware (Tower module/base board, and so on) for a specific device
- Personal Computer(PC)

### Software requirements

• The project path is:

```
<SDK_Install>/boards/<board>/usb_examples/usb_host_cdc/<rtos>/<toolchain>.
```

Note

The <rtos> is Bare Metal or FreeRTOS OS.

# **Getting Started**

### **Hardware Settings**

• The Jumper settings:

J28 1-2, J22 1-2.

Note

Set the hardware jumpers (Tower system/base module) to default settings.

### Prepare the example

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

Note

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

# Run the example

- 1. Run the host\_cdc\_serial example. The printed guide is displayed. Follow the guide to modify the demo.
- 2. Plug in the CDC device. The attached information is printed out.
- 3. Type a string and the string is sent to the CDC device if the string length is greater than USB\_HOST\_SEND\_RECV\_PER\_TIME. If the length is shorter than the USB\_HOST\_SEND\_RECV\_PER\_TIME, the string is echoed back later. After that, the CDC host reads back the string and puts it to stdout.

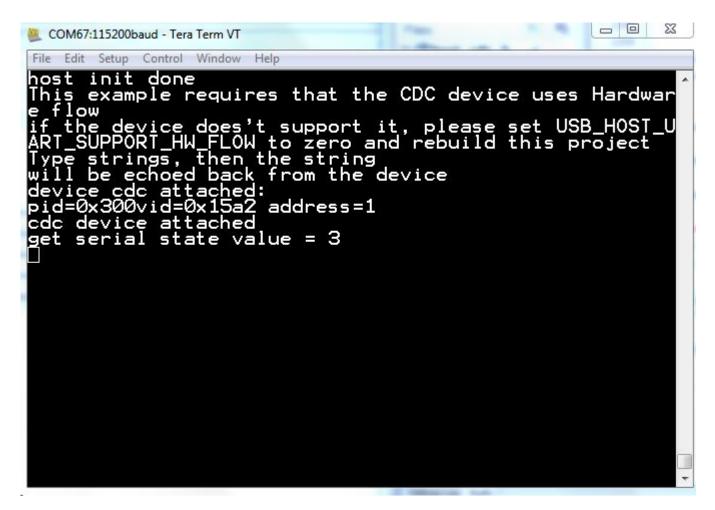


Figure 1: Host cdc output