

XMC4500 USB Host Virtual COM Port (VCOM) Example

Getting Started V1.0



Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

7

Further reading

8

Limitations

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

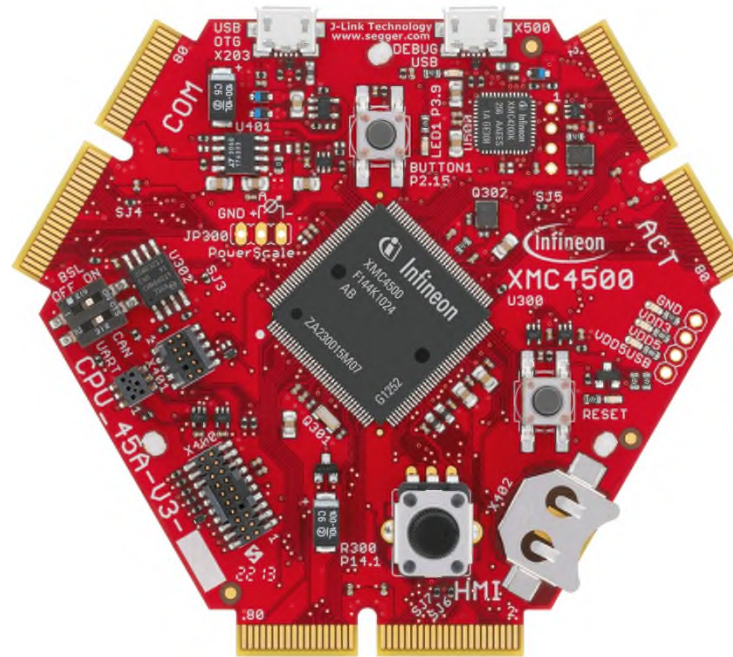
Debug hints

7

Further reading

8

Limitations



- › This example demonstrates the implementation of USB host virtual COM (VCOM) port functionality. It is based on the free LUFA USB host stack implementation which is ported to XMC4500 family and provided within this example.
- › Within this documentation you will be guided through all the building blocks of a typical USB host VCOM application on the XMC4500 family. You will be able to connect a VCOM device to the host and perform data transfers access to and from the device.
- › As a result you will be enabled to implement your own USB host VCOM functionality on the XMC4500 family.

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

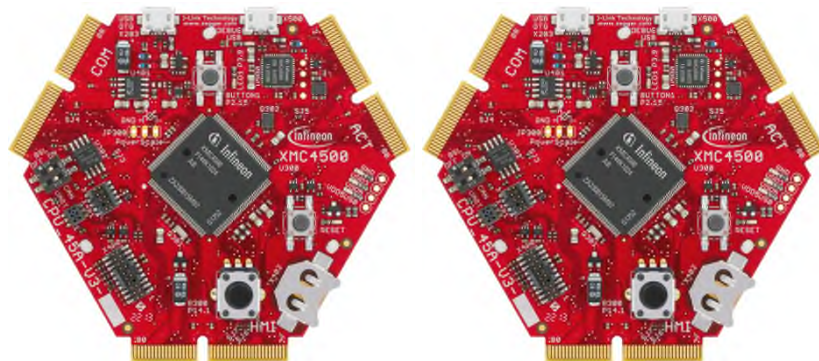
7

Further reading

8

Limitations

Requirements - hardware



Two XMC4500 General Purpose CPU_45A-V3 boards

One will behave as a USB host and another will behave as a USB device



USB cable micro-A plug to micro-B plug for host example



USB cable micro-B plug to standard-A plug for debugging and serial port terminal

Requirements – hardware and software



- › Windows laptop
- › DAVE™ installed



- › DAVE™ (v4.1.4 or Higher)
- › Download DAVE™ free of charge
Link : [DAVE™ 4 Download](#)
- › Download USB device VCOM app from DAVE™
- › Serial port terminal software like Tera Term
Link : <http://ttssh2.osdn.jp/>

Requirements – free software download



- › DAVE™ (v4.1.4 or Higher)
- › Download DAVE™
Link : [DAVE™ 4 Download](#)
- › Download USB device VCOM app from DAVE™
- › Serial port terminal software like Tera Term
Link : <http://ttssh2.osdn.jp/>

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

7

Further reading

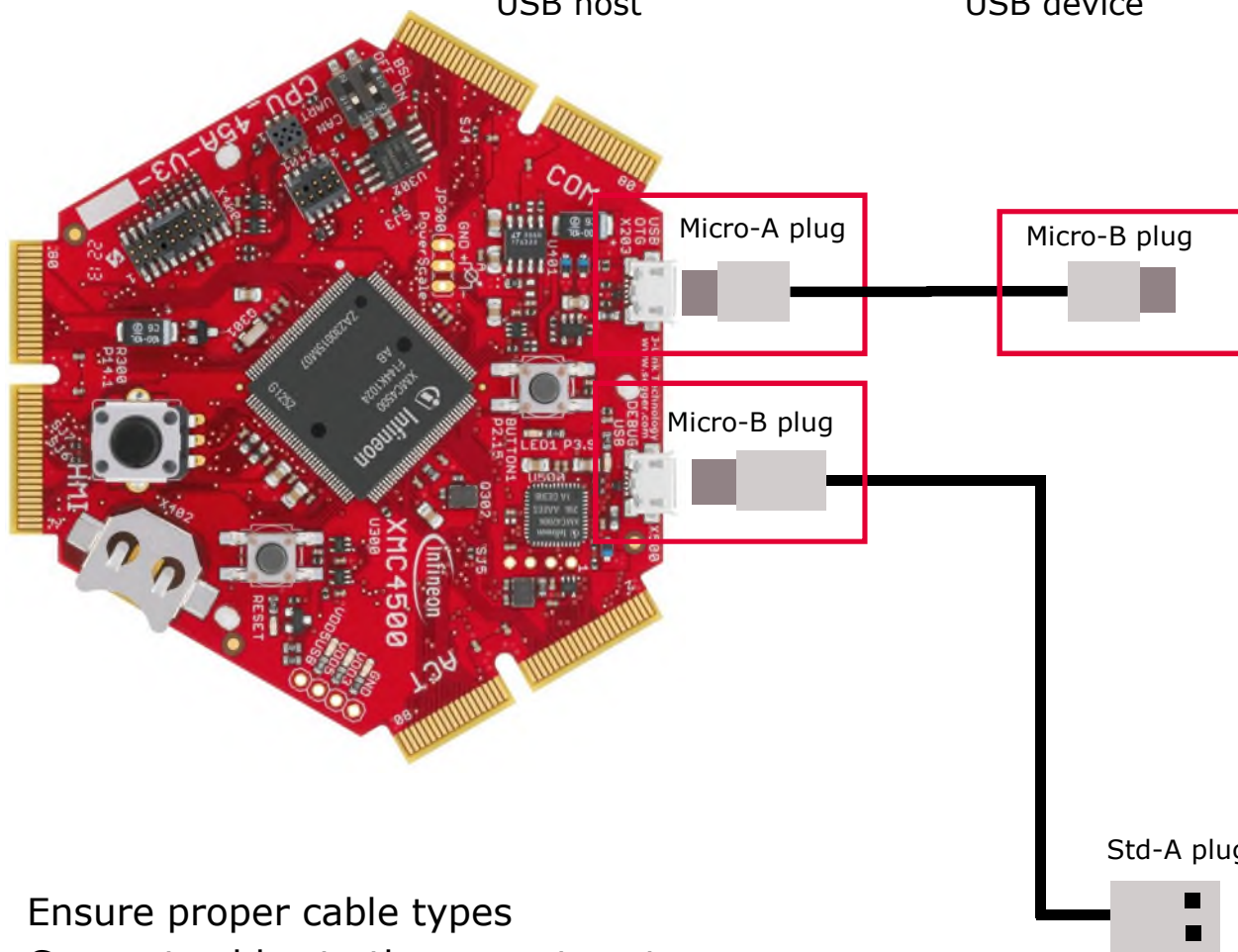
8

Limitations

Setup - hardware

XMC4500
USB host

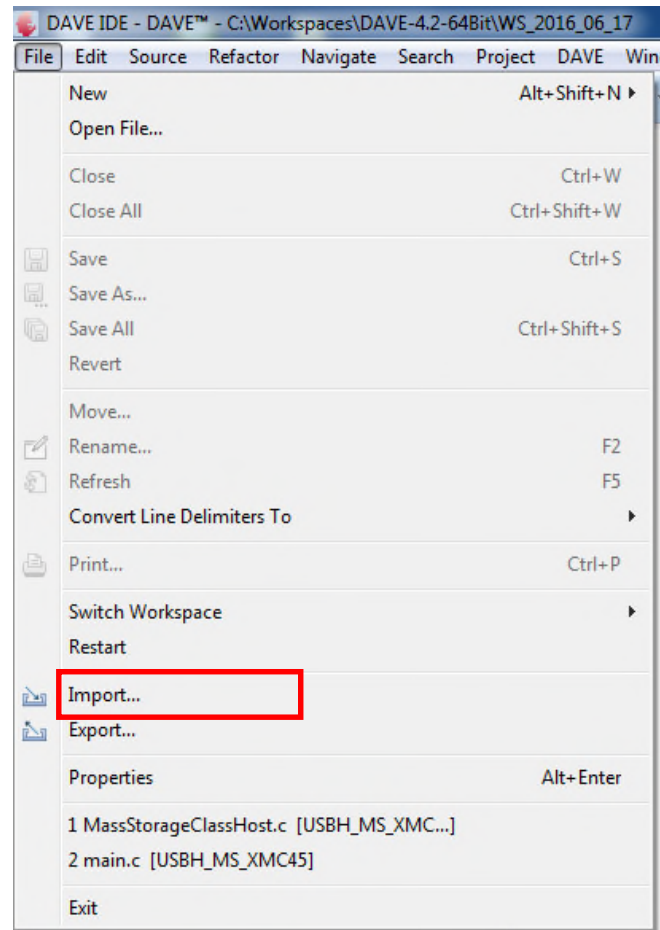
XMC4500
USB device



Windows laptop running
DAVE™ and serial
communication terminal

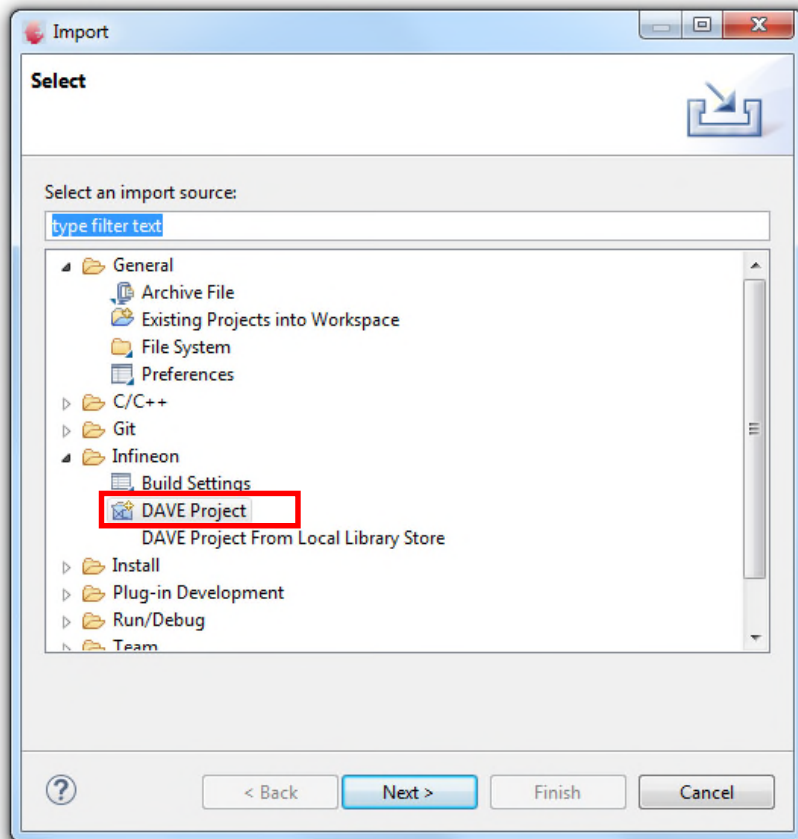
- › Ensure proper cable types
- › Connect cables to the correct ports

Setup – import VCOM example project in to DAVE™



1

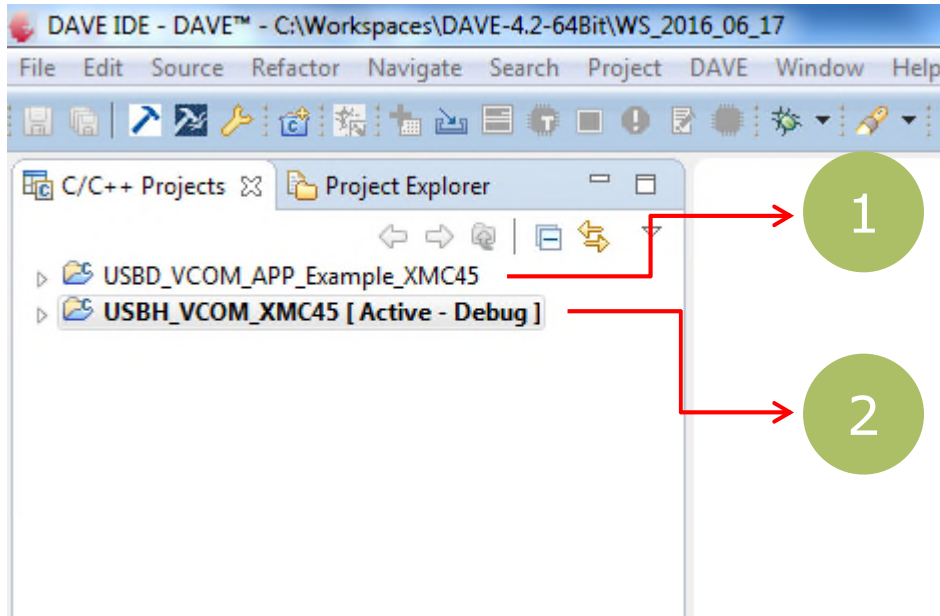
Setup – import VCOM example project in to DAVE™



2

3

Setup –import VCOM example project in to DAVE™



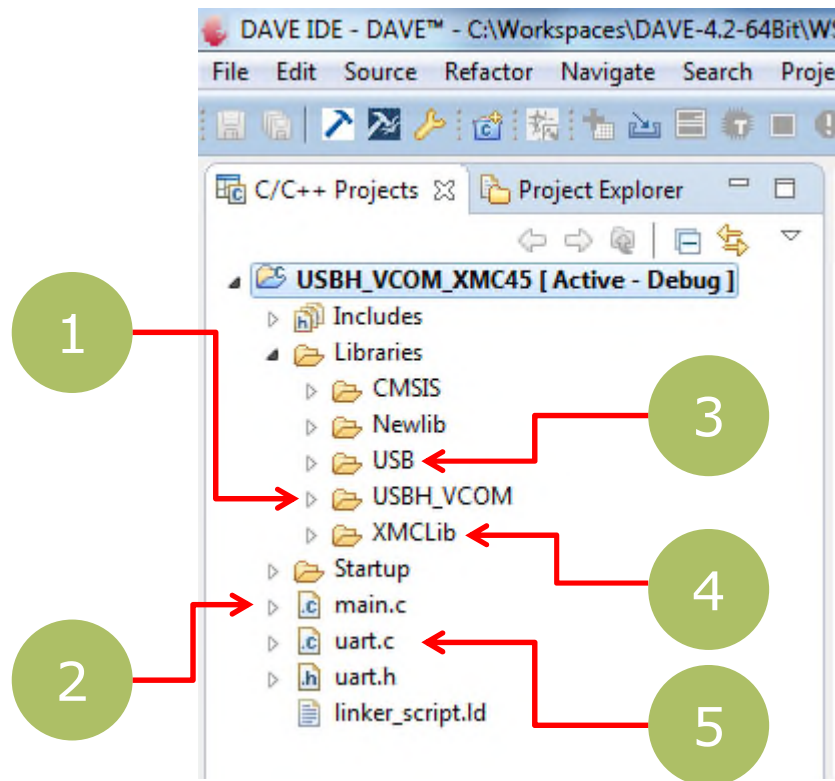
1

USB VCOM Device project

2

USB VCOM Host project

Setup – import VCOM example project in to DAVE™



› Check the folder structure of the imported project.

1

VCOM application code implementing USB data transmit and receive

2

The main file implementing the application task and echo data on UART

3

Free LUFA USB stack, CDC-ACM class and glue layer for XMC low level driver

4

XMCLib folder contains USB low level driver in file xmc_usb.c

5

Code for redirecting data to UART

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

7

Further reading

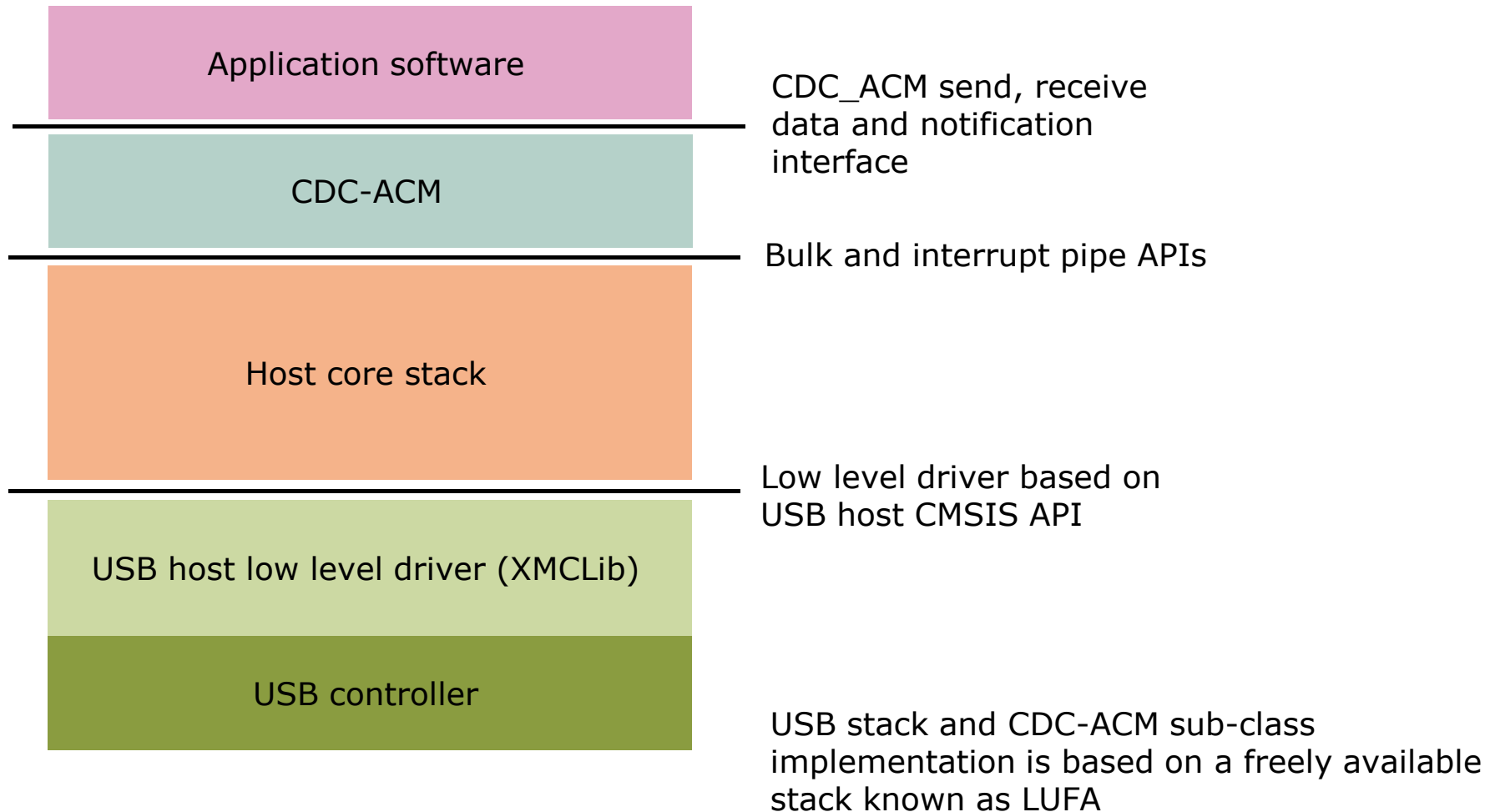
8

Limitations

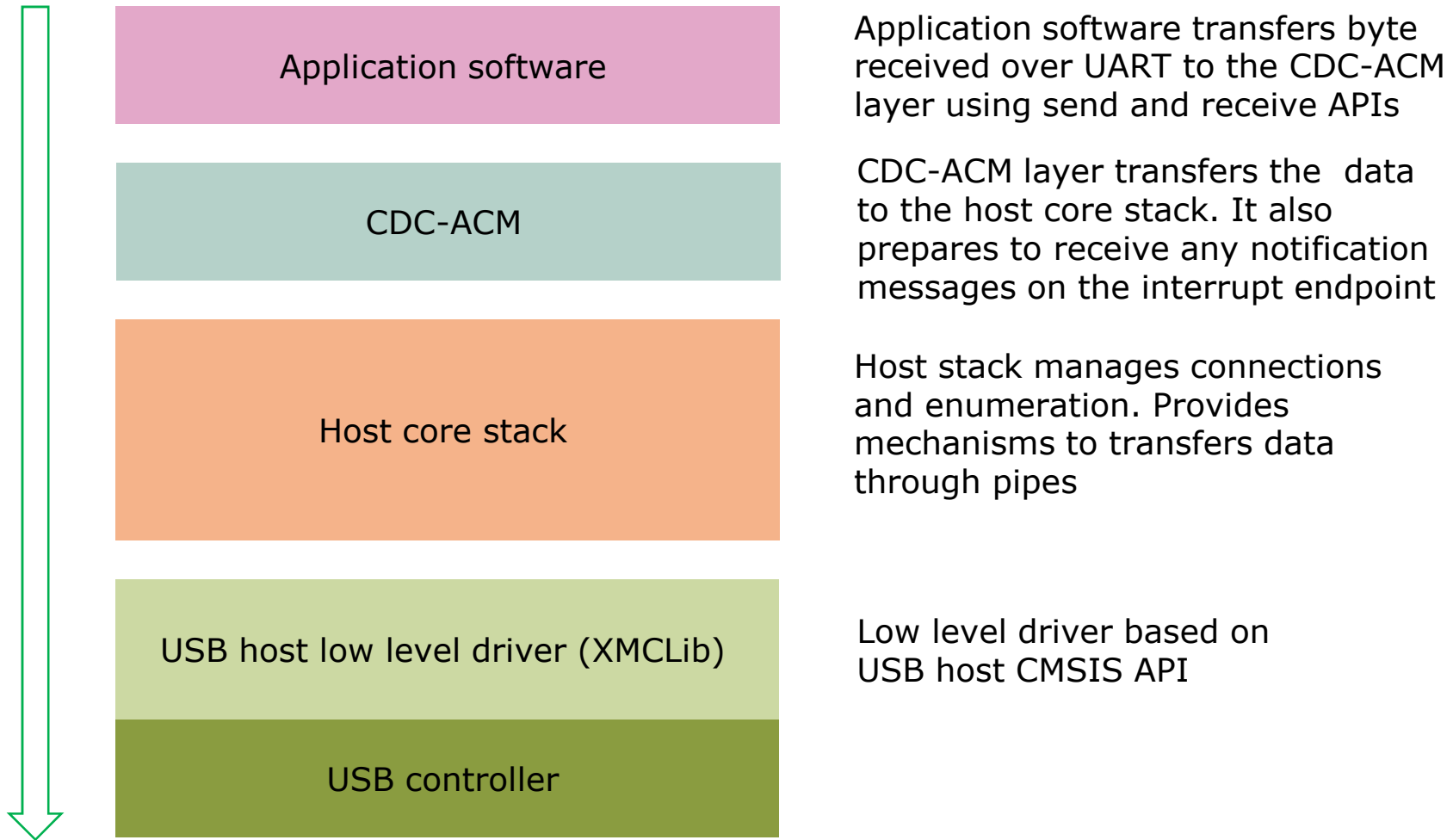
Example application behavior

- › First XMC4500 CPU_45A-V3 board is initialized as a USB host and provides VBUS on the USB port. This board runs host VCOM software.
- › Second XMC4500 CPU_45A-V3 board is initialized as USB device. This board runs VCOM device software.
- › When successfully enumerated, USB host recognizes the connected device as a CDC-ACM capable device (VCOM)
- › First XMC4500 as USB host receives data from Windows laptop running serial terminal software like Tera Term and sends data to the device XMC4500 over USB
- › Second XMC4500 as USB device loops back the received data on the USB
- › First XMC4500 as USB host redirects received loop back data to debug port connected to Windows laptop running serial terminal software like Tera Term

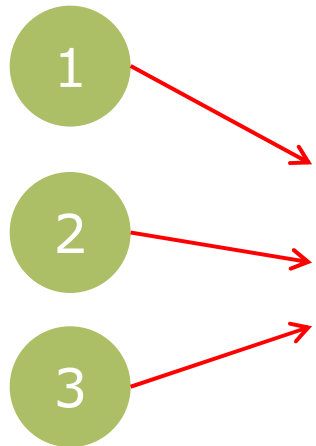
Simplified example application architecture



Example application data flow



Application – overview of main.c

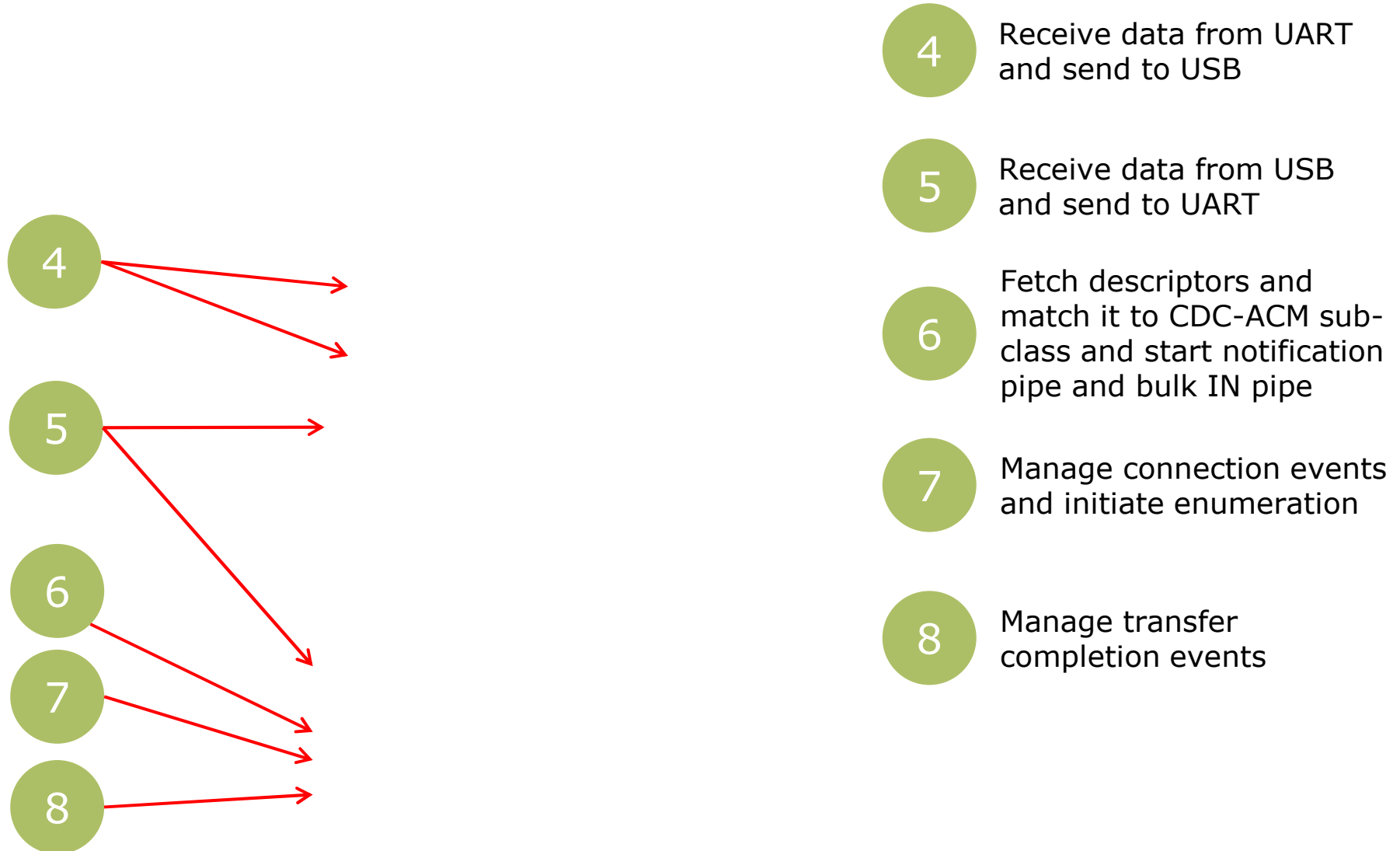


1 Clock setup for XMC4500

2 Select VBUS pin P3.2

3 Initialize USB host

Application – overview of main.c



Application – configure time delay API

- › The USB stack uses timing delay of the order of several tens of milliseconds to comply with the specification
- › Port the time delay API if required. In the example code, SysTick timer is used
- › Ensure timer is calibrated

Application - callbacks

- › Application registers for the following event callbacks



Application – Implement and register required callbacks

1

Implement call back function

1

```
157  /*Callback function executed on port interrupt*/
158  void CDC_USB_PortCb(uint8_t port, uint32_t event)
159  {
160      if(event & XMC_USBH_EVENT_DISCONNECT)
161      {
162          XMC_GPIO_SetOutputHigh(XMC_GPIO_PORT3, 9);
163          Driver_USBH0.PipeDelete(USBHost_Pipe_State[0].pipe_handle);
164          Driver_USBH0.PipeDelete(USBHost_Pipe_State[1].pipe_handle);
165          Driver_USBH0.PipeDelete(USBHost_Pipe_State[2].pipe_handle);
166          Driver_USBH0.PipeDelete(USBHost_Pipe_State[3].pipe_handle);
167          /*Reset the data handling indices*/
168          USBH_VCOM_RX_cur_index = 0;
169          USBH_VCOM_RX_prev_index = 0;
170      }
171      if(event & XMC_USBH_EVENT_REMOTE_WAKEUP)
172      {
173          /*This flag is set to remember the occurrence of remote wakeup event and
174          * to return from ISR immediately. This helps to time a 20ms delay in
175          * the context of the application main loop rather than inside the ISR context.
176          * The reason for this is that the example uses a timer interrupt whose priority is
177          * lower than the USB interrupt and therefore would result in a deadlock.*/
178          USBH_RemoteWkUp_Detected = 1;
179      }
180  }
```

Application – Implement and register required callbacks

2

```
31 /*Callback functions to be called from USB glue layer*/
32 USBH_GLUE_APP_IF_t USBVCH_CB =
33 {
34     .GetBytesInPipe = USBVCH_Pipe_BytesInPipe,
35     .GetReadByte = USBVCH_Pipe_Read_8,
36     .IsINReceived = USBVCH_Pipe_IsINReceived,
37     .PipeEventHandler = USBH_VCOM_Rx_Data_Handler,
38     .PortEventHandler = CDC_USB_PortCb
39 };
```

2

Register call back function

3

Call initialization function

3

```
49 int main(void)
50 {
51     uint8_t USBHostState;
52     int8_t ReceivedByte;
53     uint8_t *RxPtr;
54     uint32_t Rxlen;
55     uint32_t uart_data_count;
56
57     /*Configure USB and CPU clocks*/
58     ClockSetup();
59     /*Initialize UART channel*/
60     UART_init();
61     /*Select VBUS pin as P3.2*/
62     XMC_USBH_Select_VBUS(XMC_GPIO_PORT3, 2U);
63     /* Initializes the USB host driver. */
64     USB_Init_Host(&USBVCH_CB);
65
66 }
```

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

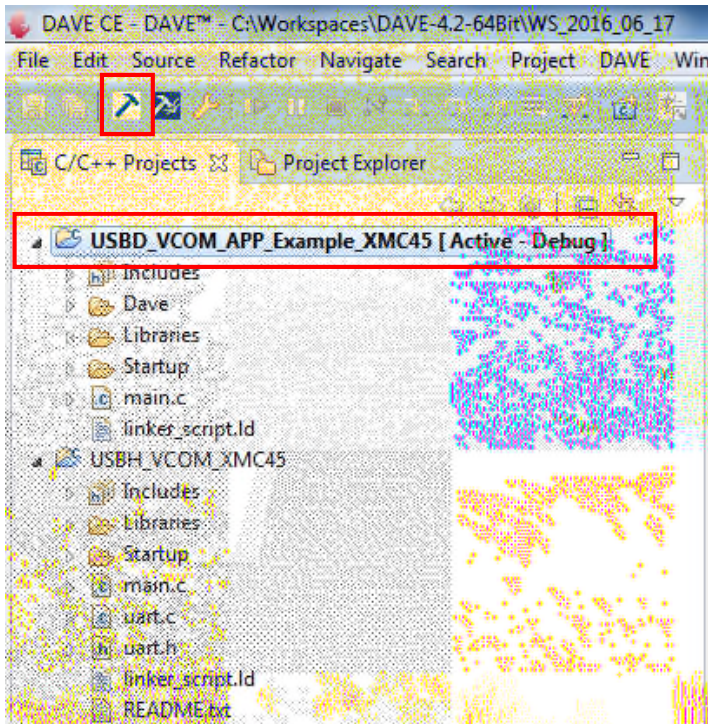
7

Further reading

8

Limitations

How to test – build and download device project



1

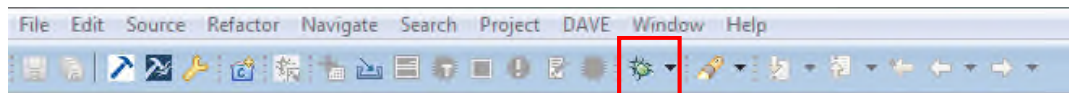
1 Make the device project 'Active' and build **device project**

2

2 Download image into **second** XMC4500 General Purpose board. Image now resides in flash memory

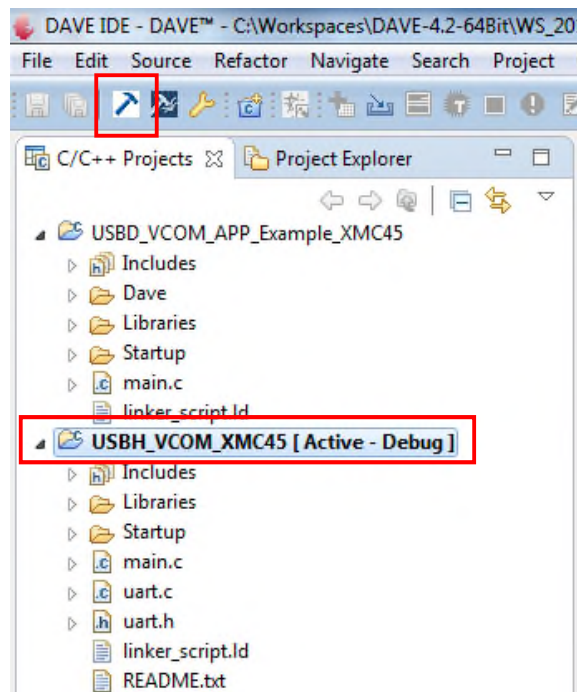
3

3 Reset XMC4500 General Purpose board to start executing the image



2

How to test – build and download host project



1

1

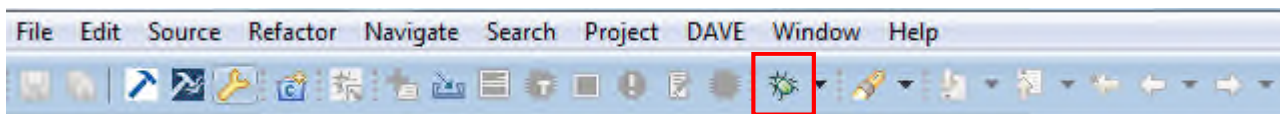
Make the host project 'Active', build and download **host project** into **first** General Purpose board.

2

Start debugger

3

Start the execution by pressing the run button



2



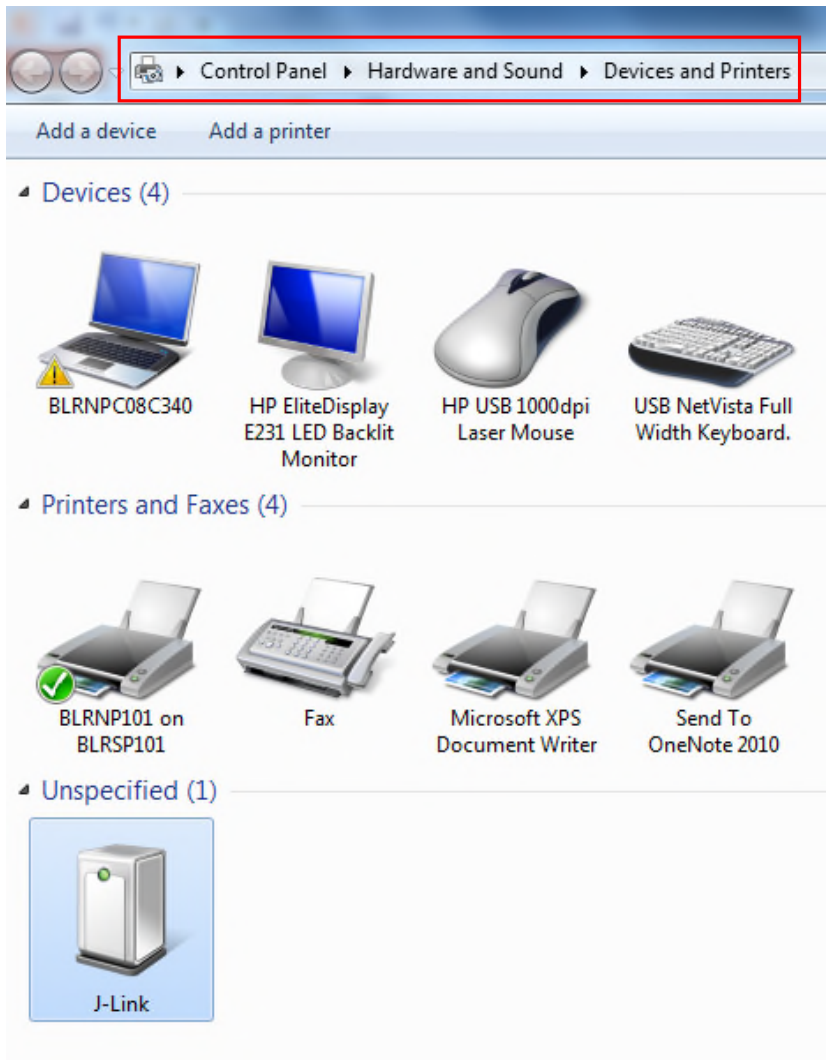
3

How to test

- 1 Connect the USB cable between the host and device XMC4500 General Purpose boards. Refer slide 10
- 2 Open serial port terminal software on Windows laptop (example Tera Term)
 - See next slide on how to locate associated COM port
- 3 Configure the terminal software for 19200 baud rate, 8 bit data, 1 stop bit, no parity
- 4 Type some characters. The typed characters will be looped back and visible on the terminal.

Finding associated COM port on Windows

1

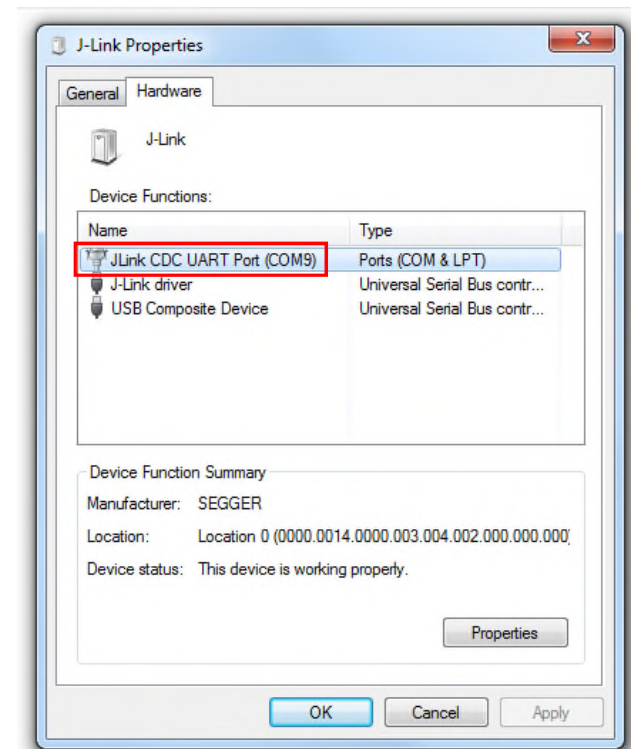


1

Open Control Panel -> Hardware and Sound -> Devices and Printers

2

Right click on J-Link and select Properties, then select Hardware tab. COM port is indicated.



Supported features

- › ACM Requests – SetControlLineState, SetLineCoding, SendBreak
- › Only ACM Notification - Serial State is implemented

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

7

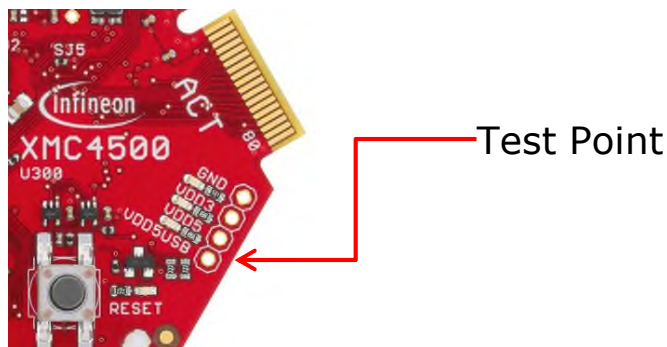
Further reading

8

Limitations

Debug Hints

- › Check VBUS is available and within permissible range (4.4V to 5.25V) on the USB host port receptacle. Lower VBUS voltages due to excessive current draw may cause USB devices to disconnect intermittently or fail enumeration. Try connecting a different USB device.



- › Long cables may cause malfunction due to signal quality issues. Use short cables if possible.
- › Check if XMC4500 VCOM device example application is running properly by connecting device to a Windows host running Tera Term. XMC4500 device will loop back characters typed on terminal.

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

7

Further reading

8

Limitations

Further reading

- › USB specifications and Micro-USB cables and connectors specification

http://www.usb.org/developers/docs/usb20_docs/

- › Communication devices class specifications
 - Class definitions for communication devices v1.2
 - Communications class, subclass specifications for PSTN devices

http://www.usb.org/developers/docs/devclass_docs/

- › CMSIS USB host API specification

https://www.keil.com/pack/doc/CMSIS/Driver/html/group_usb_interface_gr.html

Further reading

- › LUFA USB stack

<http://www.fourwalledcubicle.com/index.php>

- › Books

- USB Complete: The Developer's Guide, Jan Axelson
- USB Embedded Hosts, Jan Axelson

Contents

1

Overview

2

Requirements

3

Setup

4

Implementing the application

5

How to test

6

Debug hints

7

Further reading

8

Limitations

Limitations

- › Hubs are not supported
- › Over current protection is not supported
- › DMA Transfers are not supported
- › Low level driver does not support isochronous transfers



Part of your life. Part of tomorrow.

