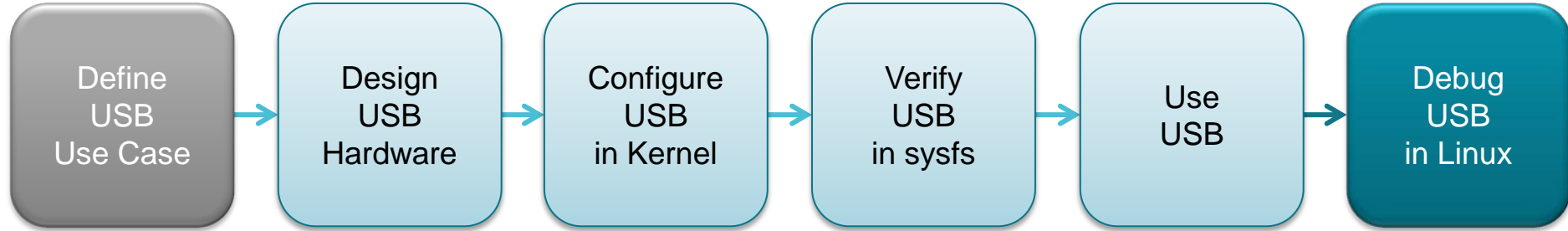


USB System Design in Sitara Devices Using Linux

[Part 7]: Debug USB in Linux

Bin Liu (EP, Processors)

Agenda



- Two categories of USB software issues
 - USB software initialization failure:
 - Linux misconfiguration
 - USB is not functional as expected:
 - software bugs
 - (Assuming hardware designed correctly)
- Support on TI E2E Forums

Guidelines on debugging USB misconfiguration

- Observing the failure:
 - Host mode
`lsusb`
 - Device mode
`/sys/class/udc/*/uevent`
- Diagnosis:
 - Kernel boot log
 - Kernel Config
 - `lsmod`
 - Device Tree
 - `chkusb.sh`

Observe misconfiguration failure in host mode

- Use “**lsusb**” command.
- In working case:

```
root@am57xx-evm:~# lsusb
```

```
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation
```

- In failure case:

```
root@am57xx-evm:~# lsusb
```

```
unable to initialize libusb: -99
```

Observe misconfiguration failure in device mode

- Check `/sys/class/udc/` folder.

- In working case:

```
root@am57xx-evm:~# ls /sys/class/udc
```

```
48890000.usb
```

```
root@am335x-evm:~# ls /sys/class/udc
```

```
musb-hdrc.0
```

- In failure case:

```
root@am57xx-evm:~# ls /sys/class/udc
```

```
<empty>
```

Diagnosis 1: Check kernel boot log

- Check kernel *dmesg* log for any USB error message:
USB PHY, USB controller, Glue driver, USB Core, DMA, Extcon, etc.

~~dmesg | grep usb~~

- Some USB related messages do not have the keyword “usb”

Diagnosis 1: Check kernel configuration

- Ensure all required config options are enabled.
- For MUSB, refer to:
 - The previous module in this training: **MUSB Config in Linux Kernel**
 - Wiki: MUSB Linux Driver Configuration
http://processors.wiki.ti.com/index.php/MUSB_Linux_Driver_Configuration
- For DWC3, refer to:
 - The previous module in this training: **DWC3 Config in Linux Kernel**
 - Wiki: DWC3 Linux Driver Configuration
http://processors.wiki.ti.com/index.php/Linux_Core_DWC3_User's_Guide#Driver_Configuration
- Use the default config provided in Processor Linux SDK kernel as the reference.
`tisdk_<xxx-evm>_defconfig`

Diagnosis 1: Check USB kernel modules

- Use 'lsmod' command to examine loaded USB kernel modules
- If any modules are missing, check to see if the *.ko module exists in the filesystem under
`/lib/modules/$(uname -r)/kernel/drivers/`

Diagnosis 1: Check device-tree settings

- Ensure USB-related DT nodes are enabled and *dr_mode* is set properly.
- Double check Extcon DT nodes with design schematics.
- For MUSB, refer to:
 - The previous module in this training: **MUSB Config in Linux Kernel**
 - Wiki AM335 MUSB Linux Porting Guide
http://processors.wiki.ti.com/index.php/MUSB_Linux_Porting_Guide
 - Wiki: MUSB DT Bindings Kernel Documentation
 - <https://www.kernel.org/doc/Documentation/devicetree/bindings/usb/am33xx-usb.txt>
 - <https://www.kernel.org/doc/Documentation/devicetree/bindings/usb/da8xx-usb.txt>
- For DWC3, refer to:
 - The previous module in this training: **DWC3 Config in Linux Kernel**
 - Wiki: OMAP DWC3 DT Bindings Kernel Documentation
<https://www.kernel.org/doc/Documentation/devicetree/bindings/usb/omap-usb.txt>
 - Wiki: DWC3 Core DT Bindings Kernel Documentation
<https://www.kernel.org/doc/Documentation/devicetree/bindings/usb/dwc3.txt>
- Examine `/sys/firmware/devicetree/` to ensure DTS changes take effect.

Diagnosis 1: *chkusb.sh* script

- The script does most of the checking mentioned previously
- The script is not provided in the Processor SDK filesystem
- How to find it?
 - It is attached in many TI E2E forum threads
 - Search for “chkusb.sh” on <https://e2e.ti.com>

Guidelines on debugging USB runtime failures

- Check hardware design
- USB driver logs
- Dynamic debug
- ftrace (function tracer)
- USB register dump
- USB protocol analyzer trace

Diagnosis 2: Check USB hardware design

- Refer to the previous module in this training: **USB Hardware Design**
- Review schematics
- Review layout
- USB EYE Diagram test

Diagnosis 2: Check kernel logs

- Kernel USB drivers print error messages when USB behaves abnormally:
 - Error messages
 - Kernel crash dump
- These messages might be the first hint of an issue.

Diagnosis 2: Dynamic debug

- To get more kernel debug logs, run dynamic debug:
 - Enable/disable kernel debug logs at Linux runtime
 - More details:
<https://training.ti.com/debugging-embedded-linux-dynamic-debug>
- Suitable for USB core, class/gadget drivers

Diagnosis 2: ftrace log

- Kernel internal tracer is designed to help understand what is going on inside the kernel.
- More details: <https://www.kernel.org/doc/Documentation/trace/ftrace.txt>
- MUSB/DWC3/xHCI controllers drivers use ftrace **event tracing**:
 - To expose driver/controller activities at runtime
 - More about event tracing:
<https://www.kernel.org/doc/Documentation/trace/events.txt>

Diagnosis 2: ftrace log in MUSB

```
root@am335x-evm:~# ls /sys/kernel/debug/tracing/events/musb
enable                musb_readb           musb_urb_deq
filter                musb_readl           musb_urb_enq
musb_cppi41_abort     musb_readw           musb_urb_gb
musb_cppi41_alloc     musb_req_alloc       musb_urb_rx
musb_cppi41_config    musb_req_deq         musb_urb_start
musb_cppi41_cont      musb_req_enq         musb_urb_tx
musb_cppi41_done      musb_req_free        musb_writeb
musb_cppi41_free      musb_req_gb          musb_writel
musb_cppi41_gb        musb_req_rx          musb_writew
musb_isr              musb_req_start
musb_log              musb_req_tx
```


Diagnosis 2: ftrace log in DWC3

```
root@am57xx-evm:~# ls /sys/kernel/debug/tracing/events/dwc3/  
dwc3_alloc_request      dwc3_ep_queue           dwc3_gadget_giveback  
dwc3_complete_trb      dwc3_event              dwc3_prepare_trb  
dwc3_core               dwc3_free_request       dwc3_readl  
dwc3_ctrl_req           dwc3_gadget              dwc3_writel  
dwc3_ep0                 dwc3_gadget_ep_cmd      enable  
dwc3_ep_dequeue         dwc3_gadget_generic_cmd filter
```

Diagnosis 2: ftrace log in xHCI

```
root@am57xx-evm:~# ls /sys/kernel/debug/tracing/events/xhci-hcd/  
enable                xhci_dbg_context_change  
filter                xhci_dbg_init  
xhci_address_ctx     xhci_dbg_quirks  
xhci_cmd_completion  xhci_dbg_reset_ep  
xhci_dbg_address     xhci_dbg_ring_expansion  
xhci_dbg_cancel_urb
```

Diagnosis 2: Dump USB module registers

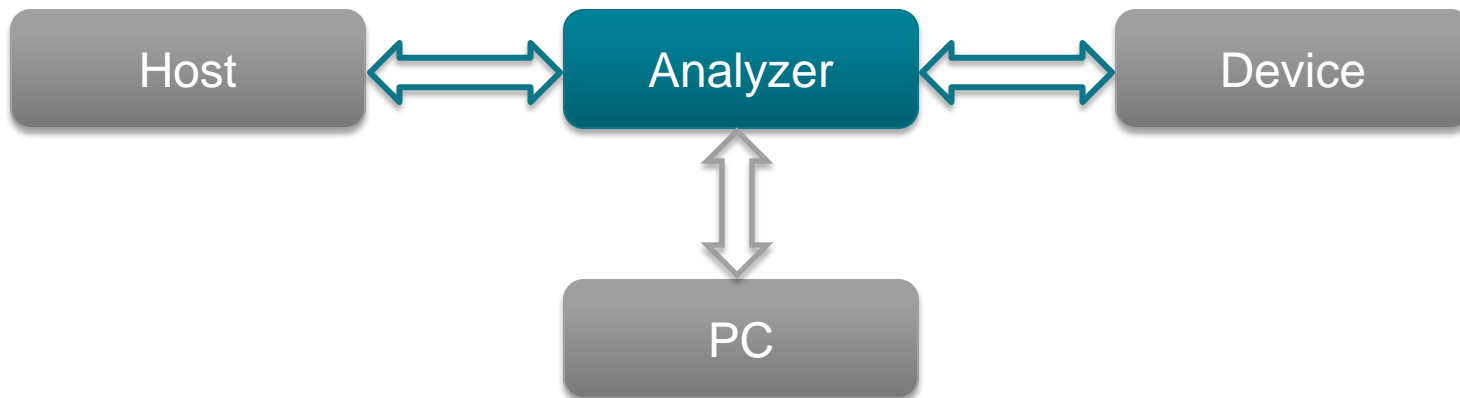
- For MUSB controller:
cat /sys/kernel/debug/**musb-hdrc.0**/regdump
- For DWC3 controller:
cat /sys/kernel/debug/**48890000.usb**/regdump
- xHCI controller driver does not have an debugfs entry to dump xHCI registers.
Use devmem2 or other tool to read registers.

Note: Replace the node in **red** above accordingly.

Diagnosis 2: USB protocol analyzer trace

Hardware-based USB protocol analyzer:

- USB packet sniffer
- Independent of USB host or device



Seeking support on TI E2E Forums

- Post your queries on Sitara Processors Forum:
http://e2e.ti.com/support/arm/sitara_arm/f/791
- Including the keyword “**USB**” in the post title.
- Watch the video *Dos & Don'ts for Communication to TI E2E Forums*:
<https://training.ti.com/dos-donts-posting-to-e2e-community-forums>
Provides tips for efficient E2E communication

For more information

- Debugging Embedded Linux Systems Training Series:
<https://training.ti.com/debugging-embedded-linux-dynamic-debug>
- USB System Design in Sitara Devices Using Linux Training Series:
<https://training.ti.com/usb-system-design-sitara-linux>
- MUSB Linux Driver Configuration
http://processors.wiki.ti.com/index.php/MUSB_Linux_Driver_Configuration
- DWC3 Linux Driver Configuration
http://processors.wiki.ti.com/index.php/Linux_Core_DWC3_User's_Guide#Driver_Configuration
- For questions about this training, refer to the E2E Community Forums at
<http://e2e.ti.com>



© Copyright 2018 Texas Instruments Incorporated. All rights reserved.

This material is provided strictly “as-is,” for informational purposes only, and without any warranty.
Use of this material is subject to TI’s **Terms of Use**, viewable at [TI.com](https://www.ti.com)