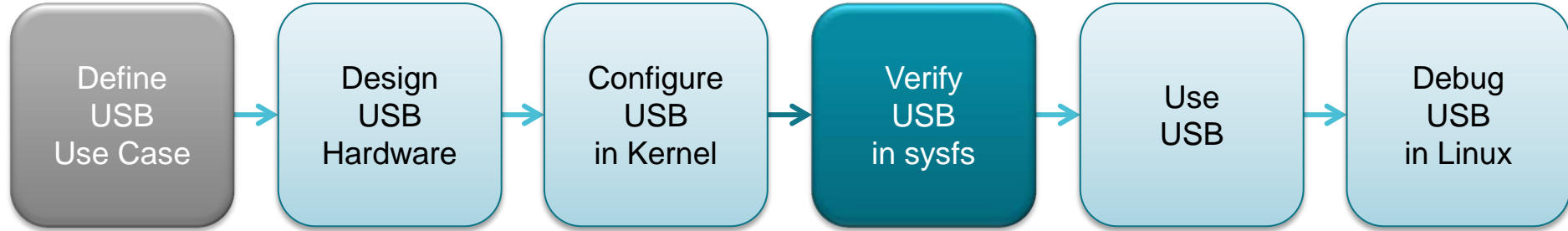


USB System Design in Sitara Devices Using Linux

[Part 4]: Verify USB in Linux sysfs

Bin Liu (EP, Processors)

Agenda



- sysfs overview
- Device and driver structure in sysfs
- USB in sysfs
 - MUSB
 - DWC3

sysfs overview

- A RAM-based virtual filesystem in Linux
- Used to export kernel internals to userspace:
 - Kernel data structure
 - Directories
 - Attributes
 - Regular files
 - Relationships
 - Symbolic links

sysfs overview

- A RAM-based virtual filesystem in Linux
- Used to export kernel internals to userspace:
 - Kernel data structure
 - Directories
 - Attributes
 - Regular files
 - Relationships
 - Symbolic links

```
root@am335x-evm:~# cd /sys/devices/soc0/  
root@am335x-evm:soc0# stat -c "%A %N" *  
drwxr-xr-x power  
-r--r--r-- family  
-r--r--r-- machine  
-r--r--r-- revision  
-r--r--r-- type  
-rw-r--r-- uevent  
lrwxrwxrwx 'subsystem' -> '../../bus/soc'
```

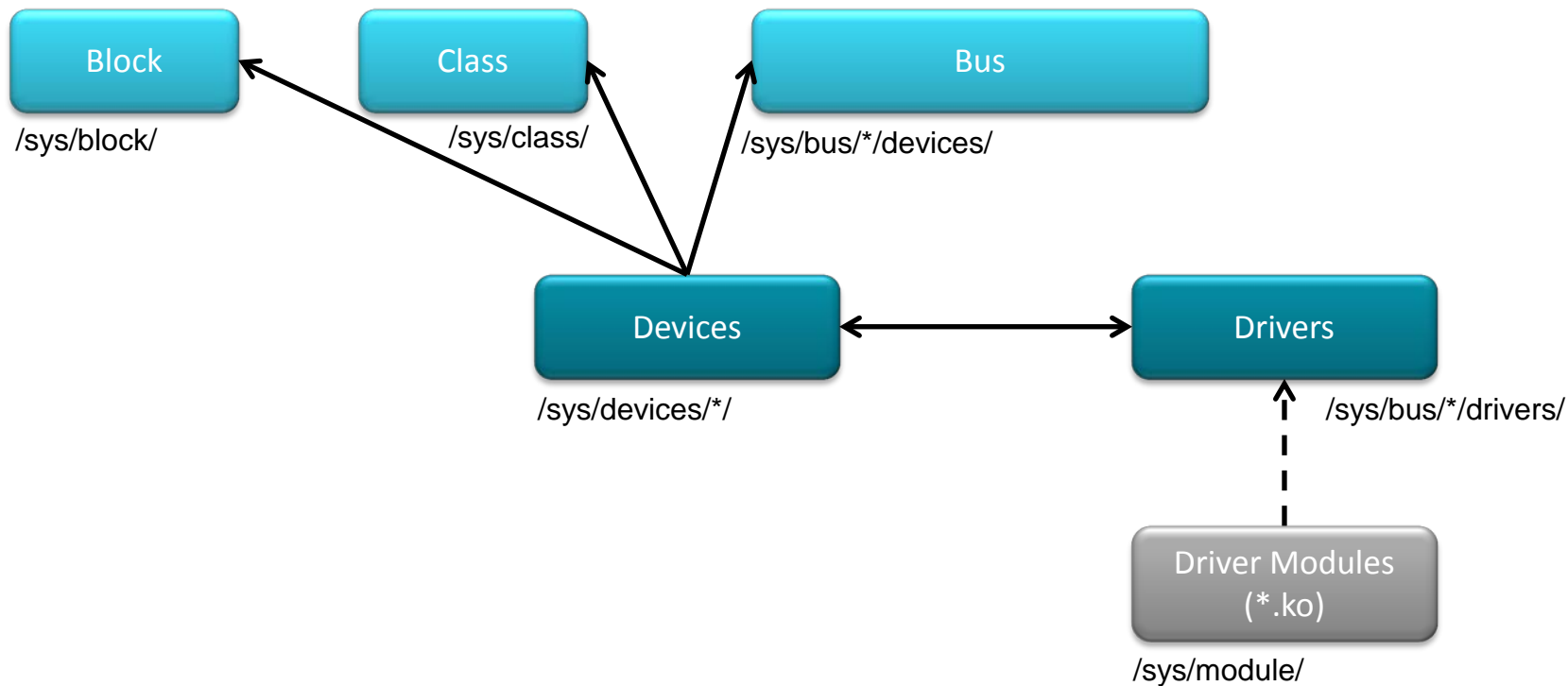
sysfs directories overview

(As in kernel v4.9; Directories are subject to change)

/sys/

- | -- **block** # *classification of devices, describe **block** devices*
- | -- **bus** # *classification of devices, describe **buses***
- | -- **class** # *classification of devices, describe **functional types** of devices*
- | -- **dev** # *describe **block/char devices** in device node format*
- | -- **devices** # *unified places describing **all devices** in a flat list*
- | -- **firmware** # ***device tree** information is here!*
- | -- **fs** # *describe filesystems*
- | -- **kernel** # */sys/kernel/debug/ !!!*
- | -- **module** # *describe all loaded ***.ko** modules*
- ` -- **power** # *“echo mem > /sys/power/state” !!!*

Devices & Drivers in sysfs



Example showing sysfs structure

Let's see how device *44e3e000.rtc* is referenced in sysfs.

It is the RTC module in AM335x:

```
root@am335x-evm:~# find /sys -name 44e3e000.rtc
```

Example showing sysfs structure

Let's see how device *44e3e000.rtc* is referenced in sysfs.

It is the RTC module in AM335x:

```
root@am335x-evm:~# find /sys -name 44e3e000.rtc
```

```
/sys/devices/platform/ocp/44e3e000.rtc
```

```
/sys/bus/platform/devices/44e3e000.rtc
```

```
/sys/bus/platform/drivers/omap_rtc/44e3e000.rtc
```

```
/sys/kernel/debug/pinctrl/44e3e000.rtc
```


Example showing sysfs structure

Let's see how device *44e3e000.rtc* is referenced in sysfs.

It is the RTC module in AM335x:

```
root@am335x-evm:~# find /sys -name 44e3e000.rtc
```

```
/sys/devices/platform/ocp/44e3e000.rtc
```

```
/sys/bus/platform/devices/44e3e000.rtc
```

```
/sys/bus/platform/drivers/omap_rtc/44e3e000.rtc
```

```
/sys/kernel/debug/pinctrl/44e3e000.rtc
```



The device

Example showing sysfs structure

Let's see how device `44e3e000.rtc` is referenced in sysfs.

It is the RTC module in AM335x:

```
root@am335x-evm:~# find /sys -name 44e3e000.rtc
```

```
/sys/devices/platform/ocp/44e3e000.rtc
```

```
/sys/bus/platform/devices/44e3e000.rtc
```

```
/sys/bus/platform/drivers/omap_rtc/44e3e000.rtc
```

```
/sys/kernel/debug/pinctrl/44e3e000.rtc
```

The device

References in
platform bus

Symbolic link to

Example showing sysfs structure

Let's see how device `44e3e000.rtc` is referenced in sysfs.

It is the RTC module in AM335x:

```
root@am335x-evm:~# find /sys -name 44e3e000.rtc
```

```
/sys/devices/platform/ocp/44e3e000.rtc
```

```
/sys/bus/platform/devices/44e3e000.rtc
```

```
/sys/bus/platform/drivers/omap_rtc/44e3e000.rtc
```

```
/sys/kernel/debug/pinctrl/44e3e000.rtc
```

The device

References in
platform bus

Symbolic link to

Bound to **omap_rtc** driver

Example showing sysfs structure

Let's see how device `44e3e000.rtc` is referenced in sysfs.

It is the RTC module in AM335x:

```
root@am335x-evm:~# find /sys -name 44e3e000.rtc
```

```
/sys/devices/platform/ocp/44e3e000.rtc
```

```
/sys/bus/platform/devices/44e3e000.rtc
```

```
/sys/bus/platform/drivers/omap_rtc/44e3e000.rtc
```

```
/sys/kernel/debug/pinctrl/44e3e000.rtc
```

The device

References in
platform bus

Symbolic link to

Bound to **omap_rtc** driver

In debugfs

sysfs device folder contents

- Each device folder has the following files/folders:
 - driver** # *symbolic link to the driver for this device*
 - of_node** # *symbolic link to the device tree node for this device*
 - power/** # *runtime power management entries*
 - subsystem** # *symbolic link to the owning subsystem in bus or class*
 - uevent** # *records of udev events*
- Optional files/folders:
 - <sub- devices>*
 - <device- or subsystem- specific attributes>*

sysfs driver folder contents

- Each driver folder has the following files:
 - `<devices>` *# symbolic link to the devices which are bond to this driver*
 - `bind` *# (write-only) entry to bind a device*
 - `unbind` *# (write-only) entry to unbind a device*
- Optional files/folders:
 - `module` *# symbolic link to the driver module (*.ko) in /sys/module/*

sysfs setup

- Compile kernel with CONFIG_SYSFS enabled
 - Enabled by default in Processor SDK Linux
- Mount sysfs into the root filesystem /sys
 - `mount -t sysfs sysfs /sys`
 - Already done in Processor SDK filesystems

USB in sysfs

```
/sys/  
| ① -- devices/platform/<ocp>/<usb> # usb devices in a flat list  
| ② -- bus/platform/devices/<usb> # platform usb device references (controller, phy, ...)  
| ③ -- bus/platform/drivers/<usb> # platform usb drivers (controller, phy, ...)  
| ④ -- bus/usb/devices/<usb> # non-platform usb device references (bus, usb devices ...)  
| ⑤ -- bus/usb/drivers/<usb> # non-platform usb drivers (hub, bus, class, ...)  
| ⑥ -- module/<usb> # usb kernel modules (*.ko) are here!  
| ⑦ -- class/udc/<usb>/uevent # to check usb gadget drivers  
` ⑧ -- /sys/kernel/debug/<usb> # usb debugfs entries
```


AM335x MUSB devices in sysfs

```
/sys/devices/platform/ocp/  
├-- 47400000.usb # usb subsystem  
│  ├── 44e10620.control # usb phy control module  
│  ├── 47400000.dma-controller # cppi41 dma controller  
│  ├── 47401300.usb-phy # usb0 phy device  
│  ├── 47401400.usb # usb0 platform glue device  
│  │   └-- musb-hdrc.0 # usb0 musb controller  
│  ├── 47401b00.usb-phy # usb1 phy device  
│  └-- 47401c00.usb # usb1 platform glue device  
│     └-- musb-hdrc.1 # usb1 musb controller
```

AM57x DWC3 devices in sysfs

```
/sys/devices/platform/
|-- 44000000.ocp/
|   |-- 48880000.omap_dwc3_1           # usb1 module
|   |   |-- 48890000.usb              # usb1 dwc3 controller
|   |   |   |-- xhci-hcd.0.auto      # usb1 xHCI controller
|   |-- 488c0000.omap_dwc3_2         # usb2 module
|   |   |-- 488d0000.usb              # usb2 dwc3 controller
|   |   |   |-- xhci-hcd.1.auto      # usb2 xHCI controller
|-- 4a080000.ocp2scp
|   |-- 4a084000.phy                 # usb1 usb2.0 phy
|   |-- 4a084400.phy                 # usb1 usb3.0 phy
|   |-- 4a085000.phy                 # usb2 usb2.0 phy
```

AM335x USB entries in /sys/bus/platform/

Devices in <i>devices</i> /folder	Device Name	Drivers in <i>drivers</i> /folder
47400000.usb	Virtual node for all devices below	am335x-usb-childs
47401400.usb	MUSB platform glue for USB0	musb-dsps
47401c00.usb	MUSB platform glue for USB1	
musb-hdrc.0	MUSB controller for USB0	musb-hdrc
musb-hdrc.1	MUSB controller for USB1	
47400000.dma-controller	CPPI41 dma controller	cppi41-dma-engine
47401300.usb-phy	PHY for USB0	usb_phy_generic
47401b00.usb-phy	PHY for USB1	am335x-phy-driver am335x-control-usb

AM57x USB entries in /sys/bus/platform/

Devices in <i>devices/</i> folder	Device Name	Drivers in <i>drivers/</i> folder
48880000.omap_dwc3_1	DWC3 platform glue for USB1	omap-dwc3
488c0000.omap_dwc3_2	DWC3 platform glue for USB2	
48890000.usb	DWC3 controller for USB1	dwc3
488d0000.usb	DWC3 controller for USB2	
xhci-hcd.0.auto	xHCI controller for USB1	xhci-hcd
4a084000.phy	usb2.0 phy for USB1	omap-usb2
4a085000.phy	usb2.0 phy for USB2	
4a084400.phy	usb3.0 phy for USB1	ti-pipe3

USB entries in `/sys/bus/usb/`

Devices in <i>devices/</i> folder	Device Name	Drivers in <i>drivers/</i> folder
usb1	USB1 bus	usb
usb2	USB2 bus	
1-0:1.0	USB1 bus roothub interface	hub
2-0:1.0	USB2 bus roothub interface	

- The information of enumerated usb devices are listed here.
- When new usb devices are enumerated, more device entries are created under ***devices/***
 - In the folder naming convention of ***bus-port.port.port...:config.interface***
- The information is presented in *lsusb* command:
 - `lsusb -t`
 - `lsusb -v -d <vid:pid>`

/sys/class/udc/<usb>/uevent

- Can be used to determine which usb gadget driver is loaded

/sys/class/udc/<usb>/uevent

- Can be used to determine which usb gadget driver is loaded
- For example, before loading a USB gadget driver on AM335x:

```
root@am335x-evm:~# cat /sys/class/udc/musb-hdrc.0/uevent
```

/sys/class/udc/<usb>/uevent

- Can be used to determine which usb gadget driver is loaded
- For example, before loading a USB gadget driver on AM335x:

```
root@am335x-evm:~# cat /sys/class/udc/musb-hdrc.0/uevent  
USB_UDC_NAME=musb-hdrc
```


/sys/class/udc/<usb>/uevent

- Can be used to determine which usb gadget driver is loaded
- For example, before loading a USB gadget driver on AM335x:

```
root@am335x-evm:~# cat /sys/class/udc/musb-hdrc.0/uevent  
USB_UDC_NAME=musb-hdrc
```
- After loaded g_ether gadget driver:

```
root@am335x-evm:~# modprobe g_ether  
root@am335x-evm:~# cat /sys/class/udc/musb-hdrc.0/uevent
```

/sys/class/udc/<usb>/uevent

- Can be used to determine which usb gadget driver is loaded
- For example, before loading a USB gadget driver on AM335x:

```
root@am335x-evm:~# cat /sys/class/udc/musb-hdrc.0/uevent
USB_UDC_NAME=musb-hdrc
```
- After loaded g_ether gadget driver:

```
root@am335x-evm:~# modprobe g_ether
root@am335x-evm:~# cat /sys/class/udc/musb-hdrc.0/uevent
DRIVER=g_ether
USB_UDC_NAME=musb-hdrc
USB_UDC_DRIVER=g_ether
```

MUSB in debugfs

/sys/kernel/debug/musb-hdrc.<x>/

- regdump (read-only):
 - To get MUSB register dump
- softconnect (read/write):
 - Simulate usb device attach/detach for MUSB in host mode
- Testmode (read/write):
 - Set MUSB controller to test mode
 - http://processors.wiki.ti.com/index.php/UsbgeneralpageLinuxCore#musb_driver_debugfs

DWC3 in debugfs

/sys/kernel/debug/<xxxxxxxx>.usb/

- link_state (read/write):
 - USB3.0 link state
- Mode (read/write):
 - DRD mode (device or host)
- regdump (read-only):
 - To get DWC3 register dump
- Testmode (read/write):
 - Set DWC3 controller to test mode
 - http://processors.wiki.ti.com/index.php/USB_Test_Mode_on_DWC3

For more information

- sysfs Kernel Documentation:
<https://www.kernel.org/doc/Documentation/filesystem/sysfs.txt>
- sysfs bus USB ABI Kernel Documentation:
<https://www.kernel.org/doc/Documentation/ABI/testing/sysfs-bus-usb>
- Rules on Using sysfs Kernel Documentation:
<https://www.kernel.org/doc/Documentation/sysfs-rules.txt>
- 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)