USB System Design in Sitara Devices Using Linux

[Part 4]: Verify USB in Linux sysfs
Bin Liu (EP, Processors)
• 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

/sys/block/
/sys/class/
/sys/bus/*/devices/
/sys/devices/*/
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
```
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
Let’s see how device 44e3e000.rtc is referenced in sysfs.

It is the RTC module in AM335x:

```bash
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
```

In debugfs

References in platform bus

Bound to omap_rtc driver

Symbolic link to

The device
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/
|-- devices/platform/<ocp>/<usb>  # usb devices in a flat list
|-- bus/platform/devices/<usb>     # platform usb device references (controller, phy, …)
|-- bus/platform/drivers/<usb>    # platform usb drivers (controller, phy, …)
|-- bus/usb/devices/<usb>         # non-platform usb device references (bus, usb devices …)
|-- bus/usb/drivers/<usb>         # non-platform usb drivers (hub, bus, class, …)
|-- module/<usb>                  # usb kernel modules (*.ko) are here!
|-- class/udc/<usb>/uevent        # to check usb gadget drivers
`-- /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 # cppli41 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 xHCl controller
    |-- 488c0000.omap_dwc3_2 # usb2 module
      |-- 488d0000.usb # usb2 dwc3 controller
      |   `-- xhci-hcd.1.auto # usb2 xHCl 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/

<table>
<thead>
<tr>
<th>Devices in devices/folder</th>
<th>Device Name</th>
<th>Drivers in drivers/folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>47400000.usb</td>
<td>Virtual node for all devices below</td>
<td>am335x-usb-childs</td>
</tr>
<tr>
<td>47401400.usb</td>
<td>MUSB platform glue for USB0</td>
<td>musb-dspss</td>
</tr>
<tr>
<td>47401c00.usb</td>
<td>MUSB platform glue for USB1</td>
<td>musb-hdrc</td>
</tr>
<tr>
<td>musb-hdrc.0</td>
<td>MUSB controller for USB0</td>
<td></td>
</tr>
<tr>
<td>musb-hdrc.1</td>
<td>MUSB controller for USB1</td>
<td></td>
</tr>
<tr>
<td>47400000.dma-controller</td>
<td>CPPI41 dma controller</td>
<td>cpi41-dma-engine</td>
</tr>
<tr>
<td>47401300.usb-phy</td>
<td>PHY for USB0</td>
<td>usb_phy_generic</td>
</tr>
<tr>
<td>47401b00.usb-phy</td>
<td>PHY for USB1</td>
<td>am335x-phy-driver</td>
</tr>
<tr>
<td></td>
<td></td>
<td>am335x-control-usb</td>
</tr>
</tbody>
</table>
### AM57x USB entries in /sys/bus/platform/

<table>
<thead>
<tr>
<th>Devices in <em>devices/</em> folder</th>
<th>Device Name</th>
<th>Drivers in <em>drivers/</em> folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>48880000.omap_dwc3_1</td>
<td>DWC3 platform glue for USB1</td>
<td>omap-dwc3</td>
</tr>
<tr>
<td>488c0000.omap_dwc3_2</td>
<td>DWC3 platform glue for USB2</td>
<td></td>
</tr>
<tr>
<td>48890000.usb</td>
<td>DWC3 controller for USB1</td>
<td>dwc3</td>
</tr>
<tr>
<td>488d0000.usb</td>
<td>DWC3 controller for USB2</td>
<td></td>
</tr>
<tr>
<td>xhci-hcd.0.auto</td>
<td>xHCI controller for USB1</td>
<td>xhci-hcd</td>
</tr>
<tr>
<td>4a084000.phy</td>
<td>usb2.0 phy for USB1</td>
<td>omap-usb2</td>
</tr>
<tr>
<td>4a085000.phy</td>
<td>usb2.0 phy for USB2</td>
<td></td>
</tr>
<tr>
<td>4a084400.phy</td>
<td>usb3.0 phy for USB1</td>
<td>ti-pipe3</td>
</tr>
</tbody>
</table>
## USB entries in `/sys/bus/usb/`  

<table>
<thead>
<tr>
<th>Devices in <code>devices/</code> folder</th>
<th>Device Name</th>
<th>Drivers in <code>drivers/</code> folder</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>usb1</code></td>
<td>USB1 bus</td>
<td><code>usb</code></td>
</tr>
<tr>
<td><code>usb2</code></td>
<td>USB2 bus</td>
<td></td>
</tr>
<tr>
<td><code>1-0:1.0</code></td>
<td>USB1 bus roothub interface</td>
<td><code>hub</code></td>
</tr>
<tr>
<td><code>2-0:1.0</code></td>
<td>USB2 bus roothub interface</td>
<td></td>
</tr>
</tbody>
</table>

- 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 gEther 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:
  ```sh
cat /sys/class/udc/musb-hdrc.0/uevent
```
  ```sh
  USB_UDC_NAME=musb-hdrc
  ```
- After loaded gEther gadget driver:
  ```sh
  modprobe gEther
  cat /sys/class/udc/musb-hdrc.0/uevent
  ```
  ```sh
  DRIVER=gEther
  USB_UDC_NAME=musb-hdrc
  USB_UDC_DRIVER=gEther
  ```
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
DWC3 in debugfs

/sys/kernel/debug/<xxxxxxxxx>.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
  – [Link](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:  
- For questions about this training, refer to the E2E Community Forums at  
  http://e2e.ti.com