

USB Debug Techniques

This page outlines the common techniques to debug USB. Each section includes a link to the details.

Contents

[sysfs](#)
[debugfs](#)
[Dynamic Debug Interface](#)
[Ftrace](#)
[Tracepoints](#)
[USB protocol analyzer](#)
[Gadget Zero](#)
[Archived](#)

sysfs

The sysfs entry for enumerated USB devices is `/sys/bus/usb/devices/`.

```
root@am335x-evm:~# ls /sys/bus/usb/devices/
1-0:1.0 1-1      1-1:1.0 2-0:1.0 usb1    usb2
```

Each entry includes all the information about its USB device. The names that begins with **usb** refer to the USB controllers.

`x-0:1.0` is a special case, which refers to the root hub's interface. The devices are named as

bus-port.port.port

The interfaces follow in the form of

:config.interface

Details:

<http://www.linux-usb.org/FAQ.html#i6>

debugfs

```
/sys/kernel/debug/usb/
-- devices
-- usbmon
```

- `devices`: List of the devices enumerated on the bus with topology information.

Details:

<kernel source>/Documentation/us/proc_usb_info.txt

- `usbmon`: to capture USB traces on the BUS.

Details:

<kernel source>/Documentation/us/usbmon.txt

```
/sys/kernel/debug/<controller>/
-- regdump
-- testmode
-- mode*: host, device, or otg
-- link_state*: speed
```

*: only available in DWC3, not in MUSB.

Dynamic Debug Interface

Runtime turning on/off the kernel debug statements of `dev_dbg()` or `dev_vdbg()`. This interface is not limited to USB, it can be used to debug any driver which is ported to `dev_dbg()` framework.

Details:

<kernel source>/Documentation/dynamic-debug-howto.txt

Ftrace is part of kernel tracing infrastructure which has zero overhead when disabled and minimal overhead when enabled. It is a very useful tool to debug USB issues.

Details:

`<kernel source>/Documentation/trace/ftrace.txt`

Another tool in kernel tracing infrastructure for kernel debugging and performance diagnosis.

Details:

`<kernel source>/Documentation/trace/tracepoints.txt`

A USB protocol analyzer is a USB bus sniffer which captures, decodes, filters and display USB bus activities.

Gadget Zero is a simple USB gadget driver that is useful for testing controller drivers. It implements two configurations, one for source/sink, the other for loopback. As of in 3.12 kernel, gadget Zero supports Control, Bulk and Isochronous transfers. The kernel community is working on implementing Interrupt transfer support.

Details:

<http://www.linux-usb.org/usbtest/>

Sitara Linux SDK 08.00.00.00 (http://processors.wiki.ti.com/index.php/?title=USB_Debug_Techniques&oldid=198207)

Keystone=

{{

1. switchcategory:MultiCore=

- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **USB Debug Techniques** here.

C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article **USB Debug Techniques** here.

DaVinci=For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article **USB Debug Techniques** here.

MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article **USB Debug Techniques** here.

OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **USB Debug Techniques** here.

OMAPL1=For technical support on OMAPL1 please post your questions on The OMAPL1 Forum. Please post only comments about the article **USB Debug Techniques** here.

MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article **USB Debug Techniques** here.

For technical : please post your questions at <http://e2e.ti.com> Please post o comments ab article **USB D Techniques** t

}}

<u>Amplifiers & Linear</u>	<u>DLP & MEMS</u>	<u>Processors</u>	<u>Switches & Multiplexers</u>
<u>Audio</u>	<u>High-Reliability</u>	▪ <u>ARM Processors</u> ▪ <u>Digital Signal Processors (DSP)</u> ▪ <u>Microcontrollers (MCU)</u> ▪ <u>OMAP Applications Processors</u>	<u>Temperature Sensors & Control ICs</u>
<u>Broadband RF/IF & Digital Radio</u>	<u>Interface</u>		<u>Wireless Connectivity</u>
<u>Clocks & Timers</u>	<u>Logic</u>		
<u>Data Converters</u>	<u>Power Management</u>		

Retrieved from "https://processors.wiki.ti.com/index.php?title=USB_Debug_Techniques&oldid=198210"

This page was last edited on 22 April 2015, at 11:31.

Content is available under Creative Commons Attribution-ShareAlike unless otherwise noted.