SDIO in FreeBSD: status update 2017-08-04

Ilya Bakulin
kibab@FreeBSD.org

FreeBSD Developer Summit
University of Cambridge
Cambridge, UK
August 01 – 04, 2017

SDIO... What's that?

Wikipedia link here

- An extension of the SD specification to cover I/O functions.
- Earlier used for various external peripherals

FreeBSD

SDIO... What's that?

- Now extensively used to connect WiFi / Bluetooth modules
- Commercial products (Android phones, iPhone)
- Various development boards (Raspberry Pi 3, Wandboard)

SDIO in FreeBSD

- Current MMC stack doesn't support SDIO in any way
- There were several unsuccessful attempts to add support for it in the past
- Architectural deficiencies

SDIO support in FreeBSD

- New CAM-based stack committed in r320844
- Supports MMC, SD and SDHC cards
- SDIO support: recognition and minimal configuration

Feature highlights

- Based on CAM which gives queuing support, I/O scheduler,...
- Extended management capabilities using camcontrol(8)
- Supports interaction with the card from userland via cam(3)

How it looks like

Output generated on Wandboard with SD and MMC cards inserted in different slots

```
# camcontrol devlist -v
scbus0 on sdhci_slot0 bus 0:
<MMC 000000 1.1 SN 10000036 MFG 00/2006 >  at scbus0 target 0 lu
scbus1 on sdhci_slot1 bus 0:
<SDIO card>                          at scbus1 target 0 lun 0 (pass0)
<>                                   at scbus1 target 0 lun 1 ()
<>                                   at scbus1 target 0 lun 2 ()
scbus2 on sdhci_slot2 bus 0:
<SDHC 00000 1.0 SN 84280657 MFG 09/2014 >  at scbus2 target 0 lu
```

How it looks like

Read card and controller settings

```
# camcontrol mmcsdcmd pass1 -I
Host controller information
Host OCR: 0x360000
Min frequency: 97 KHz
Max frequency: 200 MHz
Supported bus width:  4 bit
 8 bit

Current settings:
Bus width: 4 bit
Freq: 50.000 MHz(high-speed timing)
```

FreeBSD

How it looks like

Read 32 bytes from addr 0 of I/O space of SDIO card using block reads

```
# camcontrol mmcsdcmd pass0 -c 53 -a 0 -l 32
CMD 53 arg 67108896 flags 35
MMCIO: error 0, 00001000 00000000 00000000 00000000
IO_RW_EXTENDED: read 32 bytes w/o error:
0000   32 02 00 00 00 00 00 40 02 70 10 00 00 00 00 00  |2......
0010   00 00 01 01 00 00 00 00 00 00 00 00 00 00 00 00  |.......
```

FreeBSD

Drivers in userspace: brace yourself!

- Send MMC commands from userspace using cam(3)
- No more kernel panics during development. Trying new version means recompiling one binary and starting it on the board
- Example communicating with Broadcom SDIO WiFi chip
- TODO: Support kqueue(2) to delivery interrupts from SDIO card back to userland

Open tasks: the stack itself

- Newbus integration
- Support faster cards (UHS-I, UHS-II, fast eMMCs)
- Implement features recently added to the "old" MMC stack
- Improve support for writing device drivers in userland

Open tasks: peripheral drivers

- SDIO effectively means WiFi for now
- Raspberry Pi3 and Wandboard use Broadcom chips
- Wirless versions of Beaglebone Black use TI WiFi
- Various Atheros-based routes use AR6xxx chipsets
- At least some Chromebooks use Marvell

Broadcom support is the most important for now.

Timelines

- I'm doing this in my free time
- So, no timelines whatsoever
- Broadcom driver development depends on bhnd(4) support by Landon Fuller

Where to find more information

- SDIO on FreeBSD Wiki – first in Google "SDIO" search! Includes information on how to test the new stack on Beaglebone and Wandboard. Raspberry Pi3 is coming.
- GitHub project and task board
- GitHub branch – I'm not a committer, so I commit everything there and then convince imp@ to commit stuff

Thank you for your attention!

Q?