EV3 v1.12 Release README

11/07/2016

Change history:

v1.12 Release

Bench tested using kernel 4.9-rc2 and kernel 3.10.0-327.18.2.el7.x86_64
Inspur script ev3load_alt testing using 2.6.32-431.el6.x86_64
RTL: 00.05.04
FW   00.05.07
FPGA 00.19

Goal: To remove echo statement in script for driver load for a specific customer - Inspur

-No driver changes were needed.
-New file ev3load_alt was added for Inspur. They need to rename this for use in /etc/init.d or get external. It is the same as ev3load with 3 "echo statements commented out as requested" plus "the passed parameters when invoking insmod are no longer passed". This is due to /etc/init.d gets a parameter "start" passed to it. The customer did not want to use "service ev3load start" for example.
-Updated the version strings only - no other changes in the driver or application.

Instructions for loading driver on boot

If 'lspci' is not available on the system do

```
# yum install pciutils
```

```
# cp <path to driver source>/tools/ev3load_alt /etc/init.d/ev3load
# cd /etc/init.d
# chkconfig --add ev3load
# chkconfig --level 2345 ev3load on
# reboot
```

On boot the device nodes should all be present

v1.11 Release

Bench tested using kernel 4.60 and kernel 2.6.32
RTL: 00.03.04
FW   00.03.08

Goal: To build the EV3 driver into the Linux kernel for an older Centos 6.4 distribution.

-No driver changes were needed.
-New file named BuiltIn_kernel_Makefile
-Made change to make_partitions.sh as an experiment.
-Updated the version strings.

v1.10 Release (re-issue)

Bench tested using kernel 4.60
RTL: 00.03.04
FW 00.03.08

Goal: Using 2 EV3 cards one card is okay with 128 partitions but the second card winds up with 127 max. I decided to bump the max_partitions to the maximum number allowed by the Linux kernel which is defined by DISK_MAX_PARTS

Reference Link: http://superuser.com/questions/306734/what-is-the-maximum-number-of-partitions-with-efi

1. I changed EV_MINORS from 129 to DISK_MAX_PARTS (256)
2. I changed the date of the release but not the version. This is to be considered a re-issue.

v1.10 Release

Bench tested using kernel 4.60
RTL: 00.03.04
FW 00.03.08

Goal: To support 128 partitions per EV3
Same as v1.09 plus
Risk level: Low - a system could potentially preallocate too much unused memory due to 128 partitions.

1. Changed EV_SHIFT from 6 to 7 to support up to 128 partitions.
2. max_partitions is defaulted to 128 per card.
3. I modified "make_partitions.sh" script which is now included in the tools directory.
4. Version strings have been updated.

For example to create 128 partitions on /dev/ev3memb using 32MB partition sizes use:

```
./make_partitions.sh b 128 32M
```

There were two test cases that failed.
A. 128 partitoins on a single EV3, then power down and up again, it would not load the partitions.
B. On two cards in a single system cannot list all partitions, see only 64 partitions on one card.

I was not able to test case B due to potentially bad EV3 16GB card. I need Hao to verify both cases.

v1.09 Release

Bench tested using kernels 3.10.0-327 and 4.6.0
RTL: 00.03.04
FW 00.03.08

Same as v1.08 Release plus
Risk level is moderate. Mostly affects kernels 4.7 amd later.

Goal: To support PowerPC (Little-endian 64-bit)

Four main issues were seen on the IBM PowerPC platform

1. The IBM platform has no BIOS and depends on the Linux OS to map the BAR registers. This required the use of a none-zero class code in PCI space. I believe that 0x0180 (Other mass storage controller) _OR_ 0x0580 (Other Memory Controller) was selected - not sure which one. In any case the new class code allowed the card to be mapped on IBM's PPC system.

2. The IOCTL data structure was too large. at 8824 bytes in size it did not fit the PPC constraint of 8192 bytes/ To fix this the JSON data logger buffer was decreased from 1024 entries to 256.

3. The function rdtscli is not available under PPC. I used a macro I found online to similate it. I could not test this function. This funtion is used to get real-time high speed clock from the CPU and uses assembly language. Fortunely this is used only for gathering performance statts which may no longer be accurate anyway.

4. The native page size on PPC can be 4K or 64K. In this case it was 64K which means that one page no longer fits in one descriptor. The driver now looks at the size of the scatter/gather entry and if it is greater than 32K it will break up on SGL entry into 32K or smaller descriptors.

Additional changes:

The DMA API has changed around kernel 3.9 but still supportis the older API. The older API will go away at some point in the future. To future-proof this I implemented a switchover to the new API starting on kernel version 4.7.0. It is not a big change since the legacy support is a wrapper around the new API so the risk here is faily minimal, but DMA should be verified on 4.7.0 or later.

Compile options "bio=1" and "sgio=1" was failing. I cleaned up the compiler errors there.

Version string have been updated.

Please test

Using both current FPGA releases and the new FPGA release with the non-zero class code.
Data integrity.
On PPC system if available
kernel 4.7.0 or later
JSON data logger
performance stats under PPC and x86. This may no longer work due to performance threading changes. But we may have to de-feature this code.

v1.08 Release

Same as v1.07 Release plus
Risk level is low. Mostly affects kernels between 4.4 and 4.0 compiles.

Goal: There was a problem seen using kernels 4.0.0 under Ubuntu and compile issue seen on kernel 4.2.0 under Ubuntu.

Two real problems were fixed.

1) A bug in the compile options based on kernel version is seen that prevents the "make_request" function from returning the proper value for certain kernel versions. between 4.0 and 4.4. It would compile under 4.0.0 but the return value would not be done properly.

2) The kernel version 4.2.0-16.generic came with the current installation of Ubuntu 15.10 would not have compiled due to the same issue on #2.

3) I added another warning level option to make it closer to Ubuntu -Wreturn-type, when the user invokes "UBUNTU_WARNING_LEVEL=1"

4) Version strings have been updated.

v1.07 Release

Same as v1.06 Release plus
Risk level is small: Most significant change was the JSON string initialization. Other changes were cosmetic.

Goal: Fix compile warnings under Ubuntu
Test: Using Ubuntu 14

-Got rid of warning messages due to higher warning levels used on Ubuntu
-Added way to elevate the warning level on non-Ubuntu to closer to Ubuntu level. I added the compile options "-Wformat -Wimplicit-function-declaration -Wformat-security", but I could have missed more flags. The original ev3make works as it was by default. The compile flags apply to both driver and ev3util appilication.

Use this as a fairly good way to compile as Ubuntu on non-Ubuntu

```
# ev3make UBUNTU_WARNING_LEVEL=1
```

-Version strings have been updated.

v1.06 Release

Same as v1.05 Release plus

Goal: Support Broadwell IOCTL - The Broadwell systems are not allowing the IOCTL to dereference a pointer as used in previous versions of the driver.

IOCTL struct v1.05 and earlier:

```
{
        void *arg               <---Obscure pointer to various other data structures
        int errnum;
}
```

IOCTL struct v1.06 and later

```
{
        int errnum;
        union ioctl_union       <---Union of all other data structures - no pointer
}
```

This required every IOCTL call to change in both the application and driver layer.
This new mechanism is better since it does a single "copy from user" and a single "copy to usr" call instead of the previous case where the copy from and copy to were scattered throughout the driver IOCTL code using mixed "put_user" and "copy to user", making it difficult to maintain. Also while at it the entire IOCTL code was cleaned up and made consistent for easy of maintenance and to allow a consistent error handling behavior. A lot of code that was previously WIP or just dead code was removed.

NOTE: Due to fairly large changes the risk level is fairly high. Please test all the IOCTLs for both Broadwell and non-Broadwell systems.

-New IOCTL structure
-Unused IOCTL codes were removed.
-#defines for previous hard-coded values of errnum
-Data structure naming convention is now more consistent
-IOCTL error handling is now more consistent
-Got rid of almost duplicate routine called "aux_hw_access" and "aux_hw_access_2", now uses one routine.
-Unused routines were removed. I am not sure I got them all though.
-Code that was WIP or TBD but not used was removed.
-The script test_pattern.sh had a large number of old debug code that was removed, it was running the same test 700 times and this was not intended to go to customer. Now cleaned up.
-Version strings have been updated

v1.05 Release

Same as v1.04 Release plus

-Support for Linux kernel 4.6

Reference link:

https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=09cbfeaf1a5a67bfb3201e0c83c810cecb2efa5a

-Fixed version output of "get_message
-Version strings were updated

v1.04 Release

Same as v1.03 Release plus

-Support for Linux kernel versions up to and including 4.5
-Warning mesage cleaned up during DBG=1 compile option
-Version strings were updated for ev3util and driver.

v1.03 Release

Same as v1.02 Release plus

ev3util change only, except for version strings.

-IFP bug where ARM state was not getting set correctly was fixed. Also hardwired numbers were replaced with #defines.

-Version strings were updated for ev3util and driver.

v1.02 Release

Same as v1.01 Release plus

Added support for load-time setting of the maximum partitions. The default is 16, to support more (or less), decide on the number of partitions wanted and add one. I added a debug print to show the setting used, it will show up in the messages log file.

For example to support 36 partitions, add one and pass it to the "max_partitions" parameter. This is an option for the block device driver only. There may be a memory limitation is the number is too high. It would be interesting to find out what the realistic maximum is.

```
# ./ev3load max_partitions=37
```

Version strings have been updated.

v1.01 Release

Same as v1.00 Release plus

IOSTAT support is compiled in but not used by default. It can be compiled out altogether if wanted.

To load the driver with IOSTAT support do:

```
# ./ev3load iostat_enable=1
```

To load the driver without IOSTAT support do:

```
# ./ev3load
```

To compile out any support for IOSTAT do (it is case sensitive):

```
# ./ev3make IOSTAT=0
```

Version strings have been updated.

v1.00 Release

Same as v1.00 Alpha 26 plus

-Performance code using work queues by default has been incorporated. Work queues apply to kernels 2.6.36 and later. Earlier kernels will still use a tasklet even if tried to compile otherwise and will not use work queues for IO completion due to performance.

-A compile-time option allows users of kernels 2.6.36 and later to override work queue operation and use tasklet mode. To do this use:

```
# ev3make use_tasklet=1
```

-Removed unneeded include which prevented kernel 4.2 from compiling. Jack's change.

-Removed unused lock in proceess interrupt.

-Added new tracking variable - dmas_num_outstanding for perfmormance monitoring.

-Cosmetic cleanup of unused dead code.

-Code for work queues for older then 2.6.36 kernels is in place and works - it is just not performing well.

-EV3UTIL - program_fpga - code that was done to fix and cleanup program_fpga under Windows has been brought in.

-Version strings have been updated.

v1.00 Alpha 26

```
FPGA Version                 : 00.16
FPGA Image                   : Application image
NV RTL Version               : 00.03.04
NV FW Version                : 00.03.08
FPGA Configuration Number    : 2
FPGA Board Code              : 0
Driver Version/Date/Status : 1.0 02/01/2016 Alpha 25
EV3UTIL Version/Date/Status: 1.0 02/01/2016 Alpha 25
```

Same as v1.00 Alpha 25 plus

Per Scott's feedback:

1) Please move the placement of ledControl to after the control section. I will get you the decode.

2) Please change fpgaVersion to fpgaDmaVersion

3) Please change nvRtlVersion to fpgaNvRtlVersion

4) Please change nvFwVersion to fpgaNvFwVersion

5) Please move the placement of fpgaImage to below fpgaNvFwVersion

6) statusRegisterAccessReady is reporting Not Active which is not correct. I will double check it.

7) The flashReserveBlockLevel and associated Warning and Errors thresholds need to have the decode added.

8) Four the values that are counts, I think we can leave off the Count and just display the value. Let me know if you think this makes sense.

9) The pmuTemperatureAverage returned 0xFF so we will need to investigate that value.

10) For the log data, I think we need to wait longer for the data to be collected. We will do that tomorrow.

Done:

-Added proper decode fro flashReserveLevel and related warning and error settings.

-Added LED control decode.

-Moved LED control past the controlEnablePmuAutoTest field.

-Removed "Count" units throughout.

-Fixed PMU Average temperature bug. I had picked the wrong offset in the table.

-Fixed bug with register decode where the shifted value was not being saved. This resulted in the incorrect decoding of the fields like statusRegisterAccessReady. Many others were also affected as this is a generic decoding routine.

-Renamed some fields as requested.

-Version strings were updated.

v1.00 Alpha 25

```
FPGA Version                 : 00.16
FPGA Image                   : Application image
NV RTL Version               : 00.03.04
NV FW Version                : 00.03.08
FPGA Configuration Number  : 2
FPGA Board Code              : 0
Driver Version/Date/Status : 1.0 02/01/2016 Alpha 25
EV3UTIL Version/Date/Status: 1.0 02/01/2016 Alpha 25
```

Same as v1.00 Alpha 24 plus

-JSON API has been added. to invoke use the command-line "json all" as the typical case

ev3util /dev/ev3mema **json all**

Or each command may be individually invoked if wanted instead of the "all" command.

```
cardCtrl
cardInfo
cardStatus
errorGroup
warningGroup
errorThresholds
warningThresholds
flashStatus
pmuConfig
pmuInfo
pmuStatus
logData
```

-The driver now captures card temperature, fpga temperature and PMU temperature on an hourly basis for the last 24 hours. These are the default settings. During driver load-time these can be changed using

```
data_logger_sample_time_secs        (default 3600)
data_logger_sample_length           (default 24)
```

For example to sample every 30 seconds do (all on the same line):

```
insmod /lib/modules/3.19.8/extra/netlist/netlist_ev3.ko
data_logger_sample_time_secs=30
```

-There is a new IOCTL to allow ev3util to retrieve the logged data (aka logger data)
-There is a new set of files that contain the generic JSON code:
    ev3_json.h and ev3_json.c
-There is a new set of files that contain the EV3UTIL JSON commands specific to EV3
    ev3_json_cmd.h and ev3_json_cmd.c
-I made the implementation into separate files so that these could be integrated to Windows easier and it is better to separate the modules by functionality sunce ev3util is getting quite large.
-I also made it so that the "human readable" format could be output in some future implementation, in fact it is extensible and if some customer wants a different API we can failry easily add that as well.
-The number of partitions supported is now 16.
-Version strings have been updated.

NOT DONE:

-I did not include any of the performance enhancement code that was used in a prior experimental release of v1.00 Alpha 25. This will be merged on some future release.

v1.00 Alpha 24

FPGA Version: 00.16 "BL16"
RTL: 00.03.03
FW: 00.03.05
Card Revision C.

Older FPGA versions are supported but encouraged to upgrade to BL16 or later.

Same as v1.00 Alpha 22 plus

-Support for kernel 2.16.28
-i2cRegs.h and bit changes related to BL16 and latest ProcSys document. The out-of-order messages are gone. The interrupt status bits changed which may make older FPGA releases not work properly. I recoomend everyone going to BL16 or later if possible.
-ECC monitoring and error injection is supported and is enabled if BL16 or later FPGA is used. No passcode protection is used for this. ECC enabled also implied that AER is enabled.
-AER is enabled for BL16 and later. Unknown if it is working properly or not yet.
-Limited PIO tests are implemented and enabled. Not the ones that won't work. The ones that won't work are coded brought up to EV3 level but cannot be tested yet until FPGA changes are made. A fuller set PIO tests are compiled out. To enable use them, uncomment #defines listed at the top of netlist_ev3.c

```
// #define ALL_PIO_ACCESS_SIZES_SUPPORTED
// #define ALL_PIO_ACCESS_ALIGNMENTS_SUPPORTED
// #define MEMSET_MEMCMP_MEMCPY_ACCESS_SUPPORTED
```

-PIO performance measurement was added on pio_test 2.
-Support for hidden user commands is in for ev3util.
-Support for ECC error injection is in and generally tested but not in great detail. It seems to work.
-P/N fix for line feed is in.
-Minor formatting of perf_stats output.
-Many commands for Netlist use only are compiled in but hidden from the help menu.
rb, wb, rw, ww, rq, wq, max_descriptors, write_access, prb, pwb, prw, pww, prq, pwq
-Commands that sre still compiled out but can be compiled in for factory use, etc.
irb, iwb, di, pmu_read, pmu_write, pmu_info, pmu_update.
-User Guide was updated to v1.00 Alpha 24
-Version strings were updated.

v1.00 Alpha 23

Does not exist due to synchronization with Windows release.

v1.00 Alpha 22

FW 00.03.00 and later
Special release:
RTL Version: 00.02.0C
FW Version: 00.02.0E
BL Version: 0B => This is not important since it does not affect the register location.

Same as v1.00 Alpha 21 plus

-I merged the changes for cleaning up the warning messages seen on Ubuntu on ev3util.
-I fixed an IO timeout case using DMA interrupt coalescing.
-Version strings have been updated.

Usage:

.
To use coalescing load the driver with a value dor dma_int_factor of between 2 and 31

For example this will generate an interrupt every 16 DMAs.

```
# ev3load dma_int_factor=16
```

It has been bench tested but more testing and tweaking may be needed.

v1.00 Alpha 21

FW 00.03.00 and later
Special release:
RTL Version: 00.02.0C
FW Version: 00.02.0E
BL Version: 0B => This is not important since it does not affect the register location.

Same as v1.00 Alpha 20 plus

-I ifdef'd out access to the I2C space and the ability to program the FPGA.
-I fixed I2C access via the driver.
-Version strings were updated.

v1.00 Alpha 20

FW 00.03.00 and later
Special release:
RTL Version: 00.02.0C
FW Version: 00.02.0E
BL Version: 0B => This is not important since it does not affect the register location.

Same as v1.00 Alpha 19 plus

-I add extra level of protection from polling timer for the default case of interrupt for each DMA. A timeout had been seen and this new code is suspect.

-I the decoding of the card treats the high byte of EV3_SERIAL_NUM0 as a character for decoding.

-Version strings have been updated.

v1.00 Alpha 19

FW 00.03.00 and later
Special release:
RTL Version: 00.02.0C
FW Version: 00.02.0E
BL Version: 0B => This is not important since it does not affect the register location.

-"get_model" now displays the card and PMU part and serial numbers correctly.

-"nv" now shows the FPGA temperature.

-"get_version" now displays "Factory image" or "Application image".

-"beacon" command for ev3util is now working. All controllable LEDS will blink if "beacon 1" is used and controlled by the FW if "beacon 0" is used.

-Added real MSI interrupt counter to distinguish real interrupts from polled interrupts.

-I fixed "polling_mode". This should be functional now. Not tested much.

-Added interrupt coalescing capabilities which are off by default. This is not fully functional and will have time outs. Do not use this yet, it is work in progress.

-Other work in progress is I2C at the driver level "irb", "iwb", "di" along with implementation of the "passcode" command. The code is there but not used yet. ev3util has the I2C commands ifdef'd out.

-I changed ev3load to just pass the load time parameters directly to the driver.

-Version strings have been updated.

Usage notes:

To enable polling mode

```
# ev3load "polling_mode=1"
```

To enable interrupt coalescing

```
# ev3load "dma_int_factor=1"
```

For other parameters:
For now - you can search the source code for other passable parameters. Searh for the string MODULE_PARM_DESC and then pass the new setting via ev3load as shown earlier. These will be properly documented at some point.

v1.00 Alpha 18

FW 00.03.00 and later
Special release:
RTL Version: 00.02.0C
FW Version: 00.02.0E
BL Version: 0B => This is not important since it does not affect the register location.

Corrected descriptor alignment to 16 bytes.

Code is using new i2cRegs.h file released on 09/08/2015. Backward compatibility to the special release and other legacy versions is maintained. The new FW version is 00.03.00 and later uses the new offsets as defined by the i2cRegs.h file.

ev3util.c - Support added for all the different register offsets. Also added FPGA temperature for the "nv" command.

ev3util.c - get_model displays part number and serial number information for card and PMU.

ev3util.c - Some #defines were used instead of hardwired numbers for some recently added routines.

Version strings were updated.

NOTE: This has not been tested as of release please test using both older and new FW releases.

v1.00 Alpha 17

Special release:
RTL Version: 00.02.0C
FW Version: 00.02.0E
BL Version: 0B => This is not important since it does not affect the register location.

FPGA: EV3_RTL0109AFW0108BL0A.jic (tested with this FPGA)
FPGA: EV3_RTL010AFW0108BL0A.jic.
FPGA: EV3_RTL0102FW0101BL04.jic

Same as v1.00 Alpha 16 plus:

ev3util:
For FPGA special release only the "nv" command will change the offset of the Card temperature register to 0x414 and also read the FPGA temperature at offset 0x410 and display the results.
For all other releases the original format and offsets apply.

-Version strings were updated

NOTE: This release needs testing using the "special release" JIC.

v1.00 Alpha 16

FPGA: EV3_RTL0109AFW0108BL0A.jic (tested with this FPGA)
FPGA: EV3_RTL010AFW0108BL0A.jic.
FPGA: EV3_RTL0102FW0101BL04.jic

Same as v1.00 Alpha 15 plus:

ev3util: IFP and PMU changes received from Suzhou on 09/11/2015
-Version strings were updated

NOTE: This release needs retesting. Please use this version for future changes from Suzhou.

v1.00 Alpha 15

FPGA: EV3_RTL0109AFW0108BL0A.jic (tested with this FPGA)
FPGA: EV3_RTL010AFW0108BL0A.jic.

FPGA: EV3_RTL0102FW0101BL04.jic

Same as v1.00 Alpha 14 plus:

-Merged in IFP changes from Suzhou.
  program_fpga was created for EV3
  pmu_read
  pmu_write
  pmu_info
  pmu_update
  LAST_VALID_REGISTER_ADDR is now AFF
-Out-of-order global was moved to the card structure.
-Bug fix, the "nv" state was not displayed due to a recent change, now displayed properly.
-Version strings were updated

v1.00 Alpha 14

FPGA: EV3_RTL0109AFW0108BL0A.jic (tested with this FPGA)
FPGA: EV3_RTL010AFW0108BL0A.jic.
FPGA: EV3_RTL0102FW0101BL04.jic

Same as v1.00 alpha 13 plus

-Compiles up to kernel 3.19.8
-ev3load makes as many nodes as cards are found. It queries the driver and uses similar bash code that was used on qualtest to create the nodes.

Updated version strings.

v1.00 Alpha 13

FPGA: EV3_RTL0109AFW0108BL0A.jic (tested with this FPGA)
FPGA: EV3_RTL010AFW0108BL0A.jic.
FPGA: EV3_RTL0102FW0101BL04.jic

Goal: Use this release to debug the NV state change opertion.

I changed the version info handling to be done entirely by the driver and not ev3util.

I decode the status buffer entries such that non-DMA statuses get handled separately as they should be. The NV Status change interrupts will show up in the log file as they happen. Use this for debugging the NV status issues.

Descriptor throttle is no longer used if "bridge logic" 10 or later is used.

I removed 8 unused IOCTLs.

I cleaned up the operation of the "force_save", "force_restore" and "arm_nv" commands. These had some code from EV1 that had not been properly updated. I added a "disarm_nv" IOCTL.

The driver now reports the number of cards it sees. This will be used to allow the ev3load script to create the proper number of nodes. The "get_model" command of ev3util shows the number of cards detected. ev3load has not been updated yet to make use of this information. Planned for next release.

ev3util - I added "disarm_nv" command.
ev3util - I removed old commands like "erase_flash" and the ones that end with "_complete".
ev3util - "nv" command: Out of range STATE_REG will produce "undefined state".
ev3util - "nv" command: FPGA version is now in hex.
ev3util - I started the process of creating a multi-buffered system to transfer the FPGA RBF file data to the driver to allow the update to happen without interruptions. This is still WIP as part of the FPGA update command.
ev3util - Cosmetic cleanup, took out a few unused functions.

Version strings updated.

v1.00 Alpha 12

FPGA: EV3_RTL010AFW0108BL0A.jic.
FPGA: EV3_RTL0102FW0101BL04.jic (tested with this FPGA)

Same as v1.00 Alpha 11 plus:

Added IOCTL for CARD_READY, the command is decoded by the driver and returns TRUE or FALSE. ev3util has a new command "card_ready" to show if card is in ARMed state or not.

Fixed the source of one crash when doing multi-threaded "dd" commands. This was related to a semaphore that was not needed.

Fixed a second case of crash due to doing multi-threaded "dd" commands. This was due to not releasing pinned down memory when "Out of resources" conditions were encountered.

I found that the wait queue events were waking up all queued tasks and not one per event as expected. Changing from "add_wait_queue" to add_wait_queue_exclusive" fixed that issue.

I removed unused and unimplemented code related to MMAP_MODE_PIO. This was cosmetic and lead to potential confusion.

I added spinlocks to protect some currently disabled by default debug code.

I updated the "i2cRegs.h" to per the 8/17/2015 that I got from Scott. I believe this to be for EV3_RTL010AFW0108BL0A.jic. It works for older FW as well.

ev3util - I added "card_ready" command per Scott's email for Ocarina support.
ev3util - nv command: I changed the NV RTL and NV FW versions to be 2 hex digits per Scott's email.
ev3util - nv command: I display the 2 missing bytes of the ERROR_CODE per Jesse's comment.

Updated version strings.

v1.00 Alpha 11

FPGA: EV3_RTL0102FW0101BL04.jic and later
Same as v1.00 Alpha 10 plus:

Added ability to do asynchronous DMA via IOCTL while maintaining synchronous capability.

Cleaned out old fields and passed parameters that are confusing and no longer used.

by_enabled
do_bken
poll_thr_exit
user_req

ev3util - I created a new function called "fill_pattern_async" to demonstrate asynchronous operation. It works exactly the same as "fill_pattern" but just in asynchronous mode.

ev3util - I went through all the DMA functions and cleaned them up. They were using unused fields and were not consistent. There is more cleanup needed but there are lots of changes.

Updated version strings.

v1.00 Alpha 10

FPGA: EV3_RTL0102FW0101BL04.jic and later

Same as v1.00 Alpha 9 plus:

Performance enhancement for 2.6.x kernels by removing "unplug" routine contents. Unplug routine is disabled by default. If any problem shows up it can be enabled again using "./ev3make UNPLUG=1". Now this works similar to 3.x kernels.

Updated version strings.

v1.00 Alpha 9

FPGA: EV3_RTL0102FW0101BL04.jic and later

Same as v1.00 Alpha 5 plus

A hang of the character driver due to offset not getting updated was fixed. Some cleanup was done in the lseek routine.

A hang of the block driver under kernel 2.6.x when using "dd" and "bs=4096" or greater was fixed. This was due to a call to "unplug function" never getting called when "dd" was used under the conditions mentioned.

Added a descriptor based throttle to allow up to 800 outstanding descriptors to prevent kernel crashes due to known issue with current release.

The driver allows the use of full 256 descriptors per DMA.

Higher performance is enabled by turning off DEBUG code by default when compiling. If this is wanted use "ev3make DBG=1", otherwise DEBUG is off. Also "make DBG=1" works the same way.

Internal DEBUG code to view descriptor minimums and maximums is turned off via #define. If wanted this can be turned on by commenting out the line in netlist_ev3.c
#define DISABLE_DESCRIPTOR_DEBUG

ev3util - A new command "max_descriptors" was added to allow viewing and changing the throttle value. It works the same way that "max_dmas" works.

Old "legacy_mode" code was removed.

Version strings were updated.

v1.00 Alpha 8 - not released to anyone

v1.00 Alpha 7 (INTERNAL USE ONLY)

FPGA: EV3_RTL0102FW0101BL04.jic and later

I set the driver tuning for 3.x and later kernels to allow 256 descriptors per DMA.

I added the driver tuning settings back in for 2.6.x kernels and earlier. This has not been tested.

I took out an old "legacy_mode" parameter for EV3UTIL.

I updated the version strings.

v1.00 Alpha 6 (INTERNAL USE ONLY)

FPGA: EV3_RTL0102FW0101BL04.jic and later

I started driver performance tuning to receive larger descriptor sizes. A bad side effect is that timeouts happen much more frequently and at "smaller" IO sizes.

I added debug code to count the number of BIO READS and BIO WRITES.

I added debug code to capture the largest and smallest descriptor payload size and the largest descriptor index used. The information is dumped via the "log" command.

I updated the version strings.

v1.00 Alpha 5

FPGA: EV3_RTL0102FW0101BL04.jic and later

I increased fixed EV_MINORS from 4 to 5 to allow disk node for ev3mema plus
ev3mema1 through ev3mema4 a total of 5 disk nodes.

Use ***fdisk*** to create the partitions and then use ***partprobe*** to let Linux know to use them.

device id was being read incorrectly from offset 4, it should have been offset 2 in config space, this was fixed.

netlist_blk_get_geo - was incorrectly reporting 0 cylinders due to a 16 bit field overflow condition. Now the correct 8192 cyclinders are reported correctly.

v1.00 Alpha 4

FPGA: EV3_RTL0102FW0101BL04.jic and later

Fixed compile issue on kernel 3.8.0 (Ubuntu 13.04) - VM_RESERVED flag was removed.
ev3util - I cleaned up compiler warnings that were introduced in the previous release.
Version strings updated.

v1.00 Alpha 3

FPGA: EV3_RTL0102FW0101BL04.jic and later

ev3util - nv command calculation of last backup power and formatting per Scott.

v1.00 Alpha 2

FPGA: EV3_RTL0102FW0101BL04.jic and later

I merged the current version of i2cRegs.h
System crash fixed due to calling dbg_stats without a lock.
Version strings updated
ev3util usage dialogg was not showing "fill_pattern" or :verify_pattern".
ev3util - char_driver_test no longer dumps the buffers at end.
ev3util - nv command is formatted per Scott's feedback.

v1.00 Alpha 1

FPGA: EV3_RTL0707FW0708.jic and later

This is the first version that is intended to be used by a customer

New ev3util commands "force_restore" and "arm_nv" have been added.
Proper sizes are reported and using the device ID to determine size.
Support for any non-Netlist vendor ids was removed on NETLIST is supported.
Compile warning messages were cleaned up.
I removed debug code that had been created for the bring-up phase.
Lots of cosmetic cleanup in ev3util
ev3make is a new file in the tools directory to make and install the driver. If compiled parameters

are needed pass them to this:
For example:

```
ev3make sgio=1
```

Compiled and tested using kernels

- 3.18.17
- 3.10.0.229.4.2.el7.x86_64

v0.20

FPGA: EV3_RTL0707FW0708.jic and FPGA: EV3_RTL0707FW0706.jic

Minor change - Modified the out-of-order output message so that it does not fill the log.

v0.19

FPGA: EV3_RTL0707FW0708.jic and FPGA: EV3_RTL0707FW0706.jic

I added a buffer to track completion context and to be able to detect any out-of-order SGL completions if they were to occur. None did occur so far. A dbg_capture event of 0x69 will indicate that out-of-order has occurred if it ever sees such event.

ddr_mem_addr_hi in the descriptors was not getting set for BIO traffic. This was fixed but has no bearing on current miscompare issues.

Debug code was cleaned up to show full card address including the high byte.

FPGA version info is captured and reported by the driver.

STATUS buffer was not explicitly initialized in the driver, now it is. No effect on anything was seen.

ev3util - get_version now displays the NV version info, not the "nv" command.
ev3util - nv command now has some feedback from Scott implemented.

Version strings were updated.

v0.18

FPGA: EV3_RTL0707FW0708.jic

I aligned the descriptors to 4096 page boundaries. This seems to have fixed at least partially some issues with 1MB FIO testing. No other issues were solved by this.

I aligned the data to 64 byte alignments.
I aligned the SGL vectors created in user space but it turns out that I did not have to do this since they get copied into the driver's SGL array which are already aligned so that was not really needed.

```
fill_pattern            Data and SGL list is aligned
verify_pattern          Data and SGL list is aligned
read_pattern            I removed this unused function
fill_user_pattern       Data and SGL list is aligned
```

```
verify_user_pattern     Data and SGL list is aligned
load_evram              Data and SGL list is aligned
save_evram              Data and SGL list is aligned
char_driver_test        Data and SGL list is aligned
```

I added NV version information to the "nv" command.

I set "nv" output temperature format to 1 decimal place. It is now 3.1 format.

I fixed ProcSys debug code alignment.
from:
`0xffffffffffffff40ULL`
To
`0xffffffffffffffC0ULL`

I fixed the "CPU stuck for more than 22s" by exiting the process_completions tasklet after 256 loops and rescheduling itself if there's more work to do on exit. This worked as far as not getting that message. I ran the perf_suite and did see a timeout during mixed READ/WRITE 512 bytes which may or not be related to this particular change.

The "force_save" command has been implemented. It does not wait for the completion just starts the save and returns. Using "nv" the state machine can be monitored.

Caveats

I am noticing with the current FPGA and ast least as far back as v0.16 that the "load_evram" is taking upwards of 157 to 275 seconds to complete. The save_evram is only taking 32 seconds. Something is not right somewhere.

v0.17

FPGA: EV3_RTL0707FW0708.jic

The data buffers and SGLs have been aligned for "load_evram" and "save_evram". No changes in data compare issues were seen.

Two new debug commands were added "le" mirroring "load_evram" and "se" mirroring "save_evram" but with extra control parameters and will display SGL descriptor data as they execute.

No options uses the same defaults as the original commands.

Options
le <filename> <number of descriptors> <buffer size in KB>
se <filename> <file size in KB> <number of descriptors> <buffer size in KB>

For example:

Load file use 8 descriptors for DMA and a 4KB buffer per descriptor.
le ./src/8G_INC_QW 8 4

Save a file of 16KB using 4 descriptors and an 8KB buffer per decriptor
se ./dst/16K_INC_QW 16 4 8

Caveats

I did not get a chance to get the buffers filled with a background pattern.
Other routines below will need to at least get the SGL list aligned still

```
fill_pattern            Already ALIGNED
verify_pattern          Already ALIGNED
read_pattern            Already ALIGNED - Routine is unused will remove it.
fill_user_pattern       Already ALIGNED
verify_user_pattern     Already ALIGNED
load_evram              Unaligned buffers
save_evram              Unaligned buffers
char_driver_test        Unaligned buffers
```

WINDOW_SELECT register back to 24h
I verified and define new register offsets per register_handler.v.

I merged in Jack Shao's changes to support kernels newer than 3.14. This will compile on Ubuntu at least to kernel 3.18.17

"status" command is renamed to "nv".

version strings updated.

v0.16

FPGA: EV3_RTL0707FW0706

I used the correct WINDOW_SELECT offset and PIO seems to be working properly across all memory range.

I cleaned up the new ev3util "status" command.

The cuurent and correct "i2cRegs.h" file is being used now.

I updated the version strings.

v0.15

FPGA EV3_RTL0704FW0703.jic

I included i2cRegs.h a file from the NV3 project dated 04/23./2015. This file will need to be made common between the NV3 project code and the driver.

I changed the device id's to match the numbers suggested by Netlist. These are 0004 for 4GB, 0005 for 8GB and 0006 for 16GB cards.

I lowered the max_dms default to 31 instead of 32 to eliminate the possibility of a timeout due to a driver wrapparound condition.

I fixed a compile issue with "sgio=1".

I fixed another potential.alignment related issue
from:

```
        blk_queue_dma_alignment(card->queue, 7);
```

to

```
        blk_queue_dma_alignment(card->queue, 31);
```

ev3util: changed dp command to dump 64 entries instead of 16.
ev3util: Implemented the "status" command to decode and show NV3 registers as per Scott's script. This register access did not work for me so I could not test this code at all. Treat it as WIP.

ProcSys debug code is still in place - Can I removed this yet?

Updated version strings

v0.14

FPGA: EV3BL_20150701.jic or later

In order to be better behaved during error conditions, timeouts are now being completed to the OS with an error status. This allows the driver to be unloaded after all outstanding IOs have timed out. This is the "ERROR: module netlist_ev3 is in use" message.

DMA timeout time was lowered to 3 seconds, previously it was 60 seconds and Linux itself uses 30 seconds. The driver timeout should be lower than the Linux timeout. The only reason it was a high number was due to simultaneous DMA and PIO testing on EV1. For DMA only it should be a small number so 3 seconds was used.

test_patterns.sh script was converted to for EV3 use.

v0.13

FPGA: EV3BL_20150701.jic

I added and tested a proper chip_reset. Now a driver reload will work.

I disabled by default the capture of performance timning information during runtime, This should alleviate if not remove the kernel "oops" messages seen during high IOPs performance testing. The performance timer can be enabled and disabled using EV3UTIL. The command "perf_stats" will also report if the capture is disabled. The call that is removed is to the high performance timer called "rdtscll". This seems to be helping the oops issue and also a slight increase in IOPs seems to be happening. More testing is needed to verify that this helps. It won't hurt as the timer has no control function.

Use
"enable_stats" to see the current value of "capture_stats"
"enable_stats 1" to enable "capture_stats"
"enable_stats 0" to disable "capture_stats"

I added a driver load-time to change default the setting of capture_stats if wanted.

I updated the version strings.

ProcSys debug code is still in place.
Card memory size is still set to 8GB.

v0.12c

I fixed some 32-bit variables in ev3util that were previously being used to size the DDR3 memory and were incorrectly returning 0 due to being 32-bit.

I changed "char_driver_test" to have a background pattern of 0x44 for buffer 1 and 0x55 for buffer 2. I also dump the buffers at the end of the test. This test still uses 16K and passes with the current FPGA.

v0.12

FPGA: EV3_RTL0629FW067

Memory size is now fixed at 8 GB as opposed to 16KB of earlier versions.

Partial implementation of reset based error handling. Waiting on reset issue resolution.

Version strings have been updated.

Note:
DMA seems to be running with early testing, IOPS went down to 82.5 KIOPs and the timeout issue is still present. Previous IOPS was 202 KIOPs. The difference is between on-FPGA memory for the 16KB versions versus DDR3 off-chip memory for this 8GB version.

v0.11

Use FPGA version "EV3_20150624.jic"
Same as v0.10 plus

I enabled block device driver operation. There is no need to use "sgio=1" any more.

Now use

```
# make clean

# make
# make install
```

The block device is 16KB in size not much space to work with.

I no longer print out DEBUG2 level messages.

I added a reset on driver load. I don't think I have properly tested the reset yet. This is TBD.

This runs

205 KIOPS for 4KB READ (C2S)

187 KIOPS for 4KB WRITE (S2C)

If I use the ev3util command to limit the maximum number of outstanding DMAs to 1 the performance is still the same, thus likely indicating that the DMA engine is not currently prefetching SGLs. It seems to be doing one DMA at a time.

Judging by the current early READ IOPS for MAX_DMAS=1

then the calculated **latency is 4.88 uS**

If I leave MAX_DMAS set to the default 32 - then I get DMA timeouts after about 20 or 30 seconds of run-time. The driver does not complete timeouts this puts the driver into an "in use" state and a driver reload will not work. Thus a reboot is required. I need to change this.

The driver does not currently look at the IO context in the completion status.

On the "prd" issue. I tried this command and saw the same issue as ProcSys reported with the system crashng and rebooting. "Dazed and confused" message. The current thinking is that the completion TLP does not have a correct destination address field. Hao and I tried to hook up the PCIe analyzer in the lab but it is licensed for GEN 1 only and the system we were using did not have the capability to force GEN 1 speeds on a card. I would like to request that ProcSys generate a GEN 1 only version of the FPGA along with the normal version on the next release. Can this be done?

v0.10

Same as v0.09 plus:
Register offsets per Scott's design document updates "707-052-02-EV3-FAD-revision-mode-jun12-2015-NL.docx" are used.
Use FPGA version "EV3_20150623.jic"
FPGA card memory size is limited to 16KB.
This version has DMA operation integrated with the driver.
Basic DMA operation seems to be working using interrupts or polled mode. By default interrupt driven operation is used.

If polling is wanted then recompile setting `#define DEFAULT_ENABLE_POLLING (TRUE)`
Data integrity has a known issue believed to be in the C2S direction.

The DMA operation is integrated to the driver's structures now.
I enabled DEBUG2 level output to the log file.
Use the ev3util command called "char_driver_test" to run a simple DMA test limited to 16KB and 1KB IO sizes.
Use of the driver capabilities such as test scripts in the "tools" directory and the other ev3util commands can be attempted now. See the EV1 User Guide for details on built-in capabilities.
Continue compiling using "make sgio=1". We do not support block drivers yet.
The ProcSys debug code is still in place but may not be needed any longer.
Updated version strings.

v0.09

This includes the merge changes on v0.08 plus the items below.

1) On command "1"
Buffer 1 is filled with an incrementing QWORD pattern
Buffer 2 is filled with a 0x01 pattern

In addition these buffers are now 10x larger, 8096*10 plus 512 -- 80.5KB.

2) On command "2"
I added 2 new parameters to support filling any SGL index from 0 to 63, though only 0 through 3 will be dumped into the /var/messages file.

The new command is used as follows:

2 <number of descriptors> <starting buffer size> <buffer size increment> <direction> <sgl index>

For say the user enters the next two "2" commands:

2 4 512 128 1 0
2 4 512 64 0 1

Two SGL entries will be filled:

At index 0 we have 4 descriptors with sizes 512, 640, 768, 786 - This is a WRITE (S2C). The first descriptor starts at buffer_1 offset 0.

At index 1 we have 4 descriptors with sizes 512, 576, 640, 704 - This is a READ (C2S). The first descriptor starts at buffer_2 offset 0.

Setting the SGL at index 0 will set both the read_offset and write_offset to 0. These offsets are used to track the buffer addresses used with command "2".

In the previous example a back-to-back can be done using a WRITE followed by a READ.
Manually plugging the registers is still required.

Note that each of the two buffers keep independent offsets for their respective buffers. But two successive READs or two successive WRITEs will not use the same buffer space as the offset will increase as it is used.

If all descriptors are to be the same size use a command like 2 4 512 0 1 0 so that no bytes increment for each successive descriptor.

3) On command "3" we dump the first 4 SGL's in the array of 64. This is controlled by #define MAX_SGL_INDEX_DEBUG and can be set between 1 and 64 if needed.

4) On command "4", since the buffers are now 80.5K in size you will see a lot more data. The full buffers are dumped using this command.

5) I removed the ev3load changes.

6) Version strings were updated.

v0.08

I merged the changes provided by ProcSys
-ev3load script will exit on errors
-New ev3util command added to clear the status buffers
-Descriptor format changed to the current format of what ProcSys is currently using.
-Three new test pattern functions were added to the driver
-Version strings updated.

v0.07

I removed register access from the interrupt service routine so that only the internal counter "total_interrupts" is incremented. This allows the user to see if any interrupts were seen by using the ev3util command "log" and searching for the string "total_interrupts" at the end of the /var/log/messages file. The status buffer is also dumped when using the "log" command.

v0.06

-I fixed the printf format of

```
EV: Descriptor[0x0] System DATA PA=0x9F68C038 Card DATA PA=0x0 transfer
size=0x200
```

The card data physical address changed from 32-bits to 64-bits but the printf had not bee updated properly.

-Line 5904 had unused second index parameter - deleted
Same line - I changed %p to %llx for "page_dma" parameter to get rid of compiler warning.

I fixed the address mask from 0xfffffffffffffff8 to 0xffffffffffffff40 to ensure alignment.

I created #defines for the control bits used in the descriptors. I changed the bit definitions based on feedback. These defines start with "CONTROL_BIT".

I updated version info in the code.

This internal release has debug code for hardware bring-up, specifically the DMA engine.

Short Instructions

To compile:

Copy the tarball file into any directory

Open two terminal sessions as root. Make these as wide as possible. You will use session 1 to observe the results which are collected in the /var/log/messages file.

On session 1:

```
# tail -f /var/log/messages
```

On session 2:

```
# tar xvf netlist_ev3_v0_05.tar
# make clean
# make sgio=1          <--No block driver is compiled
# make install
# cd app
# make clean
# make
# make install
# ev3load       <--Disregard a couple of error messages related to the block driver

#  ev3util /dev/ev3map0
```

Using the new menu items

Four new debug functions were added to the ev3util menu support board bring-up. For the sake of simplicity and ease-of-typing I called these "1', "2", "3" and "4", None of these functions touches the hardware, their sole purpose is to set up the data structures needed to support DMA in both directions. The end user will use the peek/poke functions already there in order to get the hardware to do DMA.

The new menu items:

1 <no additional parameters>

Fills two DMA'able buffers with a known pattern

Buffer 1 is 8.5K and is filled with 0x22.

Buffer 2 is 8.5K and is filled with 0x33.

2 <num descriptors> <buffer size in bytes> <0 for READ | 1 for WRITE>

This fills the SGL at index 0 with up to 8 descriptors. If WRITE is selected then BUFFER 1 is used automatically. If READ is selected then BUFFER 2 is used automatically.

3 Shows the virtual and physical addresses along with data of all datra structures used. For the buffers the data displayed is the first 1024 bytes. If all the data is wanted then use "4"

4 This dumps the entire 8.5K contents of the data buffers in text format into the /var/log/messages file.

So a user will hit '1' first and fill the data buffers.

Then use '2' to fill the descriptor list.

Then use "3" to get the data that is needed to poke the DMA engine registers and start the DMA going.

Then the user pokes the DMA engine as needed.

Once the user believes that the DMA has been done use "3" and "4" to verify that the data transfer happened properly.

More details

Buffer 1 is intended for use as a WRITE (S2C) buffer

Buffer 2 is intended for use as a READ (C2S) buffer

A user can send the data from BUFFER 1 to the card, then send the data from the card to BUFFER 2 then dump both buffers and compare them against each other. The compare will need to be done visually or using a "diff' or 'cmp" utility.

All of the code that was changed for this effort is surrounded by "#ifndef PS_DEBUG". A grep for this string will yield where the work is being done.

The real work is built into the driver. The user interface is merely a transport of commands. No buffers were allocated in user space all are kernel space with the aliognments required by the EV3 design.

In the driver the menu items invoke the driver functions below. If needed they can be modified to suite a specific purpose or test case,

```
"1" init_debug
"2" setup_dma_debug
"3" show_map
"4" show_full_buffers
```

This is throwaway code and will be removed at some point. It is intended for simple test cases only. More sophisticated testing will be available when the DMA engine is integrated into the code stack.

Sample Session

EV3UTIL

```
EV3UTIL v0.5   Netlist Inc. Copyright 2011-2015, all rights reserved  -
06/12/2015 WIP


CPU Frequency: 1866.67 Mhz

EV3UTIL>1
PASSED - execution time = 0.000046 Seconds

EV3UTIL>2 5 512 1
Num Desc=0x5 Buf size=0x200 Data direction=WRITE
PASSED - execution time = 0.000020 Seconds

EV3UTIL>3
PASSED - execution time = 0.000555 Seconds

EV3UTIL>
```

/var/log/messages output

On newer Linux distributions view this using the "joutnalctl" command

Live view like "tail -f /var/log/messages"

# journalctl -f --dmesg

or

Capture the last 1000 lines for example

# journalct -n1000 --dmesg > myfile.txt

```
Jun 15 14:54:50 cherry kernel: EV: netlist_pci_probe: Vendor Id 1172 detected -
defaulting to 3.xx series card operation
Jun 15 14:54:50 cherry kernel:
Jun 15 14:54:50 cherry kernel: EV: netlist_init: netlist_init: PCI REG PASS 0
Jun 15 14:54:50 cherry kernel:
Jun 15 14:54:50 cherry kernel: EV: netlist_init: netlist_init: desc_per_page =
256
Jun 15 14:54:50 cherry kernel:
Jun 15 14:54:50 cherry kernel: EV: netlist_init: netlist_init: ev3mem block
major number = 0
Jun 15 14:54:50 cherry kernel:
Jun 15 14:54:50 cherry kernel: EV: netlist_init: netlist_init: ev3map char major
number = 247
Jun 15 14:54:50 cherry kernel:
Jun 15 14:54:56 cherry kernel: EV: FILLING PATTERN VA=0xFFFF880037188000,
pattern=0x22 byte count = 0x21A0
Jun 15 14:54:56 cherry kernel: EV: FILLING PATTERN VA=0xFFFF88009FBEC000,
pattern=0x33 byte count = 0x21A0
Jun 15 14:55:16 cherry kernel:
Jun 15 14:55:16 cherry kernel:
Jun 15 14:55:16 cherry kernel: EV: Setting up SGL, num descriptors=0x5, buffer
size(bytes)=0x200 direction=READ
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Notation: VA means virtual address  PA means
physical address
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: SGL Info next: number of SGLs supported=0x1
number of descriptors supported=0x8
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: SGL descriptor[0x0]: VA=0xffff88003757a000
PA=0x3757a000
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x0] System DATA PA=0x9FBEC000
Card DATA PA=0x0 transfer size=0x200
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A000: 00 C0 BE 9F 00 00 00 00 00 00
00 00 00 00 00 02
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x1] System DATA PA=0x9FBEC200
Card DATA PA=0x200 transfer size=0x200
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A010: 00 C2 BE 9F 00 00 00 00 00 02
00 00 00 00 00 02
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x2] System DATA PA=0x9FBEC400
Card DATA PA=0x400 transfer size=0x200
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A020: 00 C4 BE 9F 00 00 00 00 00 04
00 00 00 00 00 02
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x3] System DATA PA=0x9FBEC600
Card DATA PA=0x600 transfer size=0x200
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A030: 00 C6 BE 9F 00 00 00 00 00 06
00 00 00 00 00 02
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x4] System DATA PA=0x9FBEC800
Card DATA PA=0x800 transfer size=0x200
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A040: 00 C8 BE 9F 00 00 00 00 00 08
00 00 00 02 00 02
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x5] System DATA PA=0x0 Card DATA
PA=0x0 transfer size=0x0
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A050: 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x6] System DATA PA=0x0 Card DATA
PA=0x0 transfer size=0x0
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A060: 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Descriptor[0x7] System DATA PA=0x0 Card DATA
PA=0x0 transfer size=0x0
Jun 15 14:55:19 cherry kernel: EV: Descriptor byte dump:
Jun 15 14:55:19 cherry kernel: FFFF88003757A070: 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
Jun 15 14:55:19 cherry kernel:
```

```
Jun 15 14:55:19 cherry kernel: EV: STATUS BUFFER VA=0xFFFF88009E36A000, STATUS
BUFFER PA=0x9E36A000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A000: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A020: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A040: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A060: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A080: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A0A0: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A0C0: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel: FFFF88009E36A0E0: 0000000000000000
0000000000000000 0000000000000000 0000000000000000
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Buffer 1 VA=0xffff880037188000 PA=0x37188000
buffer size=0x21A0
Jun 15 14:55:19 cherry kernel: FFFF880037188000: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188010: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188020: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188030: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188040: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188050: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188060: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188070: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188080: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188090: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371880A0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371880B0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371880C0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371880D0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371880E0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371880F0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188100: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188110: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188120: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188130: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188140: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188150: 22 22 22 22 22 22 22 22 22 22
```

```
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188160: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188170: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188180: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188190: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371881A0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371881B0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371881C0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371881D0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371881E0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371881F0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188200: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188210: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188220: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188230: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188240: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188250: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188260: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188270: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188280: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188290: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371882A0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371882B0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371882C0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371882D0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371882E0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371882F0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188300: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188310: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188320: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188330: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188340: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188350: 22 22 22 22 22 22 22 22 22 22
```

```
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188360: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188370: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188380: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF880037188390: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371883A0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371883B0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371883C0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371883D0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371883E0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel: FFFF8800371883F0: 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22
Jun 15 14:55:19 cherry kernel:
Jun 15 14:55:19 cherry kernel: EV: Buffer 2 VA=0xffff88009fbec000 PA=0x9FBEC000
buffer size=0x21A0
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC000: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC010: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC020: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC030: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC040: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC050: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC060: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC070: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC080: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC090: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC0A0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC0B0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC0C0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC0D0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC0E0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC0F0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC100: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC110: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC120: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC130: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
```

```
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC140: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC150: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC160: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC170: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC180: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC190: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC1A0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC1B0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC1C0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC1D0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC1E0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC1F0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC200: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC210: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC220: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC230: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC240: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC250: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC260: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC270: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC280: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC290: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC2A0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC2B0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC2C0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC2D0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC2E0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC2F0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC300: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC310: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC320: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC330: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
```

```
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC340: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC350: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC360: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC370: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC380: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC390: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC3A0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC3B0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC3C0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC3D0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC3E0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
Jun 15 14:55:19 cherry kernel: FFFF88009FBEC3F0: 33 33 33 33 33 33 33 33 33 33
33 33 33 33 33 33
```

Reference Documents:

ProcSysy 707-052-01-EV3-FAD.pdf dated 04/14/2015
Netlist Software Flow Document (Internal) dated 04/27/2015

Questions for ProcSys

1) On feedback item #4 where the status was not posted. Did you set the value of the STATUS buffer address manually? The code provided does not touch the hardware.