

EVR Sharing

Dec 3, 2015

Kukhee Kim for ERMA Project Team

Evrma: EVR Modular Access Controls

- Software package including kernel driver, user space library, management tool, and epics driver.
- Abstract MRF EVR-230, EVR-300 and SLAC EVR and provide virtual EVRs
- Provide management for hardware and software resource
 - Allocate pulsegen and output channels for virtual evrs
- Provide interface for virtual EVRs
- Backward compatibility with existing event module

Software Package



- evrmaDriver – linux kernel driver
- evrma – user space library
- evrManager – resource management tool
- evrmaDoc – documents
- evrClient – epics driver/device module
(equivalent to event module)
- evrClientLab – epics ioc example

Software Stack

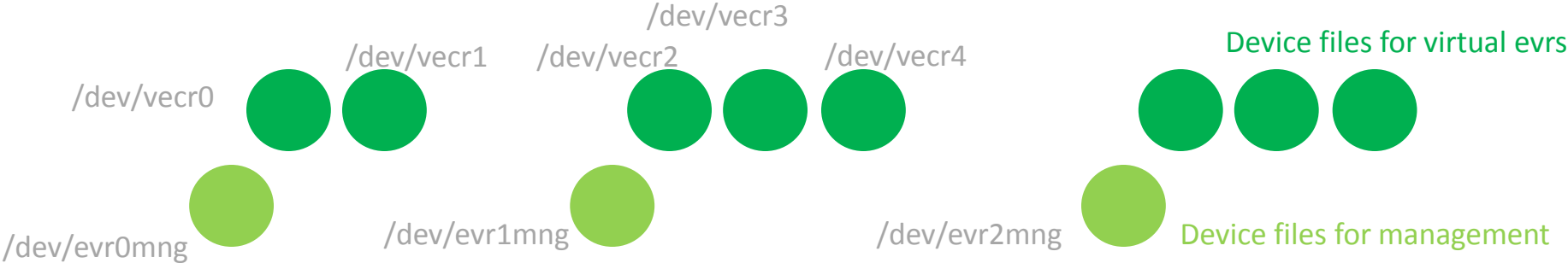


epics viocs: evrClientLab

Tool: evrManager

epics driver/device layer: evrClient

User space library: evrma



Kernel driver: evrmaDriver

EVR-230

EVR-300

SLAC EVR

Loading Kernel module



- `$IOC/<cpu directory>/kernel-modules.cmd`
- Environmental variables
 - `KERNEL_DRIVER_HOME=/afs/slac/g/lcls/package/linuxKernel_Modules`
 - `EVRMA_KERNEL_DRIVER=$KERNEL_DRIVER_HOME/evrmaDriver/<version>/<buildroot-target>`
- Module loading
 - `Insmod $EVRMA_KERNEL_DRIVER/evrma.ko`
- EVR configuration
 - `chmod 666 /dev/evr0mng` ; permission for 1st EVR module manager file
 - `cd /afs/slac/g/lcls/package/evrManager/<version>/<buildroot-target>/bin`
 - `./evrManager /dev/evr0mng init` ; initialize
 - `./evrManager /dev/evr0mng create vevr0` ; create 1st VEVR
 - `chmod 666 /dev/vevr0` ; permission for 1st VEVR
 - `./evrManager /dev/evr0mng alloc vevr0 output 0` ; allocate output channel
 - `./evrManager /dev/evr0mng alloc vevr0 pulsegen` ; allocate pulse generator
 - `./evrManager /dev/evr0mng output vevr0 0 P 0` ; assign pulse generator to output channel

EPICS Driver



- `st.cmd`
 - `eevrmaConfigure <instance> <vevrDevName>`
- Example
 - `eevrmaConfigure(0, "/dev/vevr0")`

1. How to load up kernel module
insmod <full path of kernel module>

example)

```
insmod /afs/slac/g/lcls/package/linuxKernel_Modules/evrmaDriver/R1-0-p3/buildroot-2014.08-glibc-x86_64/evrma.ko
```

```
[257863.199013] EVRMA loading in
[257863.202179] RM: hw_res_def_count=2, total resource count: 26
[257863.208077] evrma_pci_probe: vendor=0x10b5 device=0x9030 subvendor=0x1a3e subdevice=0x11e6 class=0x0 class_mask=0x0
[257863.218645] BAR0 @0xfb510000, len=0x80
[257863.222476] BAR2 @0xfb500000, len=0x10000
[257863.226578] evrma_pci_probe: start=0xfb500000, length=0x10000
[257863.232421] evrma_pci_probe to modac_mngdev_create: io_ptr=0x620000, evr_type=1
[257863.239823] OK, going on. FW_VERSION: 0x11000007, Type: 'PMC-EVR-230'
[257863.246589] RM: hw_res_def_count=2, total resource count: 23
[257863.252679] evrma_pci_probe: vendor=0x1a4a device=0x2010 subvendor=0xffffffff subdevice=0xffffffff class=0x0 class_mask=0x0
[257863.263918] BAR0 @0xfb200000, len=0x100000
[257863.268100] BAR1 @0xfb100000, len=0x100000
[257863.272277] BAR4 @0xfb300000, len=0x1000
[257863.276282] evrma_pci_probe: start=0xfb200000, length=0x100000
[257863.282334] evrma_pci_probe to modac_mngdev_create: io_ptr=0x680000, evr_type=-1
[257863.289823] OK, going on. FW_VERSION: 0x1fd2000d, Type: 'EVR-SLAC-GENERAL'
[257863.296884] RM: hw_res_def_count=2, total resource count: 24
```

MRF EVR-230

SLAC EVR

2. How to check up hardware with virtual file system

```
# ls -al /sys/class/modac-mng
```

```
total 0
drwxr-xr-x 2 root root 0 Nov 16 19:57 .
drwxr-xr-x 42 root root 0 Nov 13 20:20 ..
lrwxrwxrwx 1 root root 0 Nov 17 19:34 evr-sim-mng -> ../../devices/virtual/modac-mng/evr-sim-mng
lrwxrwxrwx 1 root root 0 Nov 17 19:34 evr0mng -> ../../devices/virtual/modac-mng/evr0mng
lrwxrwxrwx 1 root root 0 Nov 17 19:34 evr1mng -> ../../devices/virtual/modac-mng/evr1mng
```

```
# ls -al /sys/class/modac-mng/evr0mng/.
```

```
total 0
drwxr-xr-x 3 root root 0 Nov 16 19:57 .
drwxr-xr-x 5 root root 0 Nov 16 19:57 ..
-r--r--r-- 1 root root 4096 Nov 17 19:34 alloc
-rw-rw-rw- 1 root root 4096 Nov 17 19:34 dbg
-r--r--r-- 1 root root 4096 Nov 16 19:57 dev
-r--r--r-- 1 root root 4096 Nov 17 19:34 events
-r--r--r-- 1 root root 4096 Nov 17 19:34 hw_info
drwxr-xr-x 2 root root 0 Nov 17 19:34 power
-rw-rw-rw- 1 root root 4096 Nov 17 19:34 regs
lrwxrwxrwx 1 root root 0 Nov 17 19:34 subsystem -> ../../../../class/modac-mng
-rw-r--r-- 1 root root 4096 Nov 17 19:34 uevent
```

```
# cat /sys/class/modac-mng/evr0mng/hw_info
PMC-EVR-230, hw_support_hint1=1, HW: PLX, IRQ: 32,
```

pulsegen:

```
abs:0,bits:presc=16;delay=32;width=32
abs:1,bits:presc=16;delay=32;width=32
abs:2,bits:presc=8;delay=32;width=16
abs:3,bits:presc=8;delay=32;width=16
abs:4,bits:presc=0;delay=32;width=16
abs:5,bits:presc=0;delay=32;width=16
abs:6,bits:presc=0;delay=32;width=16
abs:7,bits:presc=0;delay=32;width=16
abs:8,bits:presc=0;delay=32;width=16
abs:9,bits:presc=0;delay=32;width=16
```

output:

```
OUT[0]=FP_TTL[0],MAP=63
OUT[1]=FP_TTL[1],MAP=63
OUT[2]=FP_TTL[2],MAP=63
OUT[3]=TB[0],MAP=63
OUT[4]=TB[1],MAP=63
OUT[5]=TB[2],MAP=63
OUT[6]=TB[3],MAP=63
OUT[7]=TB[4],MAP=63
OUT[8]=TB[5],MAP=63
OUT[9]=TB[6],MAP=63
OUT[10]=TB[7],MAP=63
OUT[11]=TB[8],MAP=63
OUT[12]=TB[9],MAP=63#
```


evrManager



- Syntax
 - `evrManager /dev/evrXmng command arg1 arg2`
- Commands
 - `init`
`evrManager /dev/evr0mng init` ; initialize 1st hard EVR
 - `create vdev_name`
`evrManager /dev/evr0mng create vevr0` ; create 1st vevr
`evrManager /dev/evr0mng create vevr1` ; create 2nd vevr
 - `destroy vdev_name`
`evrManager /dev/evr0mng destroy vevr1` ; destroy 2nd vevr
 - `alloc vdev_name resource [index or params]`
`evrManager /dev/evr0mng alloc vevr0 output 0` ; allocate output 0 for 1st vevr
`evrManager /dev/evr0mng alloc vevr0 output 1` ; allocate output 1 for 1st vevr
`evrManager /dev/evr0mng alloc vevr0 pulsegen` ; allocate a pulsegen for 1st vevr
`evrManager /dev/evr0mng alloc vevr0 pulsegen 16 1 1`
; allocate another pulsegen which has pre-scaler for 1st vevr
 - `output vdev_name index P source`
`evrManager /dev/evr0mng output vevr0 0 P 1` ; assign the 2nd pulsegen which has prescaler to 1st output
`evrManager /dev/evr0mng output vevr0 1 P 0` ; assign the 1st pulsegen to 2nd output

Pre-scaler and Pulsegen allocation Controls

Force to allocate a pulsegen which has pre-scaler, put non zero prescale setting

```
allocate DEV_NAME pulsegen [prescale [delay [width]]]
```

Ex)

```
./evrManager /dev/evr0mng allocate vevr0 pulsegen 4 1 1
```

; pre-scaler

```
./evrManager /dev/evr0mng allocate vevr0 pulsegen 0 1 1
```

; no pre-scaler

Document and Code Location

`/afs/slac/g/lcls/package/linuxKernel_Modules/evrmaDriver`

`/afs/slac/g/lcls/package/evrma`

`/afs/slac/g/lcls/package/evrManager`

`/afs/slac/g/lcls/package/evrmaDoc`

`/afs/slac/g/lcls/epics/R3-14-12-4_1-0/modules/evrClient`

`/afs/slac/g/lcls/epics/R3-14-12-4_1-0/iocTop/evrClientLab`

Git Repository

`/afs/slac/g/cd/swe/git/repos/linux/drivers/kernel/evrmaDriver.git`

`/afs/slac/g/cd/swe/git/repos/linux/drivers/user_space/evrma.git`

`/afs/slac/g/cd/swe/git/repos/linux/drivers/user_space/evrManager.git`

`/afs/slac/g/cd/swe/git/repos/linux/drivers/evrmaDoc.git`

`/afs/slac/g/cd/swe/git/repos/epics/modules/evrClient.git`

`/afs/slac/g/cd/swe/git/repos/epics/iocTop/evrClientLab.git`