

mbspex software for PEXOR/KINPEX

Jörn Adamczewski-Musch

GSI, Experiment Electronics: Data Processing group

EE-meeting, 02.02.2015

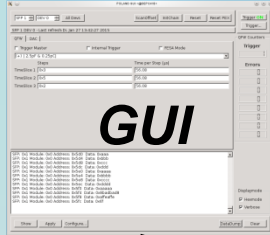
- **Motivation**
- **mbspex Linux kernel driver**
- **mbspex library**
- **usage from MBS DAQ**
- **gosipcmd tool**
- **GUIs and web interface**
- **Conclusions**

Improve PEXOR/KINPEX Linux kernel module

Goal: monitoring and configuration of optical frontends shall be possible in parallel to running MBS DAQ with separate tools (command line, batchmode, GUI)!

- function locking mechanism (semaphore) in **kernel module**: do not change registers during token/DMA readout!
- requires elementary ioctl() commands instead direct PEXOR register manipulation from MBS user readout function
- kernel module must remember gossip initialization states because it is used by several concurrent user processes
-> implicit broadcast possible!
- provide **user library** (C) with high level calls
- provide **command line tool**: generic interface for batch and GUI applications

X86 PC



```
shell>gosipcmd -z
```

DABC
webserver

tool

gosipcmd

MBS

library

libmbspex.a

kernel module

ioctl()

mmap()

filesystem

/dev/pexor0

mbspex.ko

Linux Kernel

pipe memory



PEXOR

PCIe layer
DMA

chain

mbspex kernel module features

merges previous MBS driver *pexormbs* and driver *pexor* for DABC and FESA

1. DMA send data to physical destination pointer in reserved memory (MBS "pipe" memory) instead of using internal kernel buffer pool (DABC/FESA)
2. mmap() implemented like in the *pexormbs* driver:
 - maps **memory of pexor/kinpex** => MBS can access control registers on the board (downward compatibility!)
 - maps reserved physical memory to user space => **MBS pipe memory.**
3. ioctl calls (except wait for trigger!) protected by kernel mutex
4. "atomic" **broadcast** write => put same value to same address on all slaves
5. "atomic" **frontend configuration** by single ioctl data bundle => several registers of frontend can be written at once. May also use broadcast to all slaves
6. Supports */sys/class/mbspex* nodes for register dump and debug
7. **Fully compatible with simple pexormbs driver => legacy MBS readout code works as is!**

libmbsex user space library

1. Written in C language
2. Interface to driver via file system `/dev/pexor0` and `ioctl()` calls
3. Provides high level functions:
 - PEXOR register i/o
 - DMA from PEXOR memory to PC host memory
 - gossip bus frontend register i/o (with optional slave broadcast)
 - “atomic” frontend configuration via data block (<- configuration file)
 - gossip token data request (with direct or indirect DMA)
4. Used from MBS `f_user_readout()` and by `gossipcmd` tool

MBS framework is not aware of libmbspex!

Driver access via **file operations only** (mmap(), ioctl()) => backward compatible

User code: Example f_user.c for POLAND/QFW readout

- #define USE_MBSPEX_LIB switches between mbspex library, and “old way” (local functions with direct io to mapped control registers) at compile time
- Parallel token request mode, or sequential token request (direct DMA to pipe)
- with mbspex lib and direct DMA mode: **“atomic” token data readout**
(protected against other driver calls from outside application)
- can be used as prototype for any other gossip based readout

```

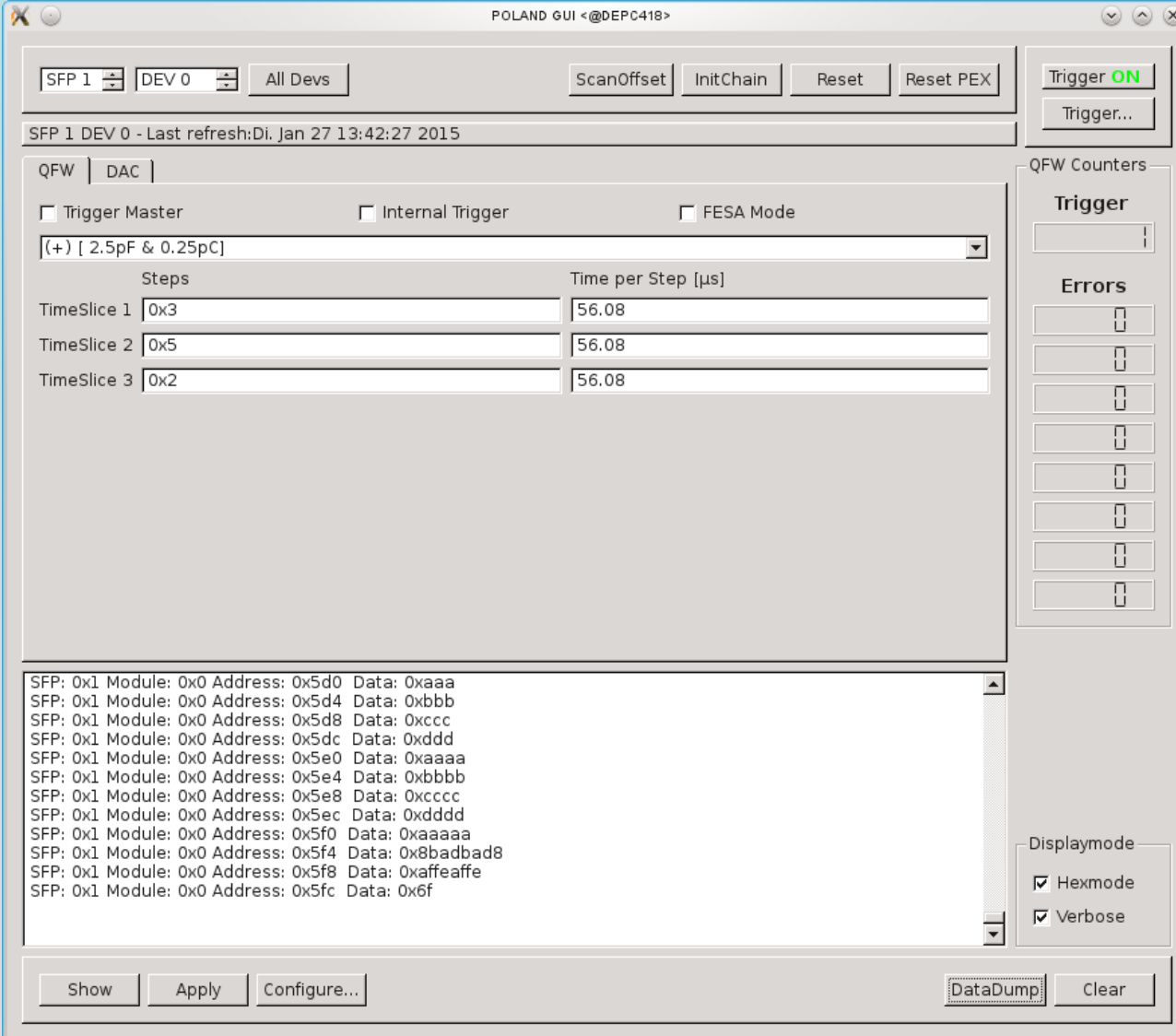
DEPC418 adamczew > bg
[1] /mbs/driv/mbspex_3.2-64_DEB/bin/PolandGui &
DEPC418 adamczew > gospicmd -h
*****
gospicmd for mbspex library
v0.42 13-Jun-2014 by JAM (j.adamczewski@gsi.de)
*****
usage: gospicmd [-h|-z] [[-i|-r|-w|-s|-u] [-b] | [-c|-v FILE] [-n DEVICE |-d|-x] sfp slave [address [value]]]
Options:
  -h          : display this help
  -z          : reset (zero) pexor/kinpex board
  -i          : initialize sfp chain
  -r          : read from register
  -w          : write to register
  -s          : set bits of given mask in register
  -u          : unset bits of given mask in register
  -b          : broadcast io operations to all slaves in range (0-sfp)(0-slave)
  -c FILE    : configure registers with values from FILE.gos
  -v FILE    : verify register contents (compare with FILE.gos)
  -n DEVICE  : specify device number N (/dev/pexorN, default:0)
  -d          : debug mode (verbose output)
  -x          : numbers in hex format (defaults: decimal, or defined by prefix 0x)

Arguments:
  sfp        - sfp chain- -1 to broadcast all registered chains
  slave      - slave id at chain, or total number of slaves. -1 for internal broadcast
  address    - register on slave
  value      - value to write on slave
  words      - number of words to read/write/set incrementally

Examples:
gospicmd -z -n 1                : master gossip reset of board /dev/pexor1
gospicmd -i 0 24               : initialize chain at sfp 0 with 24 slave devices
gospicmd -r -x 1 0 0x1000      : read from sfp 1, slave 0, address 0x1000 and printout value
gospicmd -r -x 0 3 0x1000 5    : read from sfp 0, slave 3, address 0x1000 next 5 words
gospicmd -r -b 1 3 0x1000 10   : broadcast read from sfp (0..1), slave (0..3), address 0x1000 next 10 words
gospicmd -r -- -1 -1 0x1000 10 : broadcast read from address 0x1000, next 10 words from all registered slaves (internal driver broadcast)
gospicmd -w -x 0 3 0x1000 0x2A : write value 0x2A to sfp 0, slave 3, address 0x1000
gospicmd -w -x 1 0 20000 AB FF : write value 0xAB to sfp 1, slave 0, to addresses 0x20000-0x2000F
gospicmd -w -b 1 3 0x20004c 1   : broadcast write value 1 to address 0x20004c on sfp (0..1)
gospicmd -w -- -1 -1 0x20004c 1 : write value 1 to address 0x20004c on all registered slaves (internal driver broadcast)
gospicmd -s 0 0 0x200000 0x4    : set bit 100 on sfp0, slave 0, address 0x200000
gospicmd -u 0 0 0x200000 0x4 0xFF : unset bit 100 on sfp0, slave 0, address 0x200000-0x2000FF
gospicmd -x -c run42.gos        : write configuration values from file run42.gos to slaves
*****

```

- Reset KINPEX, init SFP chains
- Read/Write any address on frontend slave
- **broadcast mode:** read/write same register to all connected slaves
- multiple words read/write
- register bit manipulation
- **configure / verify with script files *.gos**
- plain or verbose output mode



POLAND GUI <@DEPC418>

SFP 1 DEV 0 - Last refresh: Di, Jan 27 13:42:27 2015

QFW | DAC

Trigger Master Internal Trigger FESA Mode

[+] [2.5pF & 0.25pC]

Steps	Time per Step [µs]
TimeSlice 1 0x3	56.08
TimeSlice 2 0x5	56.08
TimeSlice 3 0x2	56.08

```

SFP: 0x1 Module: 0x0 Address: 0x5d0 Data: 0xaaa
SFP: 0x1 Module: 0x0 Address: 0x5d4 Data: 0xbbb
SFP: 0x1 Module: 0x0 Address: 0x5d8 Data: 0xccc
SFP: 0x1 Module: 0x0 Address: 0x5dc Data: 0xddd
SFP: 0x1 Module: 0x0 Address: 0x5e0 Data: 0xaaaaa
SFP: 0x1 Module: 0x0 Address: 0x5e4 Data: 0xbbbb
SFP: 0x1 Module: 0x0 Address: 0x5e8 Data: 0xcccc
SFP: 0x1 Module: 0x0 Address: 0x5ec Data: 0xdddd
SFP: 0x1 Module: 0x0 Address: 0x5f0 Data: 0xaaaaa
SFP: 0x1 Module: 0x0 Address: 0x5f4 Data: 0x8badbad8
SFP: 0x1 Module: 0x0 Address: 0x5f8 Data: 0xaffeaffe
SFP: 0x1 Module: 0x0 Address: 0x5fc Data: 0x6f
    
```

QFW Counters

Trigger

Errors

Displaymode

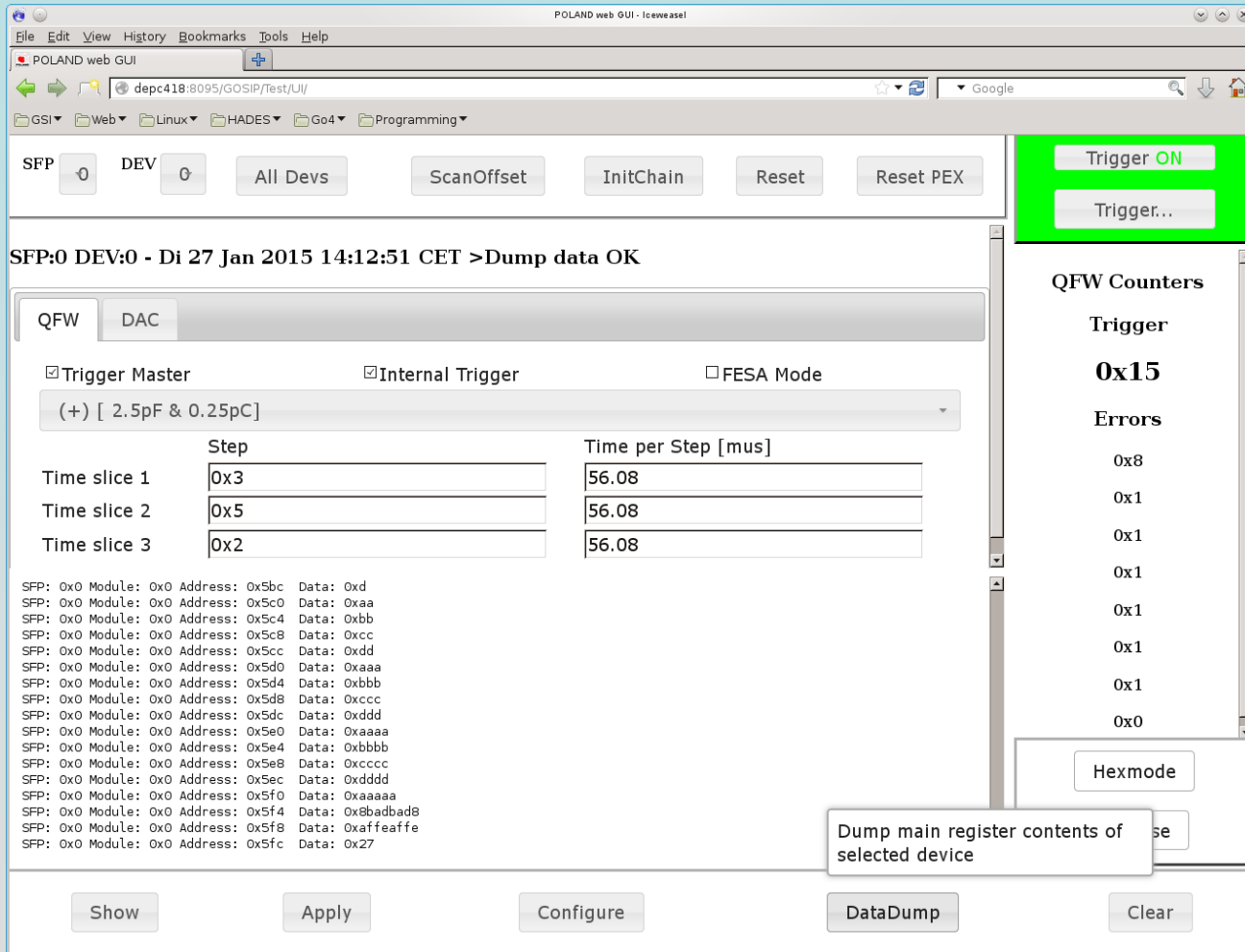
Hexmode

Verbose

Show Apply Configure... DataDump Clear

Setup of POLAND/QFW frontends for beam diagnostics

- Plain **Qt4** graphics
- uses *gosipcmd* calls via shell => decoupled from library, also works with dabc/FESA driver!
- *stdout* redirected to internal terminal
- installed at all MBS PC linuxes, call alias "poland"



Screenshot of the POLAND web GUI interface. The browser window shows the URL `depc418:8095/GOSIP/Test/UI/`. The interface includes control buttons for SFP, DEV, All Devs, ScanOffset, InitChain, Reset, and Reset PEX. A 'Trigger ON' button is highlighted in green. Below these are 'QFW' and 'DAC' tabs, and checkboxes for 'Trigger Master', 'Internal Trigger', and 'FESA Mode'. A table shows 'Time slice' settings for three steps:

Time slice	Step	Time per Step [mus]
Time slice 1	0x3	56.08
Time slice 2	0x5	56.08
Time slice 3	0x2	56.08

Below the table, a list of SFP data is shown:

```
SFP: 0x0 Module: 0x0 Address: 0x5bc Data: 0xd
SFP: 0x0 Module: 0x0 Address: 0x5c0 Data: 0xaa
SFP: 0x0 Module: 0x0 Address: 0x5c4 Data: 0xbb
SFP: 0x0 Module: 0x0 Address: 0x5c8 Data: 0xcc
SFP: 0x0 Module: 0x0 Address: 0x5cc Data: 0xdd
SFP: 0x0 Module: 0x0 Address: 0x5d0 Data: 0xaaa
SFP: 0x0 Module: 0x0 Address: 0x5d4 Data: 0xbbb
SFP: 0x0 Module: 0x0 Address: 0x5d8 Data: 0xccc
SFP: 0x0 Module: 0x0 Address: 0x5dc Data: 0ddd
SFP: 0x0 Module: 0x0 Address: 0x5e0 Data: 0xaaaa
SFP: 0x0 Module: 0x0 Address: 0x5e4 Data: 0xbbbb
SFP: 0x0 Module: 0x0 Address: 0x5e8 Data: 0xcccc
SFP: 0x0 Module: 0x0 Address: 0x5ec Data: 0xdddd
SFP: 0x0 Module: 0x0 Address: 0x5f0 Data: 0xaaaaa
SFP: 0x0 Module: 0x0 Address: 0x5f4 Data: 0x8badbad8
SFP: 0x0 Module: 0x0 Address: 0x5f8 Data: 0xaffeaffe
SFP: 0x0 Module: 0x0 Address: 0x5fc Data: 0x27
```

On the right, 'QFW Counters' are listed, with 'Trigger' set to '0x15' and 'Errors' listed as 0x8, 0x1, 0x1, 0x1, 0x1, 0x1, 0x1, 0x0. A 'Hexmode' button is also present. At the bottom, there are 'Show', 'Apply', 'Configure', 'DataDump', and 'Clear' buttons. A tooltip points to a 'Dump main register contents of selected device' button.

- DABC webserver with gossip plugin on MBS node
- javascript with jquery.ui graphics
- uses *gossipcmd* calls via dabc webserver => decoupled from library, also works with dabc/FESA driver!
- *stdout* respons redirected to browser

The screenshot shows the MBS web GUI interface with several panels:

- MBS (Green):** Contains playback and navigation controls. Below it is a terminal window showing the command `mbs> sho acq` and its output, including acquisition rates and event counts.
- Data taking (Red):** Features a filename input field (containing `test.lmd`), a size selector (set to 1000), and a star icon.
- Log modes (Yellow):** Includes a mode selector (set to `rate`) and a log history button.
- Display refresh (Green):** Has a refresh interval selector (set to 1) and a refresh button.
- Rates display (Yellow):** Shows a trending history button and a rate selector (set to 100).
- GOSIP (Blue):** Contains a `gospicmd` input field and a help icon. A red circle highlights this panel, with a red arrow pointing to the text below.
- Event Rate, Data Rate, and Server Rate (Right):** Three gauge charts showing real-time metrics: Event Rate (1), Data Rate (32.8), and Server Rate (721).
- Terminal (Bottom):** Displays the output of the `gospicmd -h` command, listing various options and their descriptions.

At the bottom of the interface, a status bar reads: **27 Jan 2015 14:05:04 CET >Start Acquisition command sent.**

a) DABC webserver with mbs/gossip plugins on MBS node

- javascript with jquery.ui graphics
- GOSIP panel for remote *gospicmd* calls
- Other controls use special sockets between MBS and webserver (mbs v6.3)

- MBS Linux device driver of PEXOR/KINPEX boards has been improved
- Applications can use driver via C library or plain file operations
- Command line tool gossipcmd:
 - Interactive monitoring and control
 - Configuration files or shell scripts
 - Interface for GUIs and webserver controls
- gossipcmd also implemented for DABC/FESA pexor driver package
- MBS DAQ runs in parallel to configuration tools
- To Do:
 - Testing and debugging with other set ups!
 - libmbspex and gossipcmd for VME Linux and LynxOS?