

MBS release V 7.1

EEL Together 27.01.2026

Jörn Adamczewski-Musch, Michael Reese, GSI/EEL
27.January 2026

MBS history and status

- MBS is general purpose DAQ system from GSI for ~30 years – [Overview talk](#)
- Recent releases: V4.3 (2004), V5.1 (2010), V6.3 (2017) [1],
V7.0 (2023) [2]
- Mostly used readout systems:
 - VMEbus + modules (GSI and other)
 - PCIe + optical fibre (gosip) + GSI front-end boards
- **New release 7.1 ready in January 2026 – [Release notes](#)**
- **Nikolaus Kurz finally retired in December 2025 –**
MBS systems will be maintained by Michael Reese and Jörn Adamczewski-Musch
Firmware developments: Shizu Minami, Ivan Rusanov

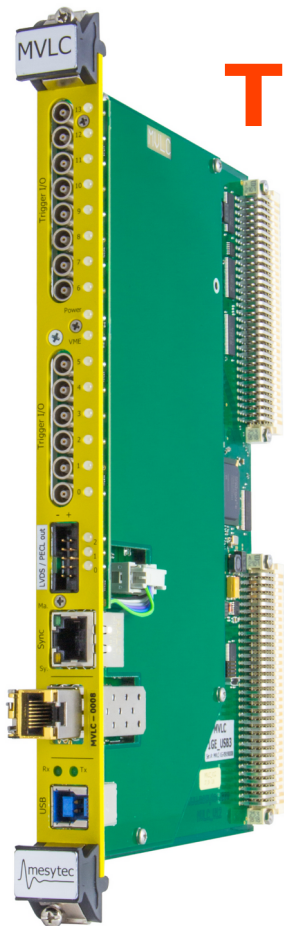
Release 7.1

- **Installed at GSI MBS cluster** at `/mbs/v71`
 - platforms: *RIO2/3/4 LynxOS; RIO4 Linux, IPV Linux, IFC Linux; X86_L64 Debian 11*
- Default MBS version (*mbslogin prod*):
please rebuild user readouts (*make clean; make;*)!!
- V7.0 is still available as *mbslogin old*
- Code repository at git.gsi.de/eel-software
NEW: MBS is open source under GPLv.3

New features

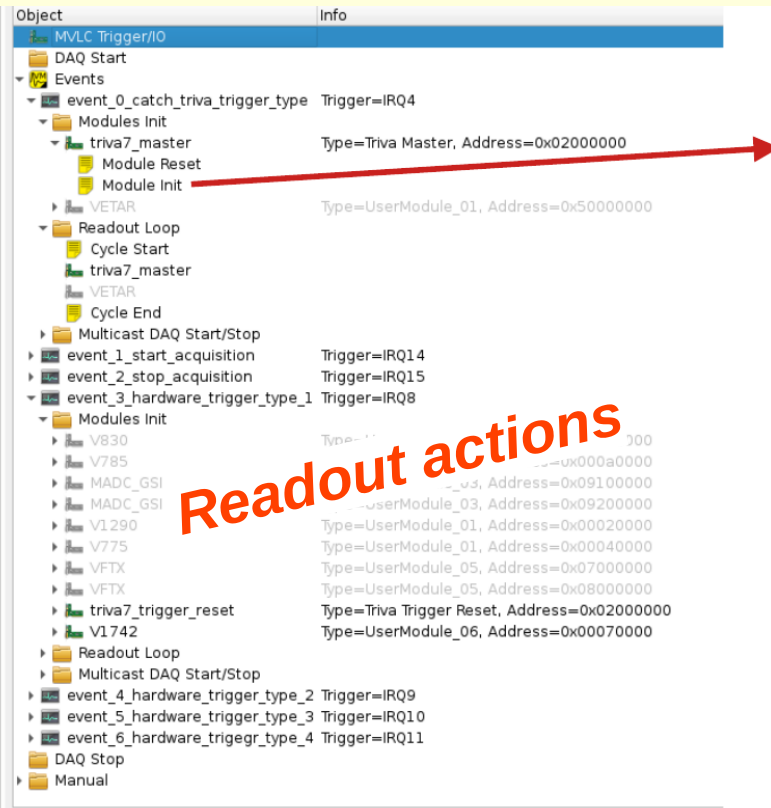
- Consolidation of readout with **mesytec MVLC VME** controller
- Software installations for **TRLOII** modules at **GSI**
- Improvements for PCIe readout with **KINPEX/GOSIP**
- **New user readout framework MURX for PCIe**
- **New user readout framework SMURV for VMEbus**

The mesytec VME controller

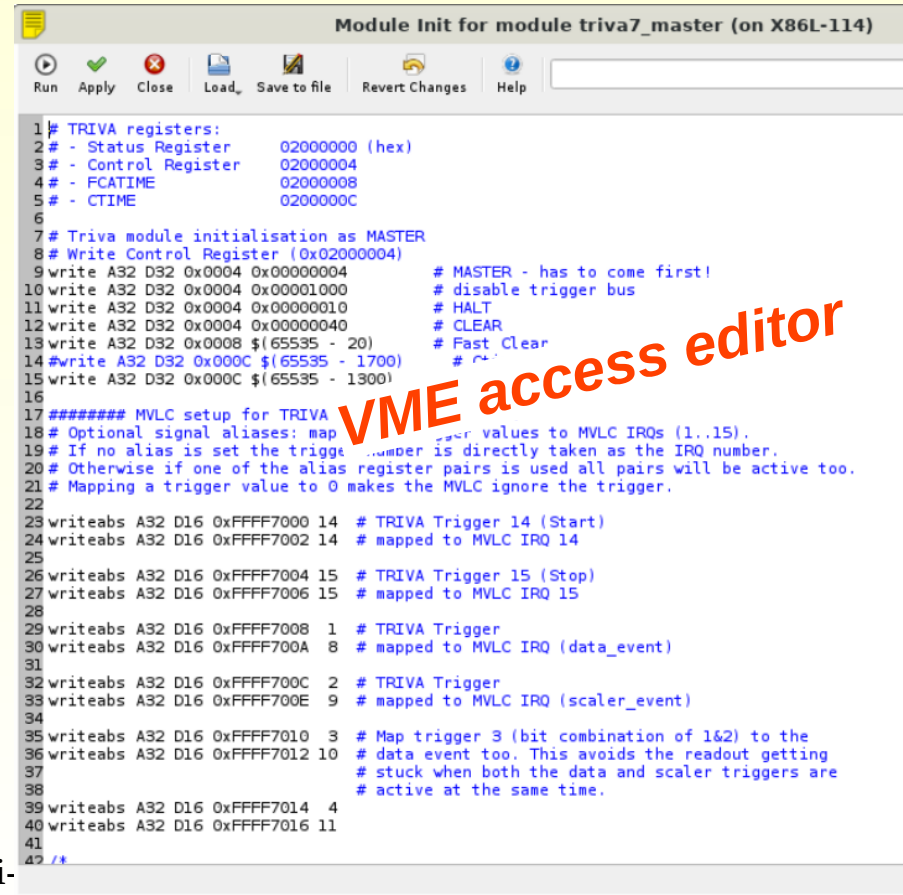


- MVLC:
 - VME controller/sequencer with **low (no) latency**
 - **Control and data transfer via USB 3** or ethernet (MBS: USB)
 - **Host can be “any” commodity MBS PC (X86L-*)**
 - Free programmable I/O
- MVME:
 - Data acquisition and “analysis” software (C++)
 - **GUI Tool to prepare initialization and readout sequences**
- **Provided by mesytec company [3]**

MVME software



Readout actions



VME access editor

MBS readout from MVLC

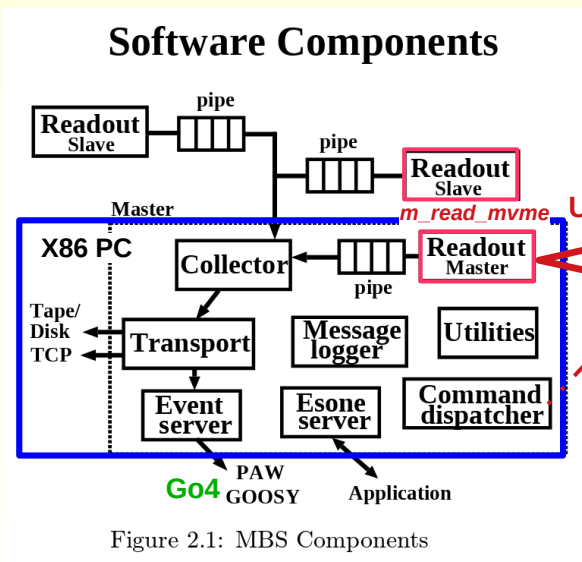


Figure 2.1: MBS Components

(from MBS user manual 2.2, 14.01.2000)

- Alternative readout process *m_read_mvme* (instead of common *m_read_meb*)
- Access to **TRIVA** registers via USB/mvme
- Reads **mvlc data stream** from USB
- **Formatting of mbs subevents** for pipe
- Formatting of **VETAR White Rabbit timestamps**
- !No user defined readout function
 - **user readout to be defined with mvme tools**

MVLC consolidation for MBS v7.1

- **Stability** improvements (experience from FRS beamtime!)
- Module **runtime control** via MBS *@script.scom* commands:
 - READ MVLC (read from any VME bus address via USB)
 - WRITE MVLC (write to any VME bus address via USB)
- Regular readout set up via **yaml files at MBS @startup**

TRLOII support with MBS v7.1

TRLOII: trigger logic + scaler firmware for VULOM modules [4]

- **Installation of TRLOII software tools** for MBS platforms
 - Standard binaries (*trlo_ctrl*,...) in PATH, additional **alias commands**:
 - *trlo_show* - print existing setup of TRLOII at standard address
 - *trlo_clear* - clear loaded setup
 - *trlo_setup* - load default setup into TRLOII
 - Tailored for GSI MBS environment at /mbs/driv/trloii/
 - **For all supported VME CPU platforms: RIO4 linux, IFC linux**
 - **New implementation: mapping TRLOII access with MVLC on X86L-** PCs (Debian 11)**
- **Example readout software** for TRLOII scaler is provided for RIO4, IFC, and MVLC systems

Refactoring of KINPEX/GOSIP firmware

(by Shizu Minami, Michael Reese, Ivan Rusanov)

- KINPEX
 - PCIe gen2 (4x 5Gb to PC host)
 - **64 bit addressing for DMA** (high memory and **larger MBS pipe**)
- GOSIP
 - flexible **data transfer frequencies** (2, 2.5, 3.15, 5, 6.25 Gb/s)
between KINPEX and upstream readout hardware

MBS v7.1 for new GOSIP firmware

Upgrade of KINPEX kernel module and library *mbspexV3*

- Select 64 bit DMA targets (additional high address word)
- Configure variable GOSIP link speed
 - Via sysfs file handle (`/sys/class/mbspex/kinpex0/linkspeed`)
 - New *gosipcmd* (aka “*goc*”) with additional parameter “-l”

Set GOSIP speed with sysfs or goc

with sysfs:

```
X86L-58 adamczew > echo "2" >> /sys/class/mbspex/kinpex0/linkspeed
```

```
X86L-58 adamczew > dmesg -T
```

```
....
```

```
[Mon Jan 12 10:32:58 2026] PEX: linkspeed will be set to preset 2 (2.5 Gb)
```

```
[Mon Jan 12 10:32:58 2026] ** pex_configure_linkspeed wants to set link speed type 2 (2.5 Gb) for SFP channel -1
```

with goc:

```
X86L-58 adamczew > goc -l 2
```

```
# set all sfp chains to linkspeed setup 2 (2.5 Gb).
```

```
..
```

mbspex driver speed specifiers:

(1) - 2.0 Gb

(2) - 2.5 Gb

(3) - 3.125 Gb

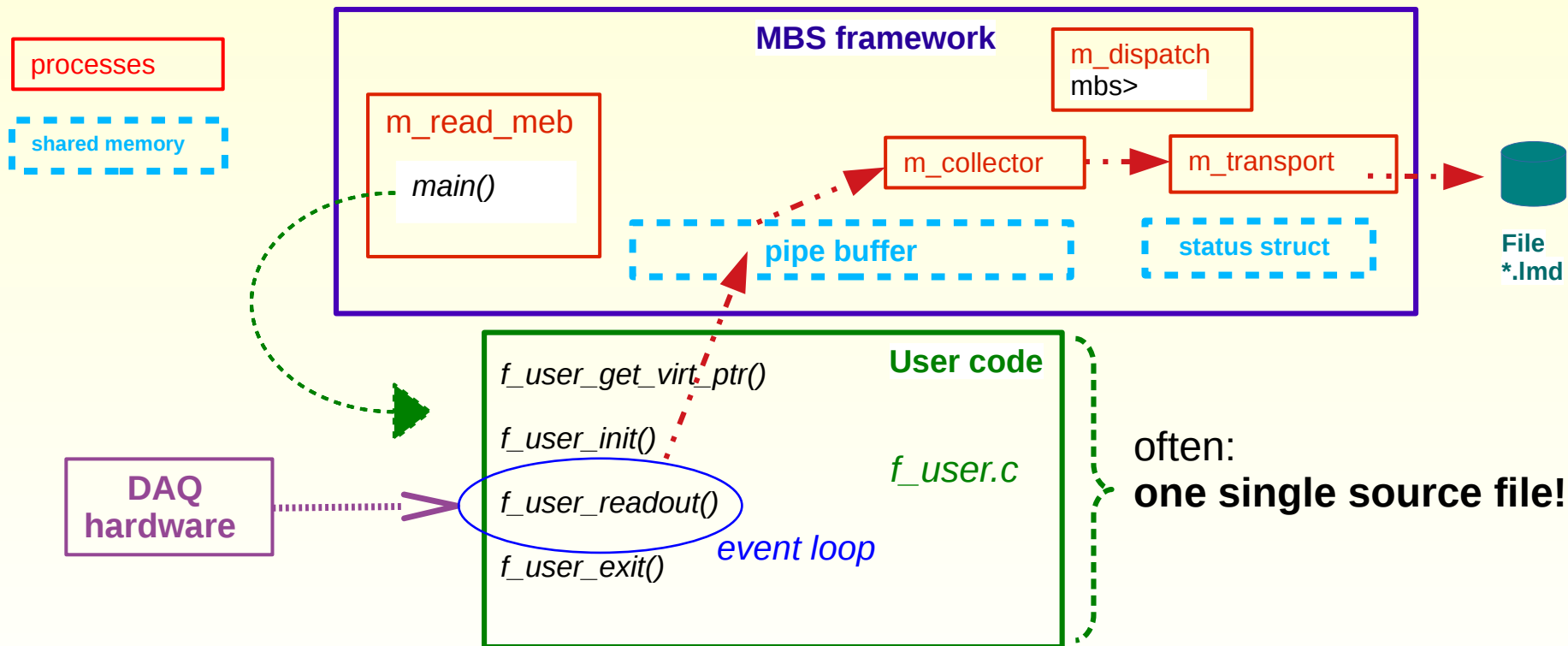
(4) - 5.0 Gb

(5) - 6.250 Gb

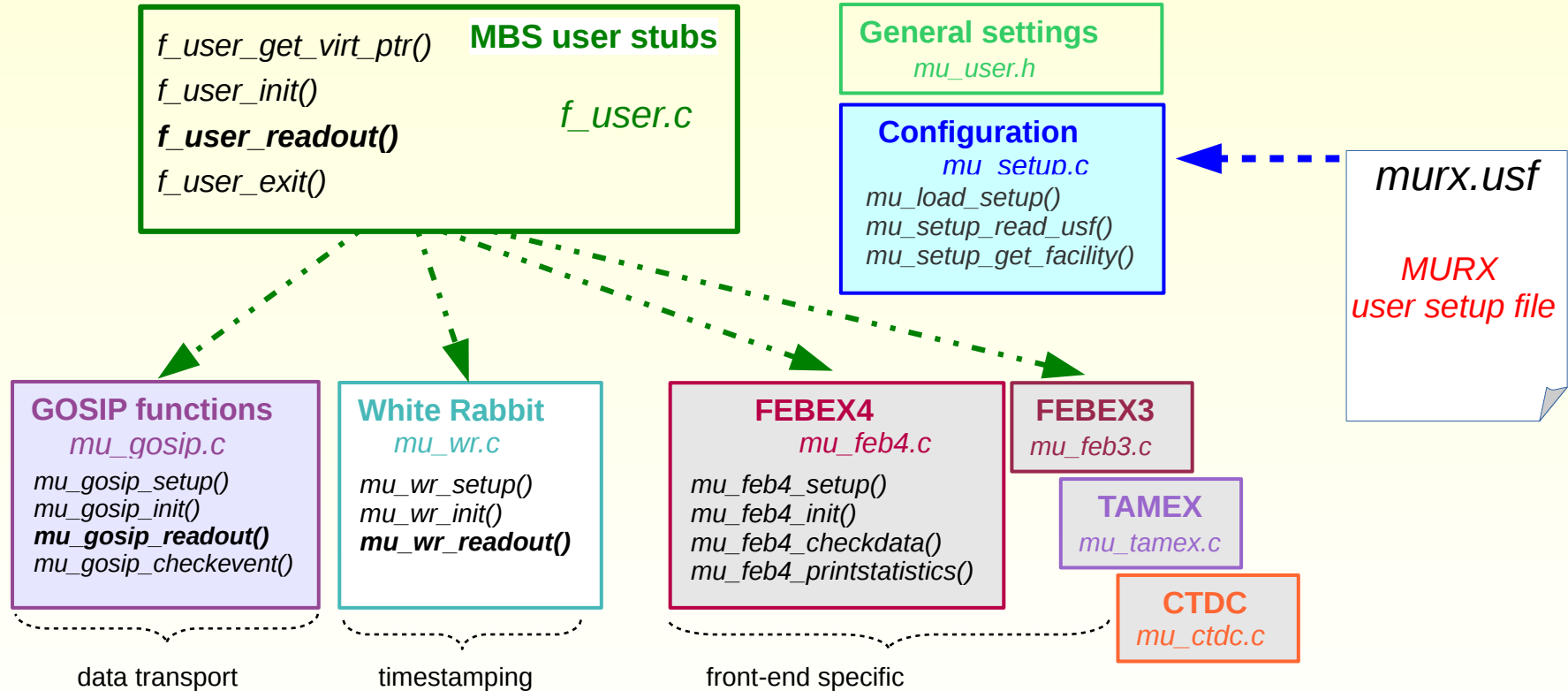
New user readout frameworks

- **MURX** (Mbs User Readout for pcieXpress systems)
with **KINPEX/GOSIP** [5]
- **SMURV** (Standard Mbs User Readout software for VMEbus)
for **VMEbus CPU** (RIO4, IPV, IFC, not MVLC!)[6]

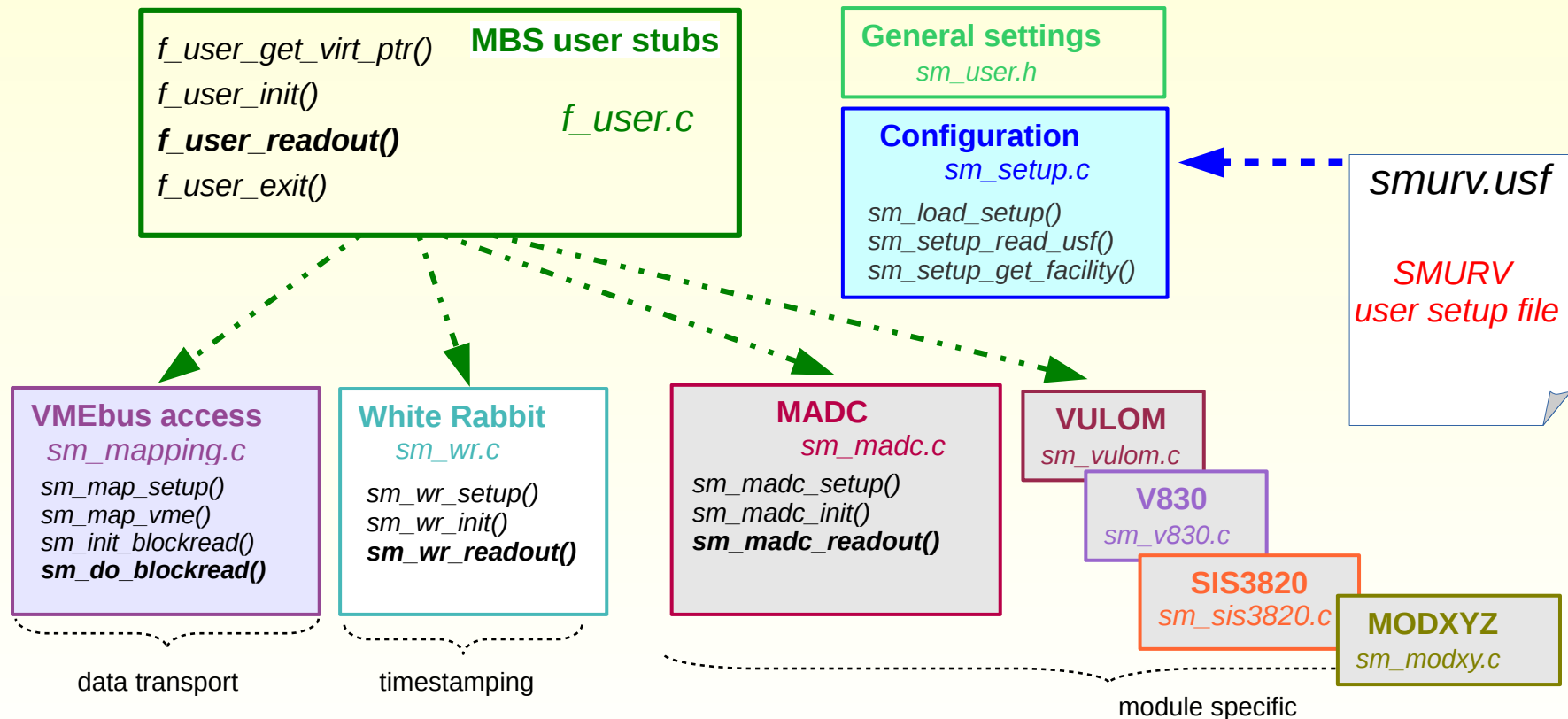
Generic MBS user readout interface



MURX: modularized source code



SMURV: modularized source code



MURX/SMURV features

- MBS user readout code is split into sourcecode modules
- Generic code is hidden from the user
 - Front-end **addressing** (GOSIP, VMEbus)
 - **Data transfer** (GOSIP, DMA)
 - **White Rabbit timestamps**
 - **Readout hardware** implementations (but: *open for new user plug-ins*)
- User defined configuration
 - **at compile time** with single sourcecode file `mu_user.h` / `sm_user.h`
 - **at runtime** with text file `murx.usf` / `smurv.usf` (**always overrides compiled values!**)

Example murx.usf (1)

```
## firstly common readout mode specs
MURX_COMMON: -
  USER_TRIG_CLEAR      = 1, -
  USE_WHITERABBIT      = 1, -
  WRITE_PARAMETERS     = 1, -
  CHECK_METADATA       = 1,-
  PRINT_STATISTIC      = 1000000

#####
MURX_GOSIP: -
# the gosip readout setup
#####
  USE_MBSPEX_LIB       = 1, -
  SEQUENTIAL_TOKEN_SEND = 1, -
  WAIT_FOR_DATA_READY_TOKEN = 1, -
  ZERO_LENGTH_DATA     = 0, -
#####
# number of slaves at each SFP chain
  NUM_SLAVES_0 = 0, -
  NUM_SLAVES_1 = 2, -
  NUM_SLAVES_2 = 0, -
  NUM_SLAVES_3 = 0, -
#####
# kind of devices at sfp chain slave positions as defined in typedef enum mu_device_kind
# MU_SLAVE_NONE = 0
# MU_SLAVE_FEB3 = 1
# MU_SLAVE_FEB4 = 2
# MU_SLAVE_TAMEX = 3
# MU_SLAVE_CTDG = 4
# MU_SLAVE_FOOT = 5
#-----
```

```
#####
SLAVE_TYPES_0 = (0, 0, 0, 0,-
                0, 0, 0, 0,-
                0, 0, 0, 0,-
                0, 0, 0, 0,-)
#####
SLAVE_TYPES_1 = (1, 1, 0, 0,-
                0, 0, 0, 0,-
                0, 0, 0, 0,-
                0, 0, 0, 0,-)
#####
SLAVE_TYPES_2 = (0, 0, 0, 0,-
                0, 0, 0, 0,-
                0, 0, 0, 0,-
                0, 0, 0, 0,-)
#####
SLAVE_TYPES_3 = (0, 0, 0, 0,-
                0, 0, 0, 0,-
                0, 0, 0, 0,-)
#####
```

```
#####
# white rabbit system setup
MURX_WHITERABBIT: -
  WR_SUBSYSTEM_ID = 0x700, -
#####
  WR_USE_TLU_DIRECT = 1, -
#####
### TLU ADDRESSES of PEXARIA for different firmwares and time latch unit types
# only required if WR_USE_TLU_DIRECT is on!
# GSI TLU is standard, ECA requires pre-configuration with saftlib at boot time
# ENIGMA , GSI TLU 0x4000100
# FALLOUT , GSI TLU 0x4000200
# ENGIMA , ECA 0x40000c0
# FALLOUT , ECA 0x4000040
  WR_TLU_ADDRESS = 0x4000100
```

Example murx.usf (2)

```
# FEBEX3/4 common parameters
```

```
MURX_FEBEX: -
```

```
# tracenlen and delay in nr of samples
```

```
    FEBEX_TRACE_LEN    = 150, -
```

```
    FEBEX_TRIG_DELAY   = 20,-
```

```
# data filter control
```

```
# (0x80 E,t summary always + data trace) always
```

```
# (0x82 E,t summary always + (data trace + filter trace) always
```

```
# (0x84 E,t summary always + data trace) if > 1
```

```
# (0x86 E,t summary always + (data trace + filter trace) if > 1
```

```
##
```

```
    FEBEX_DATA_FILTER_CONTROL = 0x80, -
```

```
## sfp_port, module_id of the module to distribute clock, 0xff
```

```
    FEBEX_CLK_SOURCE_ID =(0xFF, 0),-
```

```
# generic trigger hit finder
```

```
# for 12 bit 8, 4 ,9 (8+1); for 14 bit 14, 4, 15 (14 + 1).
```

```
    FEBEX_TRIG_SUMA    = 8,-
```

```
    FEBEX_TRIG_GAP     = 4,-
```

```
    FEBEX_TRIG_SUMB    = 9,-
```

```
# energy filter (trapez mode parameters)
```

```
    FEBEX_ENERGY_SUMA  = 6,-
```

```
    FEBEX_ENERGY_GAP   = 3,-
```

```
    FEBEX_ENERGY_SUMB  = 7,-
```

```
#####
```

```
### FEBEX3/4 threshold section
```

```
###
```

```
MURX_FEBEX_THRESH: -
```

```
# all channels of all slaves at 4 spfs
```

```
# max value 0x1FF 255 (adc counts)
```

```
#####
```

```
###SFP 0 for slaves 0...F
```

```
#
```

# channel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FEBEX_THRES_0_0	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_1	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_2	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_3	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_4	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_5	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_6	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_7	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_8	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_9	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_A	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_B	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_C	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_D	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_E	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															
FEBEX_THRES_0_F	= (0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff, 0x1ff),-															

```
#####
```

```
###SFP 1 for slaves 0...F
```

```
#
```

# channel	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FEBEX_THRES_1_0	= (0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f, 0x03f),-															
FEBEX_THRES_1_1	= (0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff, 0x0ff),-															

27.01.2026

MURX/SMURV how to use

- 1) **Clone (or download) the framework** once to a user working directory:
 - “git clone <https://git.gsi.de/eel-software/murx.git> .”
 - “git clone <https://git.gsi.de/eel-software/smurv.git> .”
- 2) **Enable MBS environment:** “*mbslogin prod*” or “*mbslogin v71*”
- 3) **Compile the binaries:** “*make clean; make all*”
(rebuild is required every time when platform is changed!)
- 4) **Adjust *setup.usf*** to current readout platform [RIO2/3/4 Lynx, RIO4 Linux, IFC, IPV, X86 Linux]
(SMURV: examples *setup.usf.XY* provided. MURX: only X86L- are used)
- 5) **Adjust *murx.usf*** (or ***smurv.usf***) for DAQ configuration
- 6) **Run mbs** and start daq: @startup.scom

Summary

- MBS release v7.1 is installed at GSI, open source now
- Auxiliary software for TRL0II adjusted and installed
- VMEbus readout with MVLC controller consolidated
- Firmware improvements for PCIe - GOSIP provided
- New user readout frameworks MURX and SMURV

Thank you!



The MBS team in Dresden August 2024

27.01.2026

J.Adamczewski-Musch, M.Reese

References

- [1] J.Adamczewski-Musch, N.Kurz, S.Linev, GSI Data Acquisition System MBS Release Notes V6.3,
Available at https://www.gsi.de/fileadmin/EE/MBS/gm_mbs_rel_63.pdf
- [2] J.Adamczewski-Musch, N.Kurz, S.Linev, GSI Data Acquisition System MBS Release Notes V7.0,
Available at https://www.gsi.de/fileadmin/EE/MBS/gm_mbs_rel_70.pdf
- [3] mesytec GmbH Co. KG , mvme - mesytec VME Data Acquisition documentation,
Available at <https://www.mesytec.com/products/nuclear-physics/MVLC.html>
- [4] The TRLOII firmware repository, available at <https://git.chalmers.se/expsubphys/trloii.git>
- [5] MBS User Readout for PCIe systems framework *MURX*, available at <https://git.gsi.de/eel-software/murx>
- [6] Standard MBS User Readout for VMEbus framework *SMURV*, available at <https://git.gsi.de/eel-software/smurv>

Bonus slides

MVLC with MBS “how to”

1.Configure MVLC with MVME GUI

- It is available as alias “mvme” on all MBS PC hosts for v7.1.
- Example readout stacks for all VME modules are provided by EEL/DAQ group

2.Export setup to yaml text file

(menu “**File:/mesyte-mvlc library import/export → mysetup.yaml**”)

3.Exit MVME GUI

4.Startup MBS with settings:

- In setup.usf: MASTER=19
- in startup.scom: **m_read_mvme** instead m_read_meb.
- in startup.scom: **initialize mvlc from yaml file: x “/mbs/driv/mvme/init_mvlc.sh ./mysetup.yaml ”**
- Command “START MVLC” to start acquisition.

Summary MVLC with MBS

- All VME modules used at GSI proved to work
- Performance improvement of factor 3 compared with RIO4 readout for a typical FRS setup
- Performance improvement of factor 5 compared with RIO4 for 2eSST readout of CAEN v1742 (150 MB/sec)
- Switching from pure MVME to MBS DAQ in a few seconds
(Use *mvme* to load MVLC sequences for MBS)
- No other hardware changes required when replacing RIO4 with MVLC
- MBS with MVLC controller works as standalone, master or slave TRIVA trigger MBS sub(system)

Check GOSIP speed with sysfs

X86L-58 adamczew > **cat /sys/class/mbspex/kinpex0/linkspeed**

```
***** PEX current GTX linkspeed *****
**** SFP channel 0 ****
*** PEX gtx register setup:
channel: 0
PEX_DRP_GTX_CPLL (0x5e): 0x1002
PEX_DRP_GTX_TXRX_DIV (0x88): 0x0011
PEX_DRP_GTX_RXCDR_CFG (0x88): 0x1020
PEX_DRP_GTX_QPLL (0x36): 0x0020
*** PEX gtx parameter setup:
QPLL_FBDIV: 0x20
CPLL_REFCLK_DIV 0x10 -> M = 1
CPLL_FBDIV_45 0x0 -> N1 = 4
CPLL_FBDIV 0x2 -> N2 = 4
TXOUT_DIV 0x1 -> D = 2
RXOUT_DIV 0x1 -> D = 2
Reference clock is set to 125 MHz
=> Line speed will be: 2 * 125 MHz * N1 * N2 / (M * D)
CPLL frequency 4000 MHz
=> Tx speed 2000 MHz
=> Rx speed 2000 MHz
*****
*** SFP channel 1 ****
*** PEX gtx register setup:
channel: 1
PEX_DRP_GTX_CPLL (0x5e): 0x1002
PEX_DRP_GTX_TXRX_DIV (0x88): 0x0011
PEX_DRP_GTX_RXCDR_CFG (0x88): 0x1020
PEX_DRP_GTX_QPLL (0x36): 0x0020
*** PEX gtx parameter setup:
QPLL_FBDIV: 0x20
.....
```

```
....
*****
*** PEX mmcm register setup:
PEX_DRP_MMC_CLKOUT0_0 (0x8): 0x1187
PEX_DRP_MMC_CLKOUT0_1 (0x9): 0x0080
PEX_DRP_MMC_CLKBUFOUT_0 (0x14): 0x134D
PEX_DRP_MMC_CLKBUFOUT_1 (0x15): 0x0000
PEX_DRP_MMC_DIVCLK (0x16): 0x2083
PEX_DRP_MMC_LOCK_1 (0x18): 0x0177
PEX_DRP_MMC_LOCK_2 (0x19): 0x7C01
PEX_DRP_MMC_LOCK_3 (0x1a): 0xFFE9
PEX_DRP_MMC_FILT_1 (0x4e): 0x0908
PEX_DRP_MMC_FILT_2 (0x4f): 0x1000
*** PEX mmcm parameter setup:
CLKOUT0:
HIGH TIME: 6
LOW TIME: 7
EDGE : 1
CLKBUFOUT:
HIGH TIME: 13
LOW TIME: 13
EDGE : 0
DIVCLK:
HIGH TIME: 2
LOW TIME: 3
EDGE : 1
lock table: 0xffd77fa401
filter table: 0xd0
*****
```

For parameters' meaning see:
https://docs.amd.com/v/u/en-US/ug476_7Series_Transceivers
https://docs.amd.com/v/u/en-US/ug472_7Series_Clocking
https://docs.amd.com/v/u/en-US/xapp888_7Series_DynamicRecon

Example smurv.usf

```
## firstly common readout mode specs
```

```
SMURV_COMMON: -
```

```
USER_TRIG_CLEAR      = 1, -
USE_WHITERABBIT      = 0, -
PRINT_STATISTIC      = 1000000
```

```
*****
```

```
# white rabbit system setup
```

```
SMURV_WHITERABBIT: -
```

```
WR_SUBSYSTEM_ID      = 0x600, -
```

```
#-----
WR_USE_TLU_DIRECT    = 1, -
```

```
#-----
### TLU ADDRESSES of VETAR for different firmwares and time latch unit types
```

```
# only required if WR_USE_TLU_DIRECT is on!
```

```
# ENIGMA , GSI TLU 0x2000100
```

```
# FALLOUT , GSI TLU 0x2000200
```

```
WR_TLU_ADDRESS       = 0x2000200
```

```
*****
```

```
# VMEbus mapping
```

```
#--- for special mapping modes on ifc/ipv/old rio- this should agree with
```

```
LOC_MEM_BASE of setup.usf
```

```
SMURV_MAPPING: -
```

```
VME_BASE             = 0x7000000,-
```

```
VME_SIZE             = 0x1500000
```

```
***** READOUT MODULES: *****
```

```
#--- The <MODULE>_OFF gives the address offset above the VME_BASE aka LOC_MEM_BASE as specified in setup.usf
```

```
#--- for tested default setup, VME_BASE was assigned to 0x7000000
```

```
***** BLOCK_MODE values:
```

```
#- 0 off (use single cycle)
```

```
#- 1 BLT
```

```
#- 2 MBLT
```

```
#- 3: 2eVME,
```

```
#- 4: 2eVME fast
```

```
#- 5: 2eSST 160 MBytes/sec
```

```
#- 6: 2eSST 233 MBytes/sec
```

```
#- 7: 2eSST 320 MBytes/sec
```

```
*****
```

```
#-----
```

```
SMURV_MADC: -
```

```
NUM_MADC             = 0,-
```

```
MADC_OFF             = 0x500000
```

```
#-----
```

```
SMURV_MDPP: -
```

```
NUM_MDPP             = 0,-
```

```
MDPP_OFF             = 0x100000
```

```
#-----
```

```
#-----
SMURV_VULOM: -
```

```
NUM_VUL_SCAL         = 1,-
```

```
VULOM_OFF            = 0xF00000,-
```

```
BLOCK_MODE           = 2,-
```

```
USE_SLIM_WINDOW       = 1,-
```

```
USE_SOFTWARE_LATCH    = 1
```