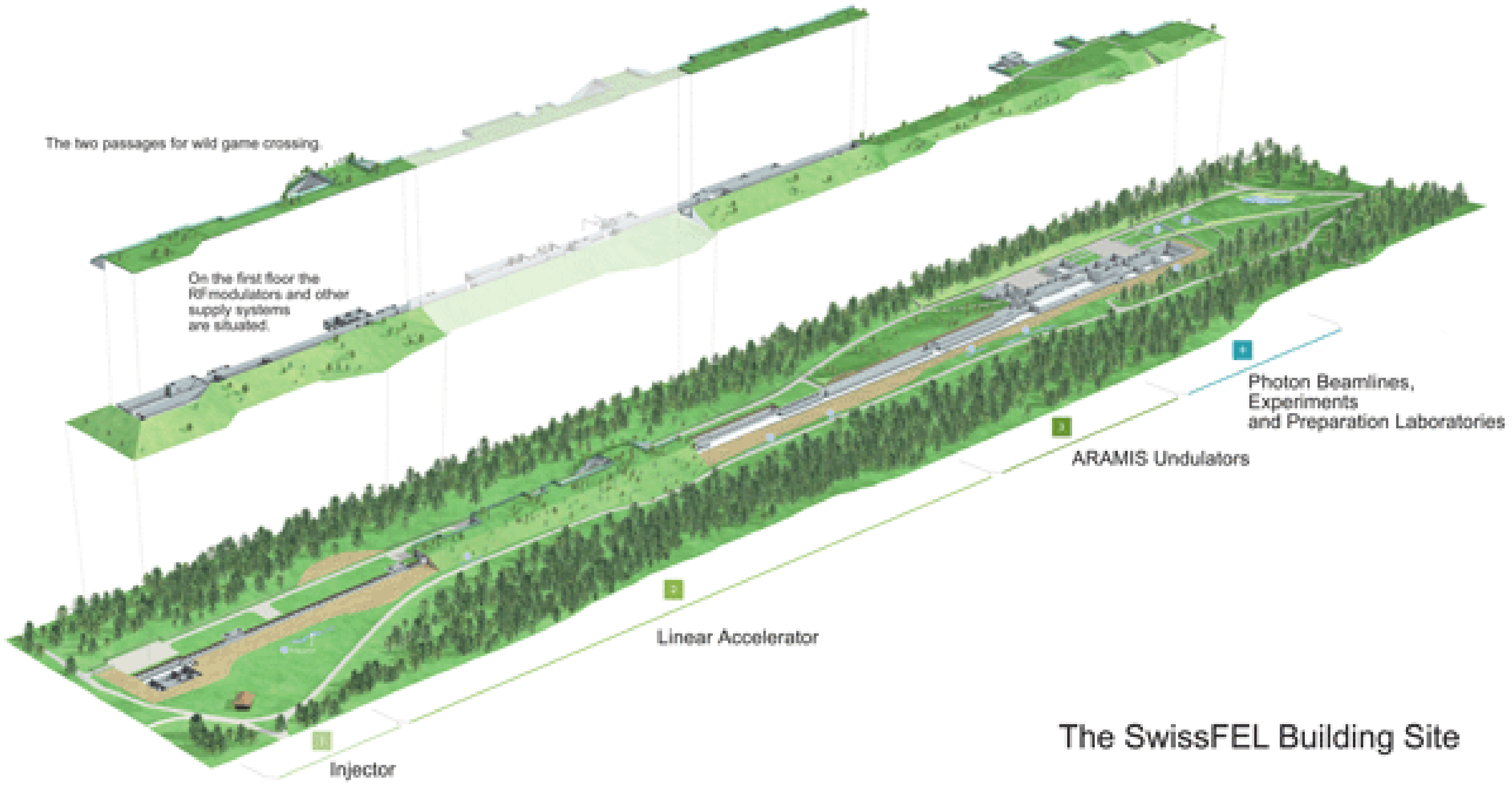
PAUL SCHERRER INSTITUT

Dragutin Maier-Manojlovic :: Embedded Systems :: Paul Scherrer Institut

EtherCAT at PSI

EPICS Collaboration Meeting 2016 – Lund, Sweden

SwissFEL


The two passages for wild game crossing.
On the first floor the RF modulators and other supply systems are situated.
Photon Beamlines, Experiments and Preparation Laboratories
ARAMIS Undulators
Linear Accelerator
Injector

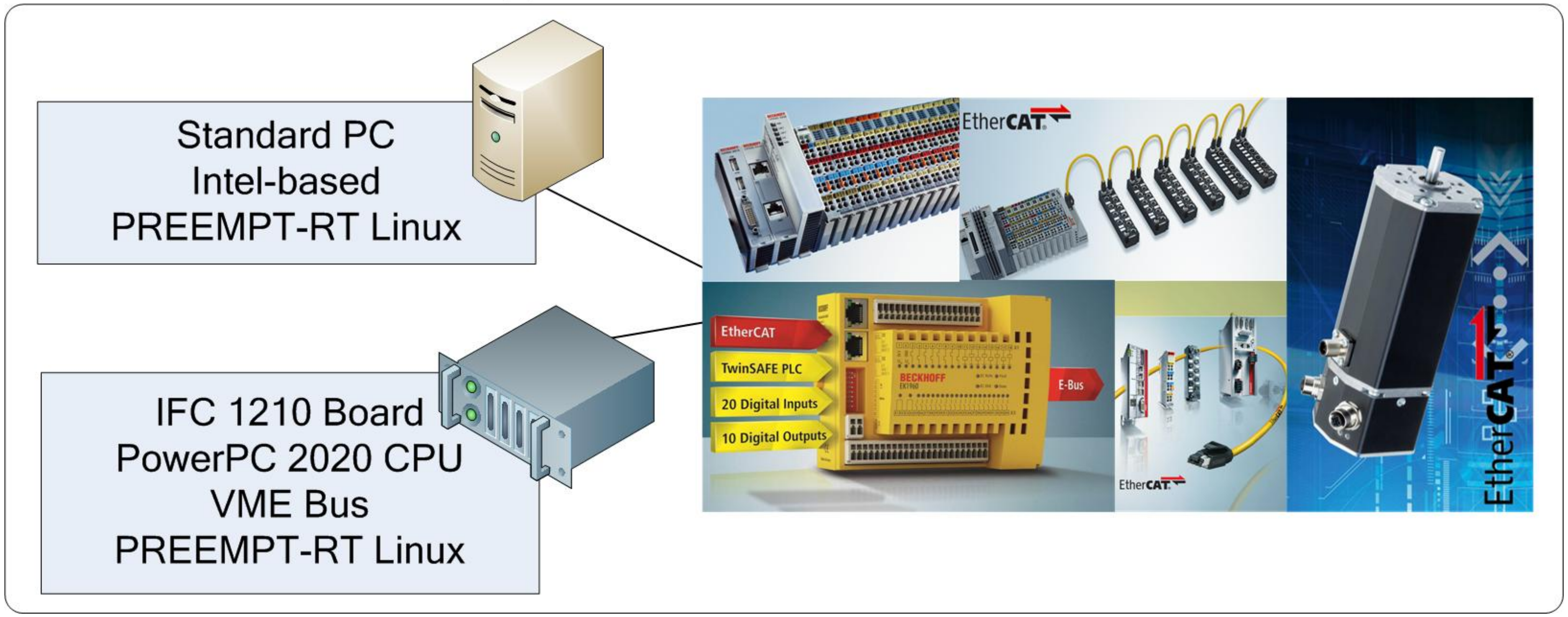
The SwissFEL Building Site

Swiss Free Electron Laser SwissFEL

EtherCAT at PSI

At PSI, EtherCAT devices are connected using Beckhoff TwinCAT and Beckhoff IPCs, but also using IgH Etherlab Master (somewhat modified) and in-house EPICS drivers and tools on various platforms – Standard Intel-based PCs for SoftIOC Servers, and Embedded solutions using standard crates and Ioxos IFC 1210 boards.

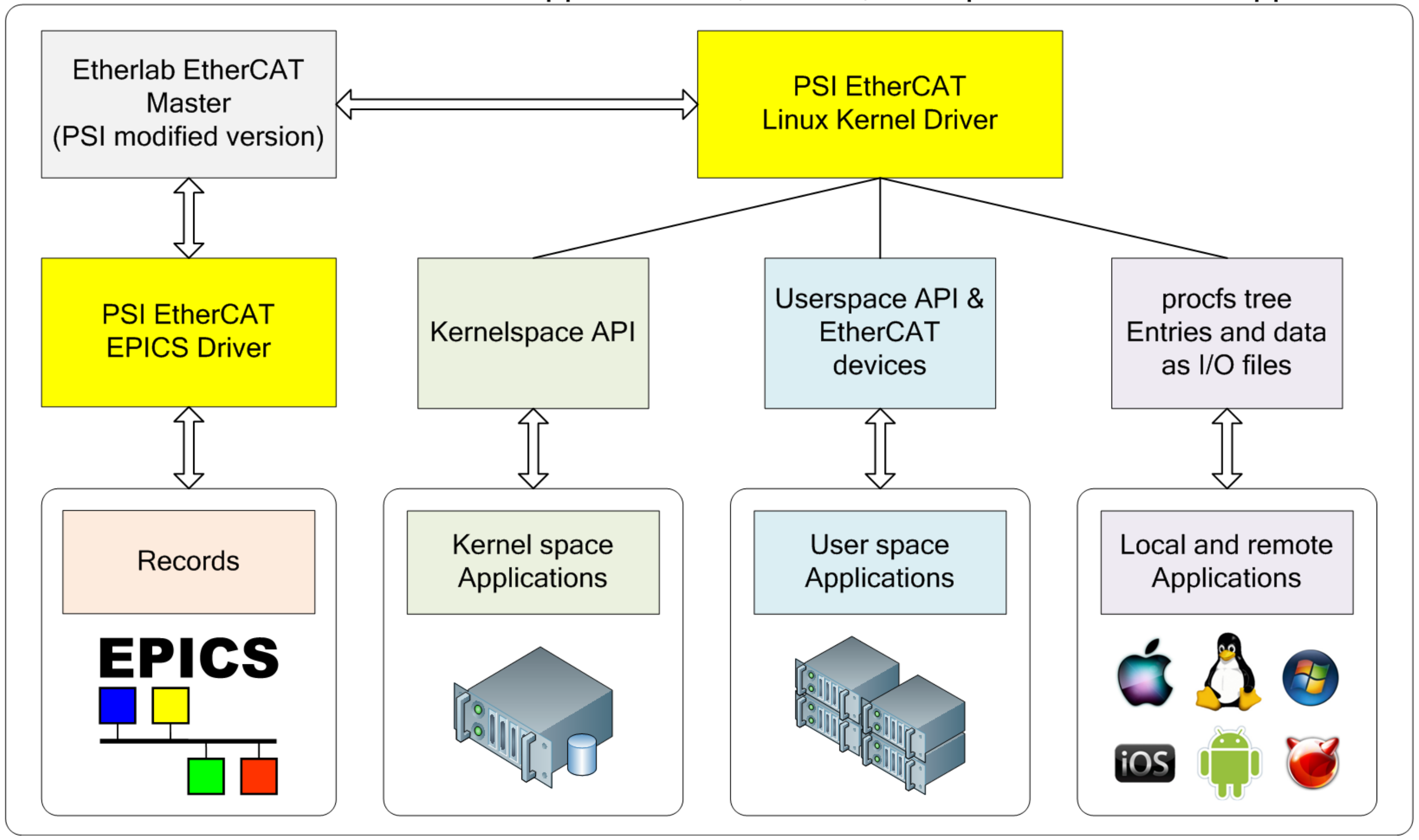
Overview - PSI EtherCAT support can be used on standard PCs or IFC 1210 Boards



EtherCAT at PSI

EtherCAT driver has two incarnations, userspace based EPICS driver, and Linux kernel driver.

Structure - PSI EtherCATdrivers support EPICS, kernel-, userspace and remote applications



Etherlab EtherCAT Master (PSI modified version)
PSI EtherCAT Linux Kernel Driver
PSI EtherCAT EPICS Driver
Kernelspace API
Userspace API & EtherCAT devices
procfs tree Entries and data as I/O files
Records
EPICS
Kernel space Applications
User space Applications
Local and remote Applications
iOS

EtherCAT at PSI

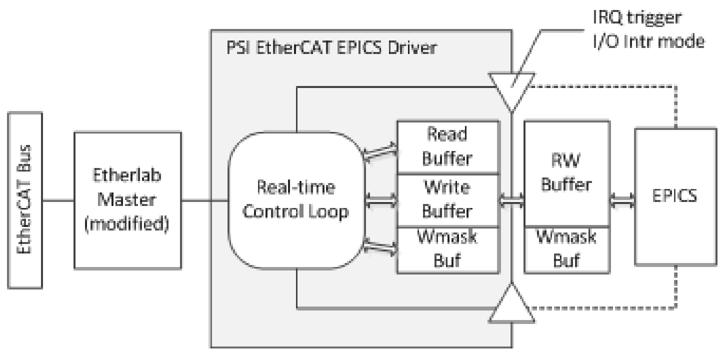
PSI EPICS EtherCAT driver and PSI general EtherCAT driver.

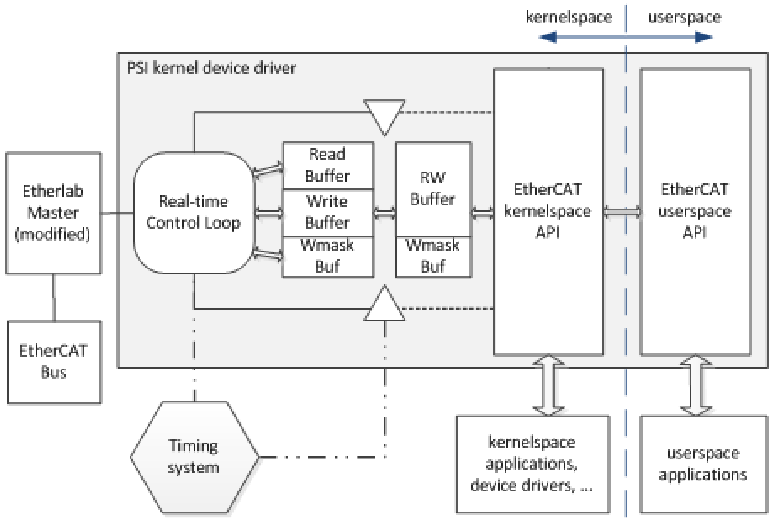
Both PSI EtherCAT and Linux drivers support **automatic scan and configuration** of EtherCAT Modules. There is no need to prepare Beckhoff or other XML configuration files or perform any other setup in advance.

Modules can also be configured **statically, either from EPICS or any other application or from the CLI.** The static configuration will work even if the modules are not reporting their own configuration (SII missing or wrong).

Optimizations allow for very high EtherCAT rates (up to 5-10 kHz and more on P2020-based IFC, with minimal CPU load (<10% CPU used at 8-10 kHz!), using PREEMT-RT on Ioxos IFC 1210. Running very stable 24/7 for weeks on end, no dropped packets, less than 0.01% of packets with jitter/microdelays, with 1000+ EPICS records set to I/O Intr scan.

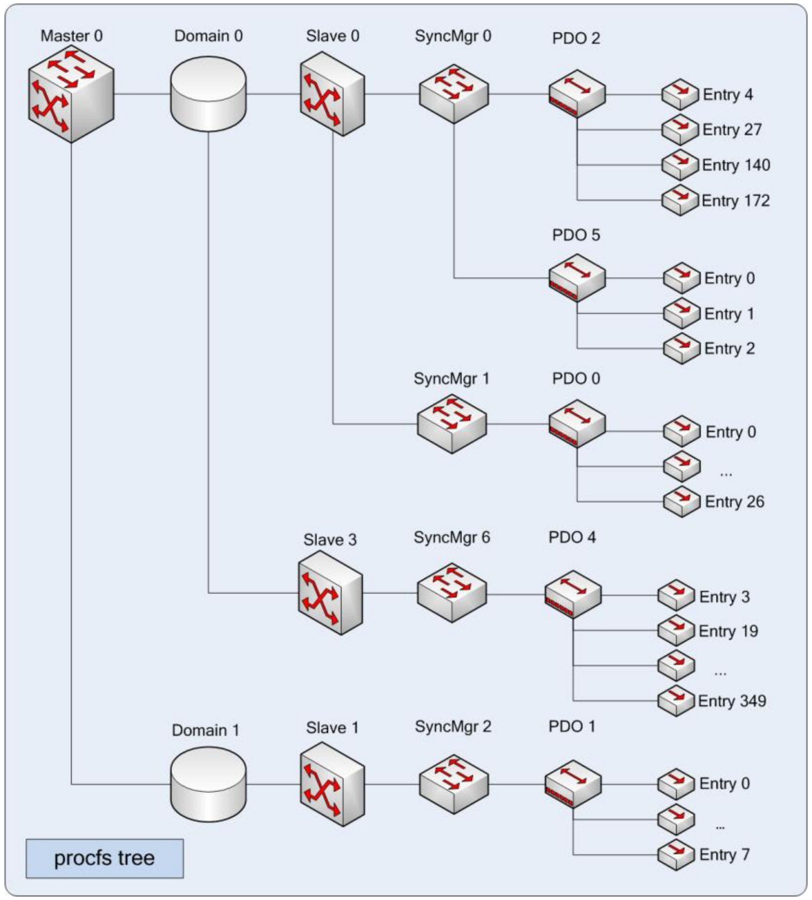
No dependency on other drivers (i.e. asyn) or software, aside from Etherlab Master itself.

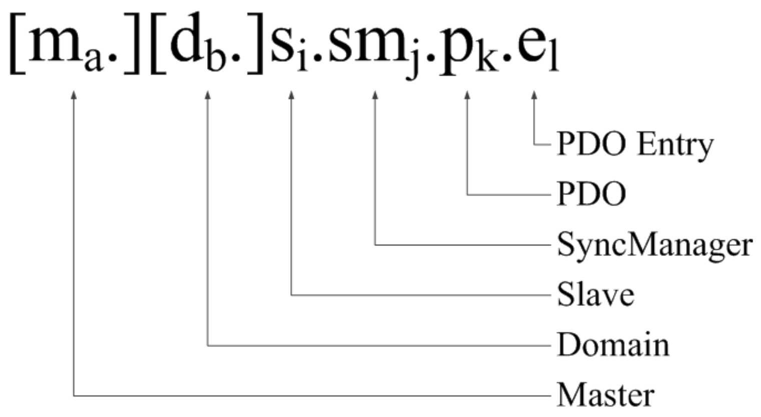




EtherCAT at PSI

To make EtherCAT support at PSI as flexible as possible, special adressing was devised.





Examples (INP/OUT addressing):

s0.sm2.p0.e0
s1.sm0.p14.e22
s5.sm1.p1.e3

EtherCAT at PSI

To increase flexibility, additional adressing modifiers and modes were introduced, such as: .o, .b, r, .lr, .l, as well as type override (t=<newtype>)

EPICS Control System - Full EPICS support and many useful addressing and type-conversion extensions

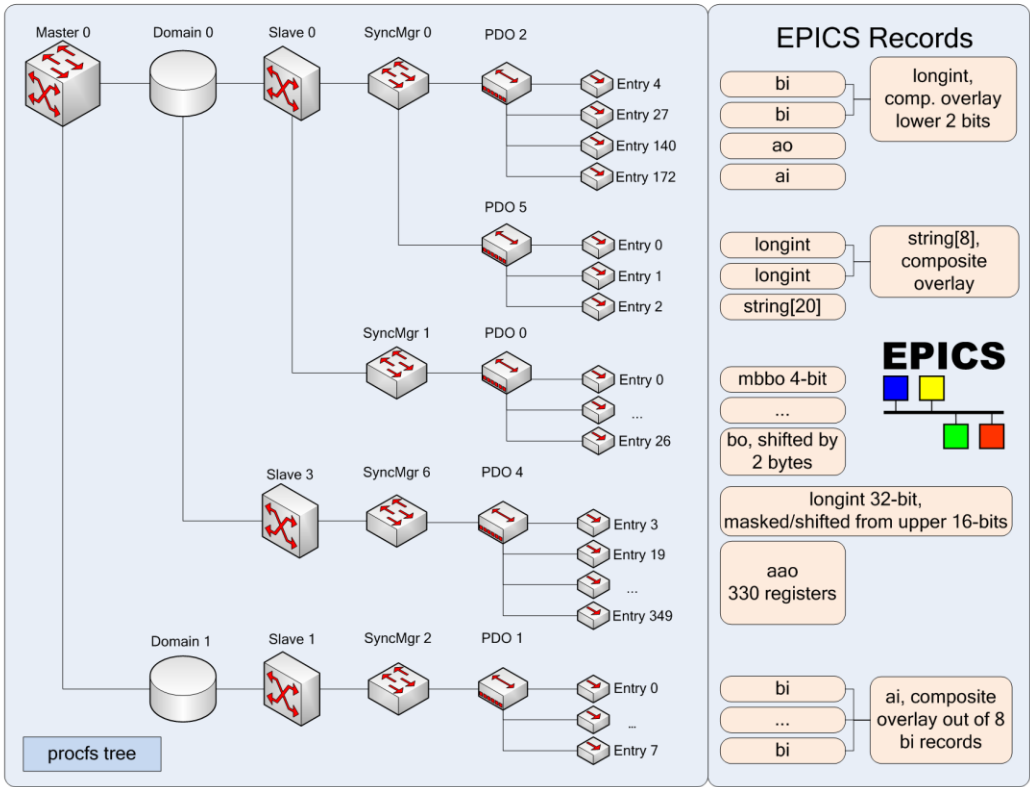
PSI EtherCAT EPICS driver supports all standard EPICS record types, and allows for interrupt-driven, real-time triggering of read/write operations for PDO Entries.

New PSI addressing style (*[Master.][Domain.]Slave.SM.PDO.Entry*, i.e. s3.sm4.p0.e136), as well as a lot of addressing extensions and modes are available:

- **forced offset** (.onnn, e.g. s0.sm0.p0.e0.o4 for +4 offset)
- **forced bit extraction** (.bnnn, e.g. s0.sm0.p0.e0.b3 for bit 3 of that entry)
- **domain register addressing** (r0-rnnn) for relative addressing
- **slave register addressing** for relative addressing of entries inside a slave module (.lrnnn) on any level, from SM to PDO to Entry
- **support for string in/out records** for string extraction from anywhere in the Domain
- **support for aai/aao records** for array-style extraction of data
- **type override** - freely change the type of the entry being read or written from and to records, using any type supported by EPICS (regDev and/or PSI EtherCAT Driver style, e.g. t=float or t=uint32,...)

Finally, all of the above extended **addressing modes can be mixed as needed**, without any restriction.

For example, it is possible to have a standard PDO Entry as a record, but to shift the actual data address being read, and then to extract a single bit from the value being read, and finally to force a type change from bit (integer) to a float.



EtherCAT at PSI

Furthermore, programmable EtherCAT modules (such as EL6692, EL6695 and many more) are also supported, as well as slave-to-slave communication

Flexibility - Easy setup of programmable EtherCAT modules

Both PSI EtherCAT and Linux drivers support **easy setup of programmable modules**. For example, to create new EL6692 entries, simply add the following:

ecat2cfgEL6692 <bridge_nr> in <num_of_bits>

...and/or...

ecat2cfgEL6692 <bridge_nr> out <num_of_bits>

Slave-to-slave - Built-in support for real-time slave-to-slave communication

Any number of **slave-to-slave** transactions can be added to allow for **real-time communication** without any additional software. Simply list all the slave-to-slave transactions, and that's it!

ecat2sts r8 r0

ecat2sts r2.b0 r0.b6

ecat2sts s2.sm0.p1.e0 s1.sm0.p1.e0

ecat2sts s3.sm3.p0.e10.b3 s4.sm2.p1.e0.b7

EtherCAT at PSI

All functionality for static configuration and reconfiguration provided by Etherlab Master was added as mini script language, to allow for any configuration needed.

ecat2cfgslave sm - Config Sync Manager

Configures one Sync Manager for the given slave.

ecat2cfgslave sm <slavenr> <smnr> <dir> <wd_mode>

ecat2cfgslave sm_clear_pdos - Clear PDOs of a SM

Clears (i.e. deletes) all PDOs for a given Sync Manager (SM)

ecat2cfgslave sm_clear_pdos <slavenr> <smnr>

ecat2cfgslave sm_add_pdo - Add PDO to a SM

Adds a PDO with index pdoindex to a Sync Manager.

ecat2cfgslave sm_add_pdo <slavenr> <smnr> <pdoindex>

ecat2cfgslave pdo_clear_entries - Clears PDO entries of a PDO

Clears (i.e. deletes) all PDO entries associated with the given PDO.

ecat2cfgslave pdo_clear_entries <slavenr> <smnr> <pdoindex>

ecat2cfgslave pdo_add_entry - Adds a new PDO entry to a PDO

Creates a new PDO entry and associates it with the given PDO..

ecat2cfgslave pdo_add_entry <slavenr> <smnr> <pdoindex> <entryindex> <entrysubindex> <entrybitlen>

PAUL SCHERRER INSTITUT PSI

EtherCAT at PSI

EPICS command-line tools have been added to allow easy overview of registers, current values, slave-to-slave communication and their connection to respective EPICS records.

```
dmap
...
[I] 0x2007:0x00  464.0     8 bits  r354 = 0x14       @ s0.sm3.p0.e81
[I] 0x2008:0x00  465.0    32 bits  r355 = 0x20160303 @ s0.sm3.p0.e82
[I] 0x2009:0x01  469.0    32 bits  r356 = 0x4203ef9b @ s0.sm3.p0.e83
[I] 0x2009:0x02  473.0    32 bits  r357 = 0x4474577b @ s0.sm3.p0.e84
[I] 0x2009:0x03  477.0    32 bits  r358 = 0x41c4b12c @ s0.sm3.p0.e85
[I] 0x2009:0x04  481.0    32 bits  r359 = 0x4196e80f @ s0.sm3.p0.e86
[I] 0x2009:0x05  485.0    32 bits  r360 = 0x44746b5f @ s0.sm3.p0.e87
[I] 0x2009:0x06  489.0    32 bits  r361 = 0x42026b48 @ s0.sm3.p0.e88
[I] 0x2009:0x07  493.0    32 bits  r362 = 0x41aba084 @ s0.sm3.p0.e89
[I] 0x2009:0x08  497.0    32 bits  r363 = 0x4474523d @ s0.sm3.p0.e90
[I] 0x2009:0x09  501.0    32 bits  r364 = 0x42032df5 @ s0.sm3.p0.e91
[I] 0x2009:0x0a  505.0    32 bits  r365 = 0x00000000 @ s0.sm3.p0.e92
[I] 0x2009:0x0b  509.0    32 bits  r366 = 0x00000000 @ s0.sm3.p0.e93
[I] 0x2009:0x0c  513.0    32 bits  r367 = 0x00000000 @ s0.sm3.p0.e94
[I] 0x200a:0x00  517.0     8 bits  r368 = 0x07       @ s0.sm3.p0.e95
...
[O] 0x2001:0x20  549.0     8 bits  r400 = 0x00       @ s12.sm2.p0.e31
[O] 0x2001:0x21  550.0     8 bits  r401 = 0x00       @ s12.sm2.p0.e32
[O] 0x2001:0x22  551.0     8 bits  r402 = 0x00       @ s12.sm2.p0.e33
[O] 0x2001:0x23  552.0     8 bits  r403 = 0x00       @ s12.sm2.p0.e34
[O] 0x2001:0x24  553.0     8 bits  r404 = 0x00       @ s12.sm2.p0.e35
[O] 0x2002:0x01  554.0     8 bits  r405 = 0xd4       @ s12.sm2.p0.e36
...
```

PAUL SCHERRER INSTITUT

EtherCAT at PSI

EPICS command-line tools have been added to allow easy overview of registers, current values, slave-to-slave communication and their connection to respective EPICS records.

```
Slave-to-slave communication

[O] 0x2002:0x01  554.0     8 bits  r405 = 0xd4       @ s12.sm2.p0.e36
                                                        .b0 = 0 <--- s4.sm0.p0.e0 = 0
                                                        .b1 = 0 <--- s4.sm0.p1.e0 = 0
                                                        .b2 = 1 <--- s4.sm0.p3.e0 = 1
                                                        .b4 = 1 <--- s5.sm0.p1.e0 = 1
                                                        .b5 = 0 <--- s10.sm0.p13.e0 = 0
                                                        .b6 = 1 <--- s5.sm0.p2.e0 = 1
                                                        .b7 = 1 <--- s5.sm0.p3.e0 = 1
```

```
EPICS records

...
[O] 0x2003:0x36  667.0    16 bits  r465 = 0x2784     @ s12.sm2.p0.e96 <--- ao TRFCB02-RWVG100-DCP10:REF-ILK-LIML = 0x2784
[O] 0x2003:0x37  669.0    16 bits  r466 = 0x2af5     @ s12.sm2.p0.e97 <--- ao TRFCB02-RWVG200-DCP10:REF-ILK-LIML = 0x2af5
[O] 0x2003:0x38  671.0    16 bits  r467 = 0x2af5     @ s12.sm2.p0.e98 <--- ao TRFCB02-RWVG300-DCP10:REF-ILK-LIML = 0x2af5
[O] 0x2003:0x39  673.0    16 bits  r468 = 0x2a34     @ s12.sm2.p0.e99 <--- ao TRFCB02-RWVG400-DCP10:REF-ILK-LIML = 0x2a34
[O] 0x2003:0x3a  675.0    16 bits  r469 = 0x26d8     @ s12.sm2.p0.e100 <--- ao TRFCB02-RILK-RFDET:SPARE-ILK-LIML = 0x26d8
…
```

PAUL SCHERRER INSTITUT

EtherCAT at PSI

EPICS command-line tools have been added to allow easy overview of registers, current values, slave-to-slave communication and their connection to respective EPICS records.

Aai live content:

```
[I] 0x2004:0x02  678.0     8 bits  r471 = 0x00       @ s12.sm3.p0.e1 ---> aai TRFCB02-RILK:SPI_DATA_OUT = { 0x1e, 0x00, 0x02, 0x4f, 0x00, 0x00,
0x02, 0xf4, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3a, 0x3b, 0x00, 0x00, 0x00, 0x00, 0x00, 0x71, 0x00, 0x00, 0x00,
0x00, 0x00 }
```

Stringin live content (with offset and length):

```
[I] 0x2005:0x02  714.0    32 bits  r504 = 0x43465254 @ s12.sm3.p0.e34 ---> stringin TRFCB02-RILK:FW-NAME =  = 'TRFCB02' (offset +0, length 8)
[I] 0x2005:0x03  718.0    32 bits  r505 = 0x00323042 @ s12.sm3.p0.e35
```

Records and slave-to-slave communication:

```
[I] 0x2006:0x01  722.0     8 bits  r506 = 0xff       @ s12.sm3.p0.e36
                                                            ---> b0: bi TRFCB02-RILK:PSYS = 1
                                                            ---> b1: bi TRFCB02-RLLE-EVR:ILK = 1
                                                            ---> b2: bi TRFCB02-RLLE:WDOG-SUMILK = 1
                                                            ---> b3: bi TRFCB02-RILK-RFSW10:RFON = 1
                                                            ---> b4: bi TRFCB02-RILK-RFSW20:RFON = 1
                                                            ---> b5: bi TRFCB02-RILK:RF-READY = 1
                                                        .b6 = 1 ---> s7.sm0.p0.e0 = 1
                                                        .b7 = 1 ---> s7.sm0.p2.e0 = 1
[I] 0x2006:0x02  723.0     8 bits  r507 = 0x07       @ s12.sm3.p0.e37
                                                                .b0 = 1 ---> s7.sm0.p1.e0 = 1
                                                            ---> b1: bi TRFCB02-RILK:MPS-LEVEL1 = 1
                                                            ---> b2: bi TRFCB02-RILK:MPS-LEVEL2 = 1
                                                            ---> b3: bi TRFCB02-RILK:ANV1-P = 0
                                                            ---> b4: bi TRFCB02-RILK:ANV2-P = 0
                                                            ---> b5: bi TRFCB02-RILK:ANV3-P = 0
                                                            ---> b6: bi TRFCB02-RILK:ANV4-P = 0
                                                            ---> b7: bi TRFCB02-RILK:ANI1-P = 0
```

EtherCAT at PSI

PSI EtherCAT drivers – future plans:

- Support for Beckhoff XML configuration DB (forced/static configuration)
- Support for SDO transfers from EPICS (slave EEPROM updates and similar)
- EtherCAT EPICS driver for Beckhoff IPCs and TwinCAT systems (Windows and Windows-CE based, using Beckhoff TwinCAT API)
- Finish support for multiple EtherCAT masters on a single host

Wir schaffen Wissen – heute für morgen

My thanks goes to:

- Dr. Markus Janousch, Dr. Elke Zimoch, Babak Kalantari and others from Controls Section and Embedded Software Group at PSI for their support
- Roger Kalt und Florian Gärtner from LLRF for testing and ideas
- IgH and Dr. Florian Pose for creating and sharing Etherlab Master