ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA067

EtherCAT BASED DAQ SYSTEM AT ESS

J. Etxeberria*, J. H. Lee, ESS, Lund, Sweden

Abstract

The European Spallation Source (ESS) [1] is a research facility being built in Lund, Sweden. The Integrated Control System (ICS) division at ESS is responsible for the whole ESS machine and facility control, i.e., accelerator, target, neutron scattering instruments and conventional facilities. Therefore, ICS has to be able to handle a wide variety of input and output signals with different user requirements. Moreover, EPICS is selected as the generic control system framework at ICS and the aiming goal is to achive high performance, cost-effective, safe, reliable and easily maintainable systems. In order to meet these goals, ICS needs hardware standardisation. Therefore, to fulfil these requirements different hardware platforms were selected, such as MicroTCA for Fast real-time Input Output, PLC for Industrial Input Output without real-time response requirements, and EtherCAT [2] for real-time requirements and cost-effective applications.

INTRODUCTION

ICS has selected EtherCAT, in order to cover the medium range data rate, e.g., several kHz. The reason is, because, EtherCAT provides many features of Industrial Ethernet at a lower price than a classic fieldbus system, and there is an available open-source EtherCAT master [2]. In addition, it meets real-time requirements with the signal data rate, because EtherCAT is a real-time Ethernet-based fieldbus that relies on conventional Ethernet frames but employs a conceptually different mechanism to communicate with multiple devices. Therefore, EtherCAT standard uses Ethernet wiring but has a specialised protocol that enables tight time synchronisation. Moreover it allows a flexible topology, and a high synchronisation between nodes.

At ESS, for instance, EtherCAT will be used when the Input Output (I/O) system needs to be beam-synchronous; needs to acquire signals in the kHz range; or needs to be spread across locations that are far from each other and would need cumbersome cabling, but still belong to one system.

EtherCAT SYSTEM ARCHITECTURE

As many other fieldbus network systems, EtherCAT is based on Master and Slave configuration. EtherCAT Master relies on standard Ethernet hardware communication with the bus, so any generic network interface card (NIC, 100 MB/s Full duplex) is sufficient. Using the open source IgH EtherCAT Master [3] makes cost-effective and flexible configuration of the EtherCAT system architecture at ESS, meaning that a typical EPICS input-output controller (IOC) can be executed within an industrial PC as an EtherCAT master. However, it needs dedicated and separated Ethernet hardware ports only for the EtherCAT communication.

Contrary to the operation of standard Ethernet, the slaves process the EtherCAT frames on the fly. This requires the use of specific hardware-integrated in the slaves; EtherCAT Slave Controllers (ESC). ESC are available from multiple manufacturers. In ESS, Beckhoff automation technology [4] will be used.

OPEN-SOURCE IgH EtherCAT MASTER

An open-source EtherCAT master [3], IgH EtherCAT Master, works as a Real Time kernel module loaded within the open source operating system Linux to communicate with peripherals devices as EtherCAT slaves through dedicated Ethernet ports. Since it is integrated in the Linux kernel, it has got better realtime characteristic and the master code can directly communicate with any Ethernet hardware. At ESS, to meet real time performance, the PREEMPT Real Time kernel patch should be used with EtherCAT. If there is no need for the real time performance, the typical Linux kernel should be used as well. The IgH EtherCAT Master can contain more than one master running at the same time. Each master is linked with one Ethernet device identified by its MAC address.

Each EtherCAT master can be matched with a character device as an userspace interface

$$\texttt{/dev/EtherCATx}, \quad (1)$$

where `x` is the unit number of the EtherCAT master. This device node is created automatically with `udev` while the master kernel module is loading.

The information transferred from slaves to masters is called Process Data Objects (PDOs), which is in different frequencies and is limited by the size of an Ethernet frame. Master is responsible for the configuration of the Fieldbus Memory Management Unit (FMMU). The IgH EtherCAT Master also provides command-line tools, which allow users to get information about the master, slaves, PDOs and Service Data Objects (SDO) of all the slaves connected to the EtherCAT chain. One example of command line tools is to read PDO information of the Beckhoff terminal `EL3202-0010`, located in the position 10 of the EtherCAT test chain as follows:

```
$ ethercat pdo -p 10 -v
  SM3: PhysAddr 0x1180, DefaultSize 8,
  ControlRegister 0x20, Enable 1
  TxPDO 0x1a00 "RTD TxPDO-Map Ch.1"
  PDO entry 0x6000:01, 1 bit, "Underrange"
  PDO entry 0x6000:02, 1 bit, "Overrange"
  PDO entry 0x6000:03, 2 bit, "Limit 1"
  PDO entry 0x6000:05, 2 bit, "Limit 2"
  PDO entry 0x6000:07, 1 bit, "Error"
  PDO entry 0x0000:00, 7 bit, "Gap"
```

* julenetxeberria.malkorra@esss.se

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA067

```
  PDO entry 0x1800:07, 1 bit, ""
  PDO entry 0x1800:09, 1 bit, ""
  PDO entry 0x6000:11, 16 bit, "Value"
```

EPICS INTEGRATION

ESS uses the PSI EtherCAT driver, `ecat2`, for EPICS Integration [5,6]. The `ecat2` is capable of automatic scanning of the existing device and module layout, followed by self-configuration and autonomous operation of the EtherCAT bus real-time loop. If additional configuration is needed, the driver offers both userspace and kernelspace APIs, as well as the command line interface for fast configuration or reading or writing the module entries.

The EtherCAT modules and their data objects (entries) are completely exposed by the driver, with each entry corresponding to `/dev/EtherCATx`. This way, any user application can read or write the EtherCAT entries in a simple manner, even without using any of the supplied APIs. However, with `ecat2`, SDO access is impossible. Thus, before the Master state machine enters in operation mode, SDOs should be configured through the EtherCAT master command line tools.

Moreover, in order to handle individual EtherCAT terminals, ESS uses the EtherCAT Slave EPICS Database, `ecat2db` [7]. Since some EtherCAT terminals use similar PDO structures, `ecat2db` share their EPICS Database for terminals which have identical PDO entry layouts. Figure 1 shows the ESS EtherCAT Software Layer.

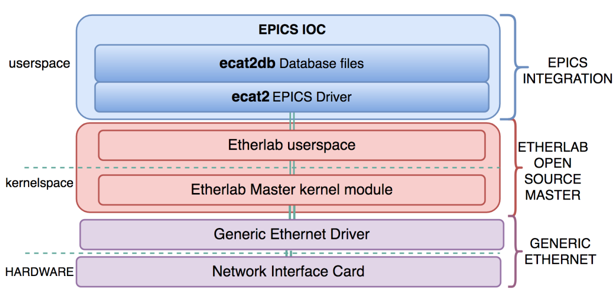

Figure 1: ESS EtherCAT Software Layer

One example for the PDO mapping to EPICS Process Variable through EPICS Database analog input record is shown as follows:

```
record(ai, "ICSLab:ecat02:EL3202-00-RTD1") {
  field(DESC, "Analog input value")
  field(DTYP, "ecat2")
  field( INP, "s10.sm3.p0.e8")
  field(SCAN, "I/O Intr")
  field(LINR, "SLOPE")
  field(ESLO, "0.01")
  field(EOFF, "0")
  field( EGU, "C")
  field(PREC, "2")
}
```

In the example, within `INP` field, `s10` shows the slave number 10, `sm3` does synch manager number 3, `p0` means PDO number 0, and `e8` does PDO entry 8.

ESS DAQ SYSTEM

In order to define a standard procedure to deploy EtherCAT based DAQ system and to test its practical performance according to different subsystem configurations, the DAQ test-bed has been setup using some Beckhoff EtherCAT terminals and couplers. Figure 2 shows the EtherCAT based ESS DAQ Testbed. And Figure 3 shows the testbed system picture.

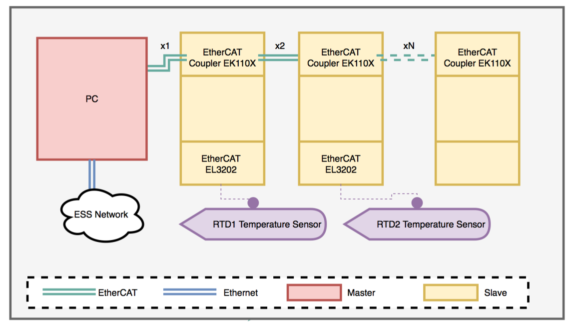

Figure 2: ESS DAQ System Testbed Schematic Drawing. The green dotted line and additional Slave will be installed for further studies.

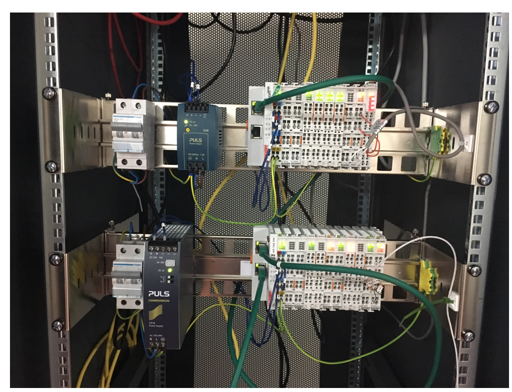

Figure 3: ESS DAQ System Testbed

The EtherCAT based ESS DAQ testbed has the following components:

- An industrial PC with Linux CentOS 7 without PREEMPT Real Time patch
- Open-Source IgH EtherCAT Master
- ESS EPICS Environment : EPICS Base 3.15.4, ecat2 2.1.2, and ecat2db 0.4.3
- Two EtherCAT Slave Chains within 19 inch DIN rail.
- Each Slave chain has the Beckhoff EK110X coupler, and several Beckhoff AI, AO, DI and DO terminals.
- Two Pt100 temperature sensors have been attached to the Beckhoff EL3202-0010 terminals.

The Process Variables (PVs), that are two temperature values (`EL3202-00-RTD1` and `EL3202-01-RTD2`), IOC heart-

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA067

beat, and system cpu load percent (SYS_CPU_LOAD) were collecting with the EPICS Archiver Appliance, which is one of the ESS ICS EPICS applications, for the basic performance studies. In addition, the simple graphical user interface, shown in Fig. 4, has been developed by using Control System Studio, which is also one of the ESS ICS EPICS applications.

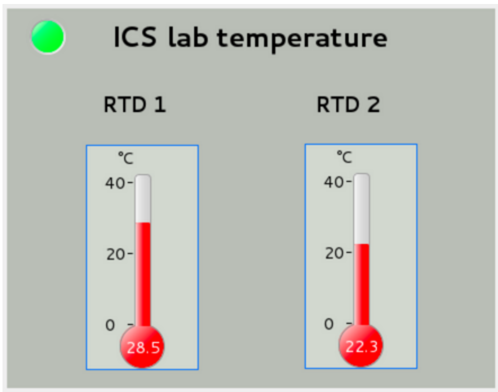

Figure 4: The simple user interface screen for the testbed. At the top left, the light green widget shows the heartbeat signal of the running EPICS IOC.

During the data collection for 10 days with 1 kHz data rate, there is no performance degrade and no glitch in the archived signals. Figure 5 shows the measured temperature signals with the EPICS Archiver Appliance.

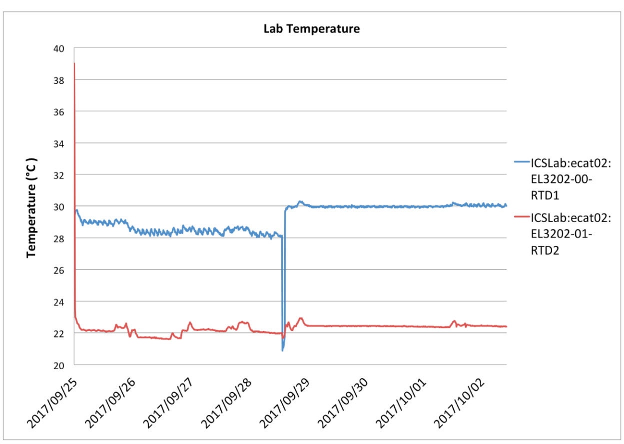

Figure 5: Measured Temperature Signals. The big temperature changes in RTD1 and RTD2 were introduced artificially.

Moreover, it is important to see the CPU behaviour while the signals are collecting. Figure 6 shows the stable CPU load around 58% with small variation.

SUMMARY AND OUTLOOK

During this study, the ESS standard EtherCAT DAQ components are identified, and the standard procedure to com-

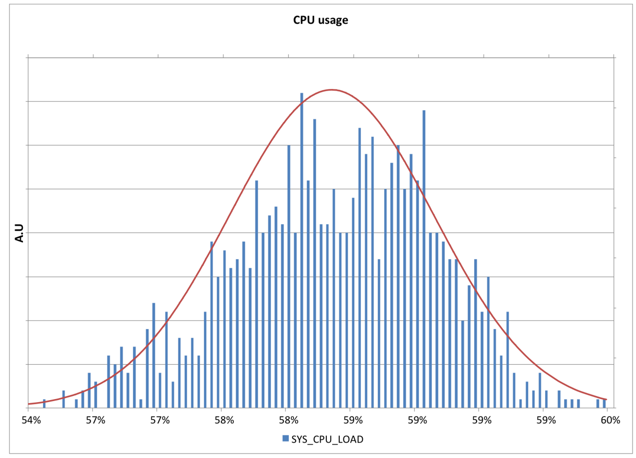

Figure 6: CPU Load Distribution. The distribution can be identified by a normal distribution. Thus, there is no abnormal event while the 1kHz signals are collecting.

bine each components to a simple and expandable EtherCAT based DAQ system, i.e., the ESS EtherCAT DAQ Crate. The first performance measurement meets the ICS expectation. On the basis that, ICS will perform more serious tests, which contains that time synchronization, latency measurement, maximum cable length between crates, different frequency implementation, several masters configuration in a PC, CPU performance test, Linux Kernels PREEMPT Real Time patch, and tests with subsystem practical implementation.

REFERENCES

[1] European Spallation Source ERIC website, `http://europeanspallationsource.se`

[2] EtherCAT Technology Group website, `https://www.ethercat.org/en/technology.html`

[3] IgH EtherCAT Master for Linux at EtherLab website, `http://www.etherlab.org/en/ethercat/`.

[4] Beckhoff, New Automation Technology website, `https://www.beckhoff.com/`

[5] PSI EtherCAT Driver for EPICS in github, `https://github.com/paulscherrerinstitute/ecat2`

[6] D. Maier-Manojlovic, "Real-Time EtherCAT Driver for EPICS and Embedded Linux at Paul Scherrer Institute (PSI)", in *Proc. ICALEPCS2015*, Melbourne, Australia, Oct. 2015, paper MOPGF027, `https://doi.org/10.18429/JACoW-ICALEPCS2015-MOPGF027`, 2015.

[7] ESS EtherCAT Slave EPICS Database in bitbucket, `https://bitbucket.org/europeanspallationsource/m-epics-ecat2db`