EXPERIENCE AND PROSPECTS OF REAL-TIME SIGNAL PROCESSING AND REPRESENTATION FOR THE BEAM DIAGNOSTICS AT COSY
COSY, IKP-4, Forschungszentrum Jülich GmbH, Jülich, Germany

Abstract
Diagnostics of beam parameters is vital for the operation of any particle accelerator and contributes to the precision of the physics experiments. At COoler SYnchrotron of the Forschungszentrum Jülich there are several beam instrumentation subsystems with data acquired and processed in real-time for machine and operator use to ensure safe and efficient performance. Here are presented current development for the Beam Loss Monitor (BLM) with regard to usage of field programmable gate arrays (FPGAs) to achieve fast data processing and integration into the Experimental Physics and Industrial Control System (EPICS) used at COSY. Also presented is a way to create and run Graphical User Interfaces based on EPICS variables with Control System Studio (CSS) connected to a data archiving system to display and use previously collected data.

COOLER SYNCHROTRON
Jülich particle accelerator and storage ring COSY (figure 1) is operated for proton or deuteron beams in the energy range of 45-2880 MeV for protons. Beam polarization, stochastic and electron cooling, storage times between few seconds to several hours and the flexibility of beam optics make COSY an ideal research site with a physics programme including machine studies and detector tests for FAIR project and preparations of the measurement of electric dipole moment experiments (EDM [1]). The same features are posing a diagnostic challenge for the beam parameter monitoring. One such parameter is beam loss rate.

BEAM LOSS MONITOR
The beam loss monitor system (BLM) shows loss of particles from the beam orbit. There are numerous processes during machine adjustment and operation contributing to beam losses, which are not easy to quantify or predict. This imposes several requirements on a BLM system: permanent and fast monitoring, logging, and feedback to the operators.

The BLM system in COSY consists of 9 radiation detectors (7 along the ring and 2 on the extraction beam line) (cf. figure 1). Figure 2 shows the signal flow scheme of each BLM crate.

Figure 2: BLM signal flow scheme: particles produce light in the scintillator, which is then converted to analogue electrical signals in the photomultiplier; these are discriminated and then digitized by the Red Pitaya (DAQ); the counts are collected and published to the EPICS [2] network from the IOC.

HARDWARE
The radiation detectors are made of encapsulated scintillating material (plastic or liquid) coupled with photomultiplier tubes (PMT). The detectors were calibrated using radioactive sources and deployed along the ring in likely or known loss locations. Installation of more detectors is planned. Each BLM crate contains modules for PMT high-voltage and preamplifier voltage supply, as well as a discriminator module developed and produced in Forschungszentrum Jülich. The discriminators are able to process both positive and negative detector pulse polarities, with $2\text{mV}$ granularity (figure 3) and have 5 analogue inputs. The data acquisition (DAQ) is performed by a Red Pitaya board [3], which is embedded in each discriminator module.
A Linux image of only 14MB is shipped with the Red Pitaya and runs a complete operating system, including the SSH server, nginx server, and busybox tools. The EPICS 3.15 base [2] was cross-compiled and added to the system providing full EPICS server and Channel Access functionality. A custom IOC with specific device support was written to control and communicate with the FPGA, process the data and serve the BLM Process Variables (PVs) to the clients on the network. NTP synchronization with a central COSY server is used to ensure correct data context with other diagnostics and control devices.

One of the clients is the GUI described below, another is the Archiver appliance, which is recording the relevant data for all IOCs and is used to correlate data posteriori.

**COUNT TAKING SCHEME**

The FPGA counts the input signals and has to deliver consistent data and the following scheme was developed to read-out the input rates.

![Count taking scheme](image)

**Figure 5:** Count taking scheme: signal representations for a single input signal $signal\_i$, clock $CLK$, and corresponding counter integers $counter$ and $clk\_counter$, keep input signal $keep\_i$, and output integers $counter\_keep\_o$ and $clk\_counter\_keep\_o$. Rising edges of $signal\_i$ and $CLK$ increment corresponding counters; rising edge of $keep\_i$ triggers storage and output of the integers in corresponding additional registers.

All input signals and the clock are counted independently with 32-bit depth, figure 5 shows an example for one counter. Edge sensitivity can be set independently for all inputs - rising edge sensitive in the example. On positive edge of $keep$ input signal, the state of all counters including the clock are stored in additional registers for asynchronous read-out (sample and hold), allowing precise and lossless rate estimation independent of CPU or network latencies. EPICS PV scan rate of 10Hz is used to trigger the read-out via $keep$ and publish the data which is sufficient for daily monitoring.

**Burst Mode**

For rapidly changing inputs an additional data taking mode was implemented. Initiated by an external trigger, which may be manual or timed to a specific event in the machine, the $keep$ register is toggled at about 38.4kHz taking 105 values for 4 counters and the clock. The values are read and cached, before they get published via a waveform PV every few seconds. The toggling is driven by the IOC, so that typical (non-low-latency) operating system constraints...
parallel running processes apply - the sampling periods deviate 1% of the time with the distribution shown in figure 6.

Figure 6: Measurement of the read-out rate in the burst mode; the clock PV waveform was recorded from the IOC and several measurements were stacked and histogrammed. The mean time slices $dT$ is $26\mu s$, with significantly longer (undesired) time slices of up to $500\mu s$ occurring with less than 1% probability (integral).

BEAM LOSS MONITOR - GUI

Figure 7: The Beam Loss Monitor GUI was made with Control System Studio (CSS). The background is a picture of the COSY ring to see where which BLM is installed. For each BLM there is a graph that shows the beam losses over time. Detailed views open with a click on one of the graphs.

PROGRAMMING TOOL - CONTROL SYSTEM STUDIO

Another example of the use of the CSS along the BLM system is the recently upgraded Beam Position Monitoring (BPM) system at COSY [4]. Here the CSS GUI and EPICS back-end are to be used to control system parameters such as gain, the calibration, and measurements parameters as well as for the data acquisition and processing.

OUTLOOK

BLM Hardware

Several improvements are to be implemented for the beam loss monitor system. FPGA driven read-out of the BLM with sufficient FIFOs and a DMA-like scheme has to be implemented to achieve sub-$\mu s$ rate sampling granularity. Integration of the charge under a pulse additionally to the pulse count is envisaged to address pulse pile-up and achieve better resolution in energy deposit recognition.

BLM GUI

As it is the GUI only shows live values. We plan to refer the beam losses to the cycle by plotting the losses against cycle time. We also want to add up the losses of one cycle to get the total losses.

REFERENCES

[3] "Red Pitaya STEMlab Documentation". Release 0.97; http://redpitaya.readthedocs.io