DATA ACQUISITION AND CONTROLS INTEGRATION OF THE AWAKE EXPERIMENT AT CERN

Veronica K. Berglyd Olsen*[a], Spencer J. Gessner[b], Jozef J. Batkiewicz[b], Stephane Deghaye[b], Edda Gschwendtner[b], Patric Muggli[b,c]

[a] *University of Oslo, Oslo, Norway*
[b] *CERN, Geneva, Switzerland*
[c] *Max Planck Institute for Physics, Munich, Germany*

Abstract

The AWAKE experiment has been successfully installed in the CNGS facility at CERN, and is currently in its first stage of operation. The experiment seeks to demonstrate self-modulation of an SPS proton beam in a rubidium plasma, driving a wakefield of several gigavolt per meter. We describe the data acquisition and controls system of the AWAKE experiment, its integration into the CERN controls system and new control developments specifically required for the AWAKE experiment.

INTRODUCTION

AWAKE is an Advanced Wakefield Experiment designed to demonstrate proton driven plasma wakefield acceleration utilising a 400 GeV proton drive beam from the Super Proton Synchrotron at CERN [1]. The first phase of the experiment has been successfully installed in the former CNGS facility, and was commissioned in October and November 2016.

The first phase of AWAKE is intended to demonstrate the self-modulation instability in the proton drive beam [2], and we had a first short 48 hour run with rubidium plasma and protons in December 2016. Further three weeks of beam time are scheduled at the end of May and in August 2017. The installation of the electron beam phase is scheduled to be completed in September 2017, with first physics expected in November.

CERN CONTROLS SYSTEM

Located at CERN, the AWAKE experiment is taking advantage of the extensive support infrastructure that already exists for experiments. This also includes integration into the CERN controls system.

The Large Hadron Collider is controlled through the Front End Software Architecture (FESA), developed at CERN. This software framework has later been extended and generalised in FESA3 to be useable by other experiments as well. FESA device classes are developed based on standardised and modular code tailored for each specific device. These classes are split into a real time and a server process. The real time process is intended to access the hardware directly, and in addition provide access to internal and external timing as well as information from other device classes. The server process provides an interface for get and set operations for device settings and acquired data, as well as subscription to any further data updates [3].

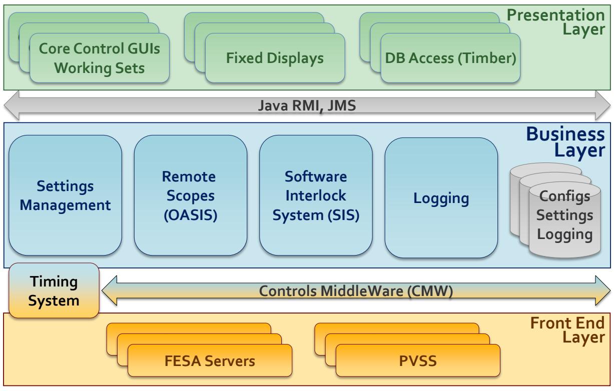

Figure 1: The CERN Control System is structured in three main layers. The Front End Layer consist of VME crates, PCs and PLCs dealing with high performance data acquisition and real time processing. These communicate with application, database and file servers as well as central timing on the Business Layer via the Controls Middleware. Graphical user interfaces and database access are found on the Presentation Layer, and interact with the Business Layer via Java APIs. [4].

FESA classes run on Front End Computers (FEC), which run on the Linux operating system. The data from these classes are fed to both data logging systems and control room displays and interfaces as outlined in Fig. 1.

DATA ACQUISITION FOR WINDOWS BASED INSTRUMENTS

The AWAKE experiment has largely been integrated into this infrastructure through direct hardware access between the instruments and the FESA framework. However, some of the instruments depend on proprietary software that is not supported by the standard Front End Computer platform running Scientific Linux. In order to get around this, and to avoid writing new software, data from three of the instruments currently have to be written to files on shared folders on Windows computers. These files are then imported by designated FESA file reader classes running on CERN supported FECs.

Three instruments currently require software or drivers only available for the Windows operating system:

[*] v.k.b.olsen@fys.uio.no

AWAKE uses Mach-Zehnder type interferometers to measure and calculate the plasma density at either end of the 10 m rubidium plasma cell. The acquired interferogram is stored as a file from the instruments software, and a fitting algoritm is applied to calculate the plasma density to within ±0.5% [5]. As the fitting is too computationally heavy to run in real time, the fitting is not currently done by the file reader, but will be performed by a separate FESA class running on a dedicated computer.

The rubidium plasma is ionised by a 780 nm, 4.5 TW peak power laser with a pulse length of 100 – 120 fs [6]. The pulse length is measured by a single shot optical autocorrelator [7]. The autocorrelator itself is a commercial product, and we extract the data directly from its camera via its Windows drivers. On the local computer we compute the projection, fit the projection with a sech^2 function using Levenberg-Marquardt, extract the pulse width, and write the width and the full image to a binary file.

In addition, we have a 4-channel Tektronix oscilloscope with 23 GHz analogue bandwidth and a 50 GSa sampling rate per channel. The oscilloscope is used to measure real time signals from various Schottky diodes installed in a proton beam diagnostic setup downstream the plasma cell. They measure Coherent Transition Radiation (CTR) emitted in the microwave band. We uses proprietary Windows software to automatically saves all channel data files at each event trigger.

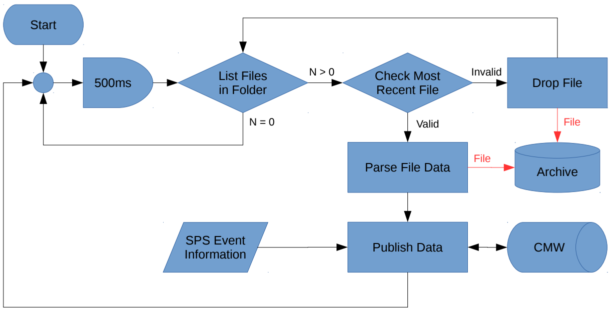

Figure 2: The FESA file readers written specifically for AWAKE all operate in the same way. A process polls a dedicated watch folder every 500 ms for new files. The file content is verified according to the instruments' expected format, and either imported and archived, or moved to a dropped folder for later manual control. Imported data is immediately made available on the CMW interface to subscribed services. [8].

Each of these instruments require their own FESA class to handle their respective file formats. However, each of these classes operate in the same way, and therefore only require individual data parsing code. A flow chart illustrating their operation is shown in Fig. 2. The beam cycle time for the experiment itself is around 30 seconds [6], but some of the instruments may be required to run at a higher frequency. The 500 ms delay is both chosen to allow for a 1 Hz data frequency with some margin, as well as to ensure there is no significant delay between the time the file is written and when the data is available to the data logging layer. The class will poll a dedicated watch folder and attempt to import all files present in reverse chronological order. This is to ensure all data is imported in case of for instance a network interuption. Valid files are then parsed, the data forwarded, and the file itself moved to an archive folder. In the event of an invalid file, a warning is raised and the file is moved to a dropped files folder for later manual verification.

PXI DIGITAL CAMERA SYSTEM

AWAKE uses the analog camera system provided by the CERN BI group to monitor the proton beam at BTV stations along the beam line. The analog camera system is radiation hard and requires minimal mechanical upkeep. However, the system is asynchronous to beam extractions. It acquires frames at a fixed rate of 50 FPS and records the frame corresponding to beam extraction. This method of acquisition is not ideal for AWAKE, because AWAKE uses scintillating Chromox screens to image the beam. The scintillation has a long decay time of 140 ms, which is longer than the frame exposure time of 20 ms. This means that under identical experimental conditions, the recorded beam intensity on the screen will vary, simply because the analog frame is not synced with the beam arrival time.

In addition to this issue, the analog cameras cannot acquire data at 10 Hz, which is the repetition rate of the laser. For purposes of feedback and stability, it is critical to monitor the laser at this frequency. A digital camera system was implemented to monitor the laser. The camera server is a PXI crate made by National Instruments, and it comprises a trigger and timing system as well as GigE framegrabbers. The digital cameras are made by Basler Ace models and the system supports both CCD and CMOS sensors in a variety of sizes. The image data and camera power are both delived by a GigE connection using the Power-over-Ethernet (PoE) standard.

The digital camera system was implemented at the laser merging point of the beamline and survived the radiation recieved during operations in 2016. Because of this success, the digital cameras are being implemented along the beamline to measure both the particle and laser beams. The cameras will be monitored for radiation exposure at areas where high doses are expected in order to understand the total integrated dose (TID) and the single event upset rate (SEU).

EVENT BUILDER

Devices at AWAKE can be read synchronously with SPS beam extractions or asynchronously between extractions. The synchronous/asynchronous distinction depends on the device. For instance, BPMs are read out only when the proton beam is present, but the temperature probes for the Rubidium cell are read out continuously in one second intervals. All of the data from the AWAKE and SPS diagnostics are recorded by the logging system and it is possible to reconstruct the experiment after the fact. It is also desirable to have fast event reconstruction. In order to facilitate this process, the Event Builder was developed to take a "snapshot" of the experiment at the time of the SPS extraction. The Event Builder is subscribed to the key experimental diagnostics, and records their values at the time of extraction, thus providing an instantly correlated dataset comprising both the synchronous and asynchronous variables.

The Event Builder is a JAVA client that is able to subscribe to any variable exposed by the CMW. The Event Builder includes a time-out feature that waits for devices to return data following an extraction. Once the time-out ends, the Event Builder collects the data from all devices and writes them to an HDF5 file, which can be analyzed instantly. This data is also copied to the CERN EOS system once per day.

SUMMARY

The integration of the AWAKE experiment into the CERN controls system posed a number of challenges. CERN Front End Computers run on Scientific Linux, a platform not supported by all our instruments. The straight forward solution was to let the Windows based instruments write data dump files on their respective computers, and then use the standard CERN Front End Software Architecture framework to write file reader classes that can import these via the network.

Due to the relatively large time interval between events, roughly 30 s, the data can be collected based on time stamps and collected per event in HDF5 files by an Event Builder. These event files are available immediately after an event, as well as backed up and stored for later analysis.

ACKNOWLEDGEMENTS

The authors would like to thank the AWAKE team at CERN, as well as the CERN Beams Department for all their assistance in getting the experiment up and running.

REFERENCES

[1] E. Gschwendtner, et al., in Proceedings of IPAC 2014 (2014), pp. 582–585.
[2] A. Caldwell, et al., Nucl. Instr. Meth. Phys. Res. A **829**, 3–16 (2016).
[3] A. Schwinn, et al., in Proceedings of PCaPAC 2010 (2010).
[4] R. Gorbonosov, AWAKE Collaboration Meeting, Dec. 2013.
[5] E. Öz, F. Batsch, and P. Muggli, Nucl. Instr. Meth. Phys. Res. A **829**, 321–325 (2016).
[6] E. Gschwendtner, et al., Nucl. Instr. Meth. Phys. Res. A **829**, 76–82 (2016).
[7] F. Salin, et al., Appl. Opt., AO **26**, 4528–4531 (1987).
[8] V. K. Berglyd Olsen, 17th AWAKE Technical Board, May 2016.