CONTROL SYSTEM DEVELOPMENTS AT THE ELECTRON STORAGE RING DELTA

D. Schirmer*, A. Althaus, F. Bahnsen
Center for Synchrotron Radiation (DELTA), TU Dortmund, Germany

Abstract

Increasing demands, mandatory replacement of obsolete controls equipment as well as the introduction of new software and hardware technologies with short innovation cycles are some of the reasons why control systems need to be revised continuously. Thus, also at the EPICS-based DELTA control system [1], several projects have been tackled in recent years:

(1) Embedding the new CHG[1]-based short-pulse facility for VUV and THz radiation [2-4] required, for example, the integration of IP-cameras, Raspberry-Pi PCs and EtherCat/TwinCat wired I/O-devices [5, 6].

(2) The request for a staff-free control room led to the programming of new web applications using Python [7] and the Django framework [8]. This development resulted in a web-based interlock system that can be run, amongst others, on Android-based mobile devices.

(3) The virtualization infrastructure for server consolidation has been extended and migrated from XEN [9] to the kernel based KVM [10] approach.

(4) I/O-units which were connected via conventional fieldbus systems (CAN, GPIB, RS-232/485), are now gradually replaced by TCP/IP-controlled devices.

This paper describes details of these upgrades and further new developments.

INTRODUCTION

DELTA, a 1.5-GeV electron storage ring, is operated since 1999 by the TU Dortmund University as a synchrotron light source for campus-based, regional and international users. Since 2011, the facility has been extended by a short-pulse source for VUV and THz radiation making use of the CHG principle. An upgrade to EEHG[2] is in preparation [11]. Not only for these reasons the EPICS-based DELTA control system has been revised and complemented in many fields.

NEW SOFTWARE

Network Administration Tool

Since the number of control system network devices and, thus, the complexity of the network topology increased constantly, it was mandatory to develop a DELTA-specific management tool. With the help of this tool, all individual devices and their connection properties are registered centrally. It administers network interfaces, IP/MAC numbers and assignments to domains. Furthermore, it generates consistent DNS/DHCP configuration files as well as location and network plans. The program is implemented as a web application and is based on the high-level Django/Python framework [7, 8]. All data and configurations are stored in the DELTA MySQL [12] database.

Web Applications

The local radiation safety authority has given its consent to a non-permanent attendance of the control room during standard machine operation. For security reasons, however, it was necessary to develop an additional alert system, which notifies the operator to machine malfunctions via a mobile phone.

For real-time monitoring of the accelerator interlock warnings, a web-socket connection is performed, which keeps a full-duplex communication link between a client (e.g., an Android-based mobile phone) and a server continuously open. The connection uses WAMP[1] [13], a web-socket sub-protocol (with RPC[2] and publish & subscribe mechanism).

On the server side (see Fig. 1 below), *Crossbar.io* [14] is the central software component. It interacts as a message WAMP-router for different so-called service nodes. One node provides the service to read EPICS records another calculates und triggers interlock warnings (all implemented with Python). In addition, it launches web services that provide panels accessible from various URLs. There are web pages for machine status (e.g., beam current, lifetime, insertion device status), interlock notifications as well as the electronic logbook (see Fig. 2).

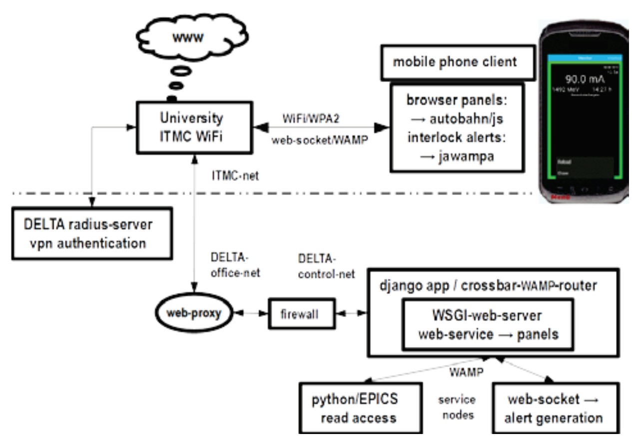

Figure 1: Software setup of the WiFi alert system.

On the client side (see Fig. 1 top), e.g., an Android-based smartphone, an *Autobahn/JS*-application [15] (an implementation of WAMP in javascript) displays the

*detlev.schirmer@tu-dortmund.de
[1]Coherent Harmonic Generation
[2]Echo -Enable Harmonic Generation

[1]Web Application Messaging Protocol
[2]Remote Procedure Call

ISBN 978-3-95450-148-9

server-side spawned panels, and a *jawampa*-application [16] (library to support WAMP to Java) indicates the interlock warnings, all in real-time. The communication is established via WiFi inside the DELTA building. For security reasons, all WiFi-clients must connect to a specially preconfigured ITMC[1]-WiFi controller including an authentication mechanism (radius server [17]). After a successful login, a web-proxy inside the DELTA office network relays the link through a firewall to the dedicated web server/WAMP router within the machine control network, from where EPICS access is possible. An acoustic and vibration alarm will be activated on the mobile phone in the case of a machine alert or server connection lost (see Fig. 1).

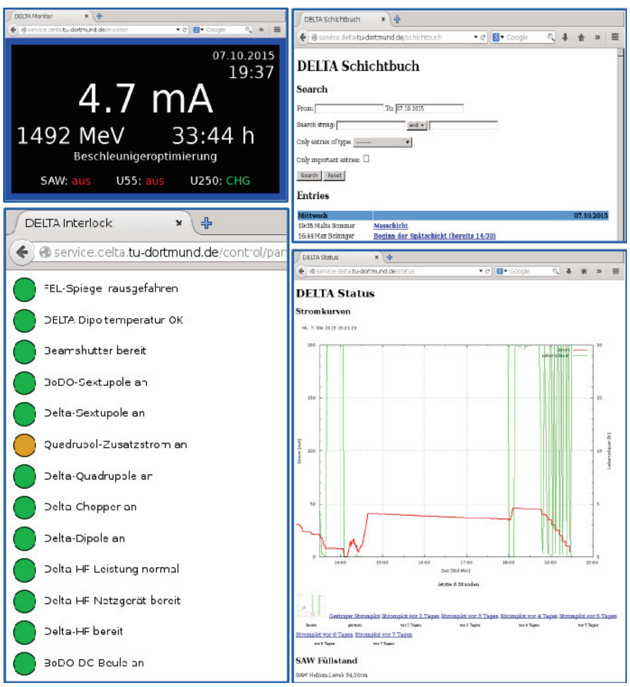

Figure 2: Web applications (panels for: DELTA monitor, logbook, interlock and status).

As another consequence, the central display server for all status monitor panels in the machine hall and the office corridors uses the same standardized web service (see URLs as already discussed above). Also PC-client access to the web-based logbook (elog) or the interlock panel is provided within the DELTA-Intranet or via VPN-access even worldwide. Dedicated permitted URLs are also accessible directly from the Internet (see Fig. 2).

Furthermore, a new acoustic message announcement system has been implemented. Alarms triggered by electron beam losses, temperature warnings, and orbit control problems as well as the injection timer are now audible in the control room and the machine hall. A PyEpics [18] script evaluates corresponding Epics records and plays appropriate speech announcements (pre-arranged voice-files).

[1]IT & Medien Centrum of the TU Dortmund

Server Consolidation

Since 2009, the XEN paravirtualization technique [9] was successfully in use. It was an ideal basis to consolidate and modernize the DELTA client/server control system network [19]. Nowadays, all server-CPUs support hardware-assisted virtualization (VT, AMD-V) and, in addition, kernel-based virtualization (KVM [10]) is by default merged into the standard Linux kernel. The migration to this kind of full virtualization was required for maintenance reasons. KVM is driven by *qemu* [20] and provides an API to control VMs with GUIs like *virt-manager* [21]. In the course of this migration, all server and virtual machines have been upgraded to the latest operating system (Linux Debian 7/8 [22]). This upgrade implies also the introduction of *systemd* [23], a new system and service manager for Linux-OSs and the successor of *sysvinit* [24]. Thus, even all EPICS soft IOC shells are now centrally managed by this powerful OS-integrated software package.

Also for the purpose of maintenance the default installation of all servers is now centrally and automatically managed by the software suite *Ansible* [25]. This software package provides a simple way to automate administration tasks and to setup a complete server infrastructure. Essentially, it is a model-driven configuration management and ad-hoc task execution tool. The in-house developed network tool stores primarily all IT devices including their connection properties in database entries. Based on this information, *Ansibel* generates, updates and transfers the corresponding configuration files and restarts all affected servers subsequently. Thus, the complete DELTA network configuration such as DHCP, DNS, VPN, NTP and user/group installation as well as login accounts (ssh keys, passwords) are automatically supervised site-wide.

INTEGRATION OF NEW HARDWARE

Today, real-time requirements are fulfilled satisfactorily by specially designed fast electronic circuits (e.g. DSP-driven booster synchrotrons [26] or FPGA-operated fast orbit feedback systems [27, 28]). But in most instances accelerator control applications do not need strong real-time features. As many hardware units provide standard Ethernet interfaces, they can be controlled over the TCP/IP network. Thus, for example, the error-prone RS232-serial to CAN-bus readout of the vacuum gauge control units has been replaced by MOXA port communication devices [29] that allow to control different kinds of serial devices directly over a TCP/IP-based Ethernet, avoiding the extensive VME-CAN-controller (VxWorks [30]) middle-layer.

For the same reason, several PLCs (e.g., electron gun control, valve and vacuum-pressure interlock) were subsequently equipped with TCP/IP communication modules. Even recent power supplies for the storage ring quadrupole magnets, also prepared for Ethernet connectors, are now driven directly over the TCP/IP control network, as far as synchronized magnet ramping

ISBN 978-3-95450-148-9

on a sub-second timescale is not required. The I/O-bus interface for the EPICS device support is realized by use of the well-established StreamDevice [31]/asynDriver [32].

More special hardware and software is necessary in order to drive mirrors, for example, in the evacuated laser beam lines of the short-pulse facility. Because the current system has been discontinued, a new control unit is being evaluated. Here, stepper motors [33-35] are controlled over EtherCAT [6], a vendor-initialized real-time Ethernet (see Fig. 3).

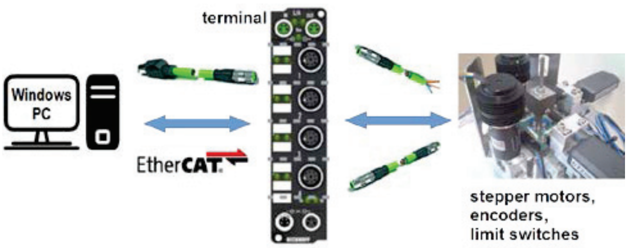

Figure 3: Hardware setup of stepper motor control [35].

The hardware configuration, such as a setup of motor parameters, is managed by the TwinCAT system manager [5], whereas the motor movement logic is programed on a PLC controller. The link to EPICS is accomplished via a vendor specific OPC[1]-server [36] and an EPICS IOC-shell with OPC driver support (OPC client) [37]. OPC is an interoperable standardized platform for industrial automation [38]. All three software levels are running on the same Windows-PC internally communicating over DCOM [39] (see Fig. 4).

Figure 4: Software setup of stepper motor control [35].

As the software encoding of the stepper motors is imprecise due to slippage and x/y motion coupling, the determination of the true mirror position must be realized by high-accuracy contact sensors and additional limit switches [33]. The analog sensor signals are connected to converter device providing a standard serial (RS232) communication interface. This port can be used again by RS232 EtherCAT terminal or by a single board controller (SBC, Raspberry-Pi) running Linux and PyEpics [33, 40]. A similar readout is possible for the digital signals of the limit switches (see Fig. 5).

[1]Open Platform Communication

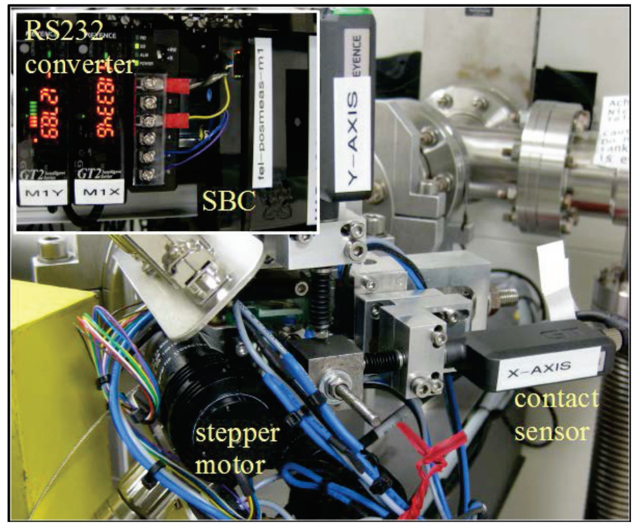

Figure 5: SBC for mirror position read back.

SINGLE BOARD COMPUTER (SBC) PROJECTS

SBCs are more and more the system of choice for "quick-and-dirty" all-round solutions. They run as central display-servers in the control room to present live images of oscilloscopes which are distributed inside the storage ring hall, or work as DHCP/DNS/VPN backup servers for the office network. Further examples are:

Test Bed for DAC Controller Cards

The DELTA booster synchrotron (BoDo) is ramped rather slowly by VME-IOC computers (from 70 MeV to 1.5 GeV and back in approx. 7 seconds). The ramp curve is user-defined and has to fulfil several sensible conditions. All booster magnet power supplies are driven by 16-bit DAC boards. The functionality of these boards is quite crucial and must be checked in advance by a simple SBC-based test bed. A python script on the SBC converts the booster ramping curve to a 16-bit digital I/O signal which is mapped via a photocoupler adapter card for voltage level matching and finally fed to the DAC-board input. The resulting analog DAC output value is measured and analyzed under several test conditions (e.g. ramp cycle frequency, curve shape, temperature variation) by a circuit analyser.

Laser Beam Alignment Diagnostics

For near-field angle measuring of the beam guidance in the laser laboratory, a SBC, equipped with a standard 5-megapixel CCD camera, is in preparation (see Fig. 6). A converging lens focuses the beam on the CCD chip. Changes of beam angle yield a shift of the focal point of the Gaussian intensity distribution. A python script quantifies this displacement and writes the measured value to EPICS records. Background radiation can be reduced via a SBC-GPIO triggered slit. This basic cost-saving setup allows angular measurement within very short distances with a resolution of 50 µrad in both transverse directions [40].

ISBN 978-3-95450-148-9

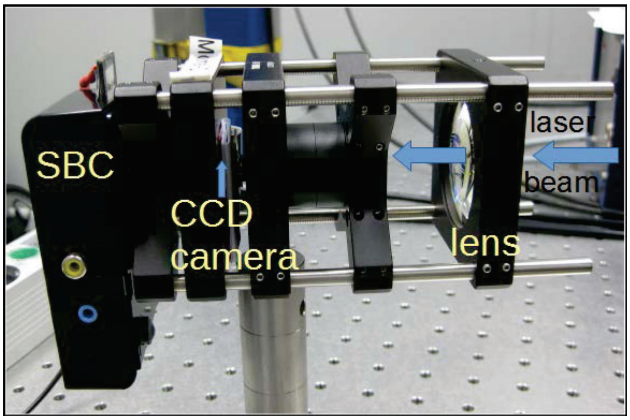

Figure 6: Test setup for laser beam diagnostics [40].

CONCLUSION AND OUTLOOK

The already existing hardware and software at the DELTA storage ring undergoes continuous modifications, mainly pushed by short industrial development cycles. The integration of new experimental setups described in this paper and others not discussed here are primarily for the self-evident purpose that accelerator and laser components can be remotely controlled or read out. To keep track of the increasing number of devices and their complex networking, the installation of new administration tools was necessary.

The future work will still put emphasis on the development of advanced web applications for machine control and monitoring. Furthermore, data-mining and post-processing has increasingly become an important task. Therefore, to optimize data analysis processes, an entire redesign of the DELTA MySQL database is essential.

ACKNOWLEDGMENT

We would like to thank all colleagues of the DELTA team for permanent support. In particular we thank Dennis Zimmermann, Carsten Mai and Boris Sawadski for constructive discussions and for providing detailed information about mirror control system, laser beam diagnostics and single-board controller applications.

REFERENCES

[1] D. Schirmer et al., Proc. ICALEPCS'07, Knoxville, 356.
[2] H. Huck et al., Proc. FEL 2011, Shanghai, 5.
[3] S. Khan et al., Synchrotron Radiation News, 26:3, 25 (2013).
[4] P. Ungelenk et al., Proc. IPAC 2014, Dresden, 1936.
[5] http://www.beckhoff.de/twincat
[6] http://www.beckhoff.de/EtherCAT
[7] https://www.python.org
[8] http://www.djangoproject.com
[9] http://www.xenproject.org
[10] http://www.linux-kvm.org
[11] S. Hilbrich et al., Proc. FEL 2014, Basel, 255.
[12] https://www.mysql.com
[13] http://wamp-proto.org
[14] http://crossbar.io
[15] http://autobahn.ws/js
[16] https://github.com/Matthias247/jawampa
[17] http://freeradius.org
[18] https://pypi.python.org/pypi/pyepics
[19] D. Schirmer et al., Proc. ICALEPCS'09, Kobe, 741.
[20] http://wiki.qemu.org
[21] https://virt-manager.org
[22] https://www.debian.org
[23] http://freedesktop.org/wiki/Software/systemd
[24] https://wiki.debian.org/Debate/initsystem/sysvinit
[25] http://www.ansible.com
[26] B. Keil, Ph.D. thesis, TU Dortmund (2003).
[27] M. Höhner, Ph.D. thesis, TU Dortmund (2015).
[28] P. Towalski, Ph.D., TU Dortmund (in preparation).
[29] http://www.moxa.com
[30] http://www.windriver.com
[31] http://epics.web.psi.ch/software/streamdevice
[32] http://www.aps.anl.gov/epics/modules/soft/asyn
[33] B. Sawadski, Masters thesis, TU Dortmund (2015).
[34] Dennis Zimmermann, private communication.
[35] J. Grewe, M. Taghavi, DELTA Int. Report, 1+2/2014
[36] http://beckhoff.com
[37] http://www-csr.bessy.de/control/SoftDist/OPCsupport
[38] http://www.opcfoundation.org
[39] https://www.microsoft.com
[40] Carsten Mai, private communication.

ISBN 978-3-95450-148-9