ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM021

STANDARD DEPLOYMENT OF MICROTCA AT THE EUROPEAN SPALLATION SOURCE

F. Chicken, J.J. Jamroz, J.P.S. Martins
European Spallation Source ERIC, Lund, Sweden

Abstract

This paper outlines the deployment strategies for more than 300 Micro Telecommunications Computing Architecture (MTCA) systems at the European Spallation Source (ESS) ERIC. These units are integral to the control systems spanning from the ion source to the instrument halls, encompassing Radio Frequency (RF), Beam Instrumentation (BI), Machine Protection (MP), and Timing Distribution (TD). As integration paths for these systems have matured, so have the deployment methods. This paper highlights the standardized deployment approaches for MicroTCA Carrier Hubs (MCHs) and Concurrent Technologies computers (CPUs), aimed at enhancing system maintainability, flexibility, and integration efficiency.

BACKGROUND

MTCA has been selected as the field technology to implement the distributed electronics along the ESS facility. The systems are being built with:

- 9U MicroTCA.4 system for physics with RTM,
- 3U MicroTCA.4 system for physics with RTM.

The MTCA-based ESS systems that were published in detail:

- Front-end electronics in [1],
- Generic data acquisition in [2],
- TD in [3],
- BI-nBLM in [4].

An initial successful outcome was confirmed during the beam commissioning in 2022 which was presented in [5].

Scope

The total number of MTCA systems to be deployed, once the facility is fully operational, is more than 300. Table 1 presents the ESS overview with the stakeholder breakdown.

Overview examples of the MTCA systems being built are presented in Figs. 1 and 2.

Integration

During the initial building of MTCA [6] systems, the setup was defined by the integrator, and typically the firmware of

Table 1: The Distribution of MTCA Systems within ESS Machine

ESS ≈ 300			
RF	**BI**	**TD**	**MP**
175 x 9U	70 x 3U 15 x 9U	35 x 9U	10 x 3U

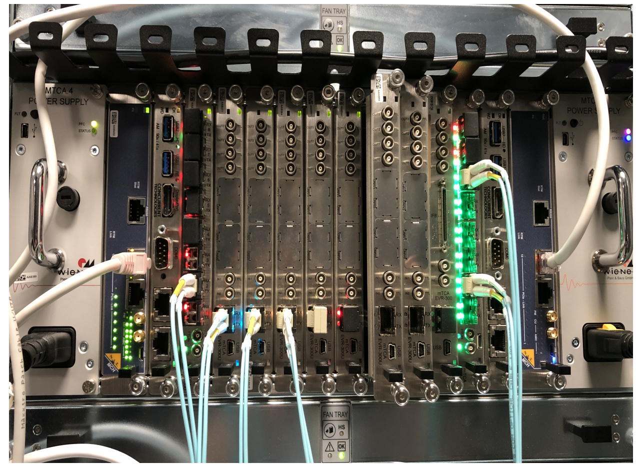

Figure 1: 9U MTCA system for TD.

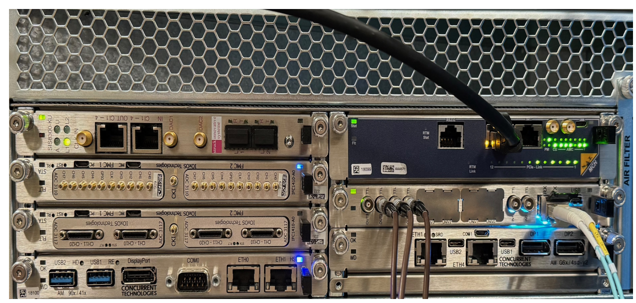

Figure 2: 3U MTCA system for BI.

the MCH ran the firmware version installed by the manufacturer and the Experimental Physics and Industrial Control System [7] (EPICS) environment installed might only have a few selected libraries, and kernel drivers. This process was highly time-consuming, created many variations between crates, and was not well designed for easy updating or maintenance upon deployment. It also proved problematic for debugging issues that arose, given the array of Advanced Mezzanine Card (AMC), FPGA Mezzanine Card (FMC), and Rear-Transmission Module (RTM) being used across the different applications and stakeholders. The MCH with its connections is presented in Fig. 3.

MCH DEPLOYMENT

The MCH model used at ESS is the NAT-MCH-PHYS board [8]. The MCH has an initial communication using an RS232 connection to a Moxa Technologies Serial-to-Ethernet device (MOXA), to allow a serial connection between the unconfigured MCH and the ESS network (presented in Fig. 4). The purpose of this is to read sufficient information from the MCH (board serial number (SN), me-

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM021

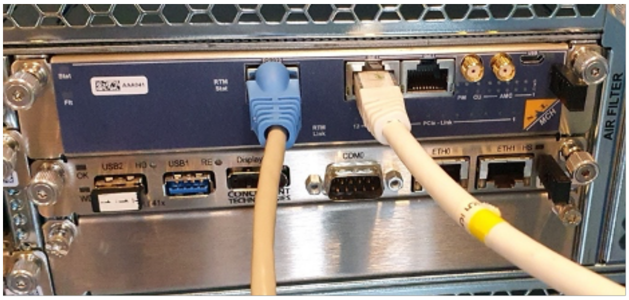

Figure 3: Picture of 3U chassis with MCH connected.

dia access control (MAC) address, and current firmware version), and create the necessary hostname within the ESS control system using a custom in-house designed application for Configuration Management database [9] (CSentry), and be assigned an Internet Protocol (IP) address within the chosen network. Once communication via the new IP address is possible, the next step is for the firmware to be installed and the base configurations and Peripheral Component Interconnect Express (PCIe) virtual switch to be set. The configuration is dependent on the chassis type used (either 3U or 9U rack-mount type) as the backplane differs between the chassis.

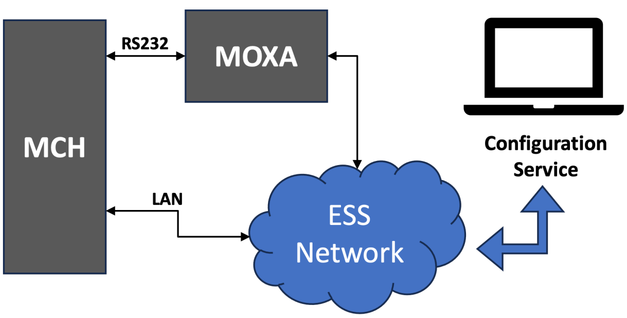

Figure 4: Schematics of MCH connections.

Development

The automated deployment of the MCH has gone through 3 main developmental phases.

Initial Deployment The first version of this process was a command line tool in Bash with a simple HTML graphical user interface (GUI) which was just a wrapper for the command line. This gave a web-based interface that was simple for anybody needing to deploy an MCH. The interface is presented in Fig. 5.

Second Version A second more advanced version used Jupyterhub (a multi-user web server for Jupyter notebooks) as the interface. The Bash was converted to Python programming language, which again could be run through the command line but this time with "python3" needed as a prerequisite. Otherwise, the Jupyterhub notebook provided a nice GUI, with easy changes to the MOXA IP address, port number, and backplane configuration possible. The Jupyter interface is presented in Fig. 6.

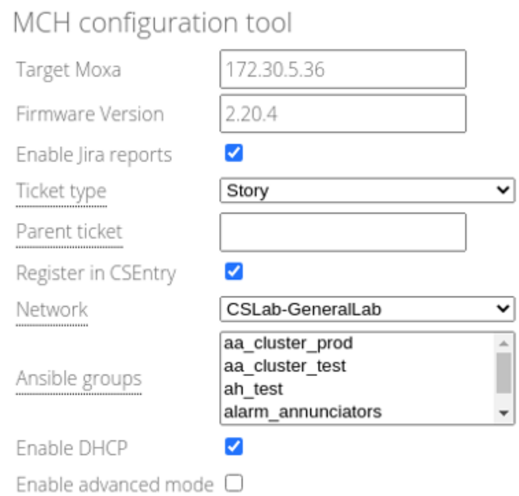

Figure 5: HTML GUI used for the first implementation.

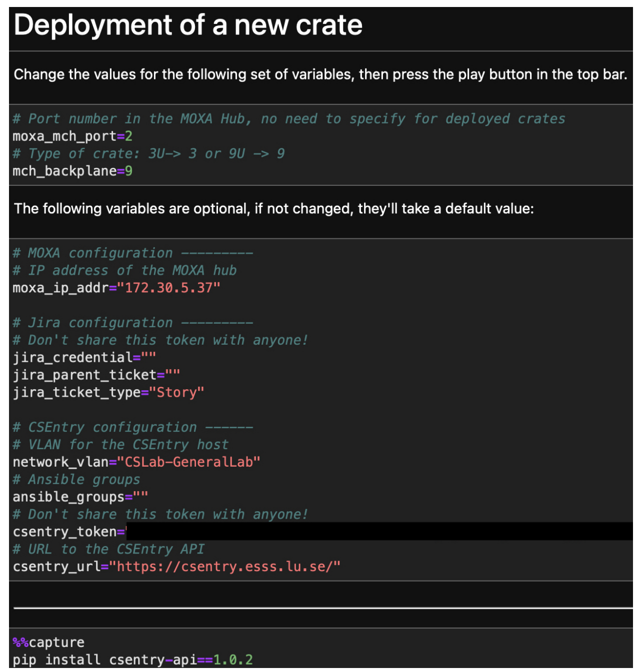

Figure 6: Jupyterhub Notebook.

Current Deployment Finally following stricter changes to the ESS IT security, with tightened changes to the firewall, it was unable to use the Juypterhub scripts to run the serial connection through the MOXA box to the MCH. Now deployment of this script is either through a virtual environment to run the Python scripts on a lab-based workstation or directly using Gitlab-CI (DevOps platform with continuous integration (CI)). The fixed IP address for the MOXA and the port number, along with the backplane type are set via the script command line arguments. The Gitlab-CI in action is presented in Fig. 7.

CPU DEPLOYMENT

ESS is using 2 CPUs both made by Concurrent Technologies [10], the now obsolete AM900, and its replacement

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM021

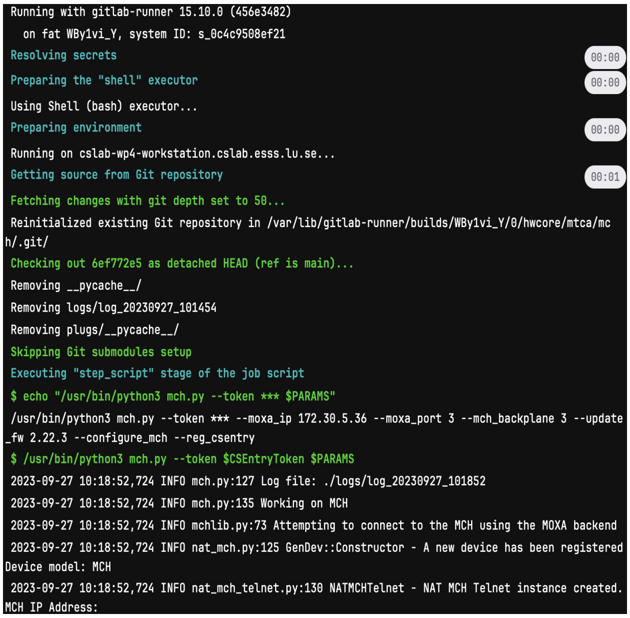

Figure 7: Gitlab-CI for MCH configuration.

AM-G6 version. The older AM900 version is mostly used by the TD as the processing requirements are lesser than those for RF or BI. Both boards are configured in the same way, only a few differences in the BIOS configurations are made.

Initial Deployment

At the start of the deployment of MTCA, the CPU was flashed with a full version of the CentOS 7 operation system (OS) via a pen drive. This gave the CPU a GUI when a monitor was connected, and a generic login was typically used to avoid multiple accounts and passwords being remembered. However, the EPICS environment needed manual installation which caused some mismatch on each device depending on who set up the CPU and which libraries were added.

Current Deployment

After some discussion with the ESS ICS Infrastructure team who were also deploying CentOS 7 on their servers and virtual machines, it was agreed that with some investigation into the Basic Input/Output System (BIOS) of the CPU, it would be possible to use their deployment tools and install them over the network.

These changes to the BIOS from the factory settings, mostly concerning the need to use a Preboot Execution Environment (PXE) upon starting, and setting the Unified Extensible Firmware Interface (UEFI) shell as the primary boot option. Within the BIOS options it is possible to expose the MAC addresses for that specific CPU board, needed to register the device within the CSentry application (see Fig. 8). When the device is registered, along with the list of users with administration level access, and hostname and IP are set, the Linux OS can be chosen (currently CentOS 7 but will be upgrading to a Yocto Project OS in the future). This

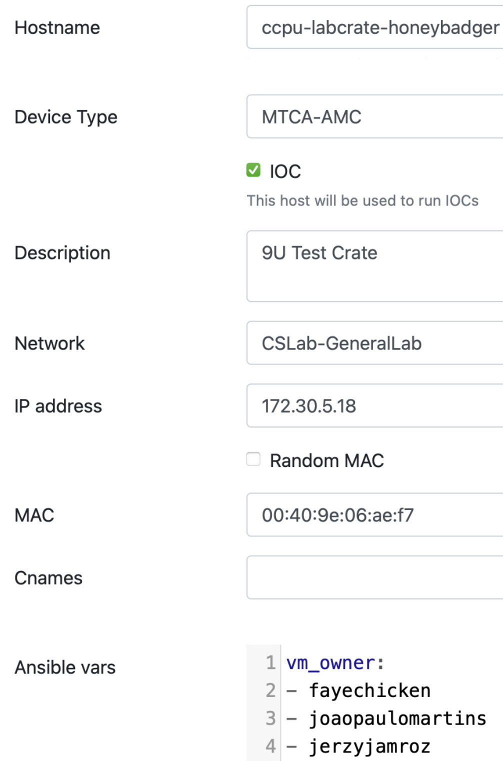

Figure 8: Registering a new CPU in CSEntry.

triggers the job to run using the Ansible (a suite of software tools that enables infrastructure as code) playbook to run the OS installation. A follow-up playbook is automatically triggered after the successful completion of the first installs of the ESS EPICS Environment (E3) with all the standard libraries and driver kernels. The Ansible in action is presented in Fig. 9.

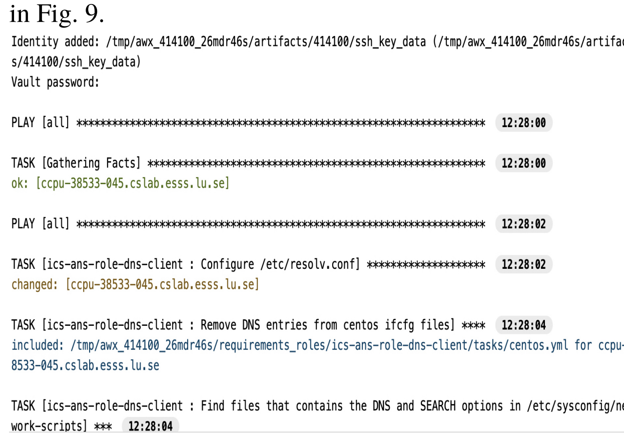

Figure 9: Job run in Ansible to install Linux OS and E3 playbook.

BENEFIT OF SCRIPTING AND PLAYBOOKS

The purpose of creating these scripts and playbooks was to first make the setup process the same regardless of who was building the crate. A driving requirement was to make the steps easy to follow so those unfamiliar with the MTCA

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THMBCM021

standard could assemble the base components. Another requirement was to ensure the future maintainability of these devices is improved, with remote access and deployment in mind. Now it is possible to run the scripts or playbooks as long as the system is connected to the network, and multiple jobs can be run simultaneously, instead of having to either manually install a new OS on each CPU or connect individually via the Secure Shell Protocol (SSH) to each MCH to make changes.

CONCLUSION

The deployment methods discussed in this paper represent a significant advancement in the deployment of MTCA crates at the European Spallation Source ERIC. By focusing on standardization, clarity, and future-proofing, these strategies enhance maintainability, accessibility, and integration efficiency across the control systems.

ACKNOWLEDGEMENTS

The author would like to thank Felipe Torres Gonzalez, Jeong Han Lee, Linn Berntsson, Peter Van Velze and Ross Elliot who helped with initial deployment tools and contributed massively to this drive for standardization.

REFERENCES

[1] J. Martins, S. Farina, J. Lee, and D. Piso, "MicroTCA.4 Integration at ESS: From the Front-End Electronics to the EPICS OPI", in *Proc. of International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS'17), Barcelona, Spain, 8-13 October 2017*, Barcelona, Spain, 2018, pp. 1692–1694. doi:10.18429/JACoW-ICALEPCS2017-THPHA133

[2] S. Farina, J. Lee, J. Martins, and D. Piso, "MicroTCA Generic Data Acquisition Systems at ESS", in *Proc. of International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS'17), Barcelona, Spain, 8-13 October 2017*, Barcelona, Spain, 2018, pp. 118–124. doi:10.18429/JACoW-ICALEPCS2017-TUAPL01

[3] J. Jamróz, J. Cereijo García, T. Korhonen, and J. H. Lee, "Timing System Integration with MTCA at ESS", en, in *Proc. ICALEPCS'19*, vol. ICALEPCS2019, 2019, USA. doi:10.18429/JACOW-ICALEPCS2019-WEPHA071

[4] Y. Mariette *et al.*, "New Neutron Sensitive Beam Loss Monitor (nBLM)", in *Proc. ICALEPCS'19*, New York, NY, USA, 2020, p. 136. doi:10.18429/JACoW-ICALEPCS2019-MOMPL008

[5] R. Miyamoto *et al.*, "Beam Commissioning of Normal Conducting Part and Status of ESS Project", en, in *Proc. LINAC'22*, vol. LINAC2022, 2022, UK. doi:10.18429/JACOW-LINAC2022-MO1PA02

[6] Micro Telecommunications Computing Architecture (MTCA), https://www.picmg.org/openstandards/microtca

[7] Experimental Physics and Industrial Control System, https://epics-controls.org,https://docs.epics-controls.org

[8] N.A.T GmbH, https://nateurope.com

[9] B. Bertrand, S. Armanet, J. Christensson, A. Curri, A. Harrisson, and R. Mudingay, "ICS Infrastructure Deployment Overview at ESS", in *Proc. ICALEPCS'19*, New York, NY, USA, 2020, p. 876. doi:10.18429/JACoW-ICALEPCS2019-WEAPP04

[10] Concurrent Technologies, https://www.gocct.com