ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BC001

FIVE YEARS OF EPICS 7 – STATUS UPDATE AND ROADMAP*

R. Lange†, ITER Organization, St. Paul lez Durance, France
A. N. Johnson, S. Veseli, Argonne National Laboratory, Lemont, IL, USA
K. Shroff, Brookhaven National Laboratory, Upton, NY, USA
T. Korhonen, S. Rose, European Spallation Source ERIC, Lund, Sweden
H. Junkes, Fritz-Haber-Institute, Berlin, Germany
S. M. Hartman, K. Kasemir, Oak Ridge National Laboratory, Oak Ridge, TN, USA
L. B. Dalesio, M. Davidsaver, G. McIntyre, Osprey DCS LLC, Ocean City, MD, USA
G. White, SLAC National Accelerator Laboratory, Menlo Park, CA, USA
M. R. Kraimer‡, Self Employment, USA

Abstract

After its first release in 2017, EPICS version 7 has been introduced into production at several sites.

The central feature of EPICS 7, the support of structured data through the new pvAccess network protocol, has been proven to work in large production systems. EPICS 7 facilitates the implementation of new functionality, including developing AI/ML applications in controls, managing large data volumes, interfacing to middle-layer services, and more. Other features like support for the IPv6 protocol and enhancements to access control have been implemented.

Future work includes integrating a refactored API into the core distribution, adding modern network security features, as well as developing new and enhancing existing services that take advantage of these new capabilities.

The talk will give an overview of the status of deployments, new additions to the EPICS Core, and an overview of its planned future development.

INTRODUCTION

EPICS 7

The EPICS (Experimental Physics and Industrial Control System) software toolkit has been publishing regular updates of its major version 3 release series for more than 30 years [1].

In December 2017, the 10-year initial development phase of the next-generation EPICS Base culminated in a large, tedious merge operation. After testing and verification, the first release of the combined existing EPICS V3 code and the new modules (working title "EPICS V4") was published as "EPICS 7" [2].

pvAccess

The pvAccess (PVA) network protocol, which was introduced with EPICS 7, extends the features of the classical EPICS protocol Channel Access (CA). PVA provides the ability to transport user-defined arbitrary structures without the need to recompile client or server code. Its distinguishing advantage over similar industrial protocol standards is a highly efficient subscription mechanism: Both server and client have an identical copy of the complete structure. When an update is sent, only the changed elements of the structure are transferred over the network and updated on the target side.

Initial Goals

A few main goals for EPICS 7 were identified and set from the very beginning, as part of the "EPICS V4" development:

- Compatibility of the Input/Output Controller (IOC) software. To allow a non-intrusive and gradual update process for the many existing installations, the EPICS Process Database and all interface layers below shall not change.
- Scope continuity between CA and PVA. Installations shall be able to run pvAccess as a full replacement for the classical Channel Access protocol, across all networks and layers.
- Minimal configuration. Simply migrating an installation from CA to PVA shall not require complex additional configuration.
- User-level libraries in multiple programming languages for client and server. EPICS Base shall provide well-documented user-level libraries to allow the creation of PVA server and client applications in C/C++, Java and Python.

EVOLUTION AND ACHIEVEMENTS

Normative Types

Two of the major factors contributing to the success of EPICS have been standard record types in the Process Database and the standard compiled-in structures of the CA network protocol. Based on these two concepts, generic EPICS applications work across many installations.

To keep that advantage, PVA includes well-defined structures, called Normative Types (NT), which are a superset of the CA standard structures.

Normative Types enable the efficient creation of generic applications and ensure the functional compatibility between CA and PVA. While pvAccess allows arbitrary data

* Work supported in part by the U.S. Department of Energy under contracts DE-AC02-76SF00515 and DE-AC05-00OR22725.
† ralph.lange@gmx.de
‡ Died 31 January 2022

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO01

structures to be transmitted, control system components can only efficiently communicate if there is agreement on how the structure for a fundamental value of a Process Variable (PV) and the associated metadata, e.g., timestamp and alarm status, are expressed. PVA's Normative Types meet this need by defining standard structure types for scalar and array data types, covering numeric, enumerated and string data. These structures are very similar to the compiled-in types used by Channel Access. This equivalence allows straightforward mapping, which creates an easy migration path.

As shown in Listing 1, a double-typed NTScalar is a Normative Type that combines a "double value" with a timestamp, alarm state and display metadata like units and suggested precision. It holds the same information as the DBR_CTRL_DOUBLE structure does for Channel Access, with a few additions like an optional "userTag" in the time stamp or a "description" and "form(at)" in the display detail, which are not found in the DBR_CTRL_DOUBLE.

```
double value
alarm_t alarm
    int severity
    int status
    string message
time_t timeStamp
    long secondsPastEpoch
    int nanoseconds
    int userTag
structure display
    double limitLow
    double limitHigh
    string description
    string units
    int precision
    enum_t form
        int index
        string[] choices [Dec, Hex, Exp, …]
control_t control
    double limitLow
    double limitHigh
    double minStep
valueAlarm_t valueAlarm
    boolean active
    double lowAlarmLimit
    double lowWarningLimit
    double highWarningLimit
    double highAlarmLimit
    int lowAlarmSeverity
    int lowWarningSeverity
    int highWarningSeverity
    int highAlarmSeverity
    byte hysteresis
```

Listing 1: Structure of double type NTScalar.

As mentioned before, pvAccess handles subscriptions to structures very efficiently: While a CA subscription for a DBR_CTRL_DOUBLE data type always transmits the complete structure with every update, a PVA subscription will only transmit the parts of the NTScalar structure that have changed. Fields like the "description" and limits tend to remain constant and will thus not contribute to the network traffic during PVA subscription updates.

In addition to scalar and array types, which are similar to CA data types, PVA introduces new Normative Type structures, e.g., NTNDArray and NTTable, which have no CA equivalents. The former holds an N-dimensional array, e.g., a 2-dimensional image, combined with metadata to describe the image and its format. The latter holds tabular data where the columns may have different types.

Normative Types are also available for other structures useful for scientific data. Examples include Normative Types for a matrix, a multichannel data source, histograms, etc. Details are provided in [3]. By using such Normative Types, clients can be configured rather than requiring specific coding. This is an extension of the long-time EPICS philosophy of providing a toolkit that engineers can configure to meet common control system needs.

The Normative Types specification [3] is meant to be open and extensible, allowing the addition of more types for structures deemed to be generic and of general interest.

For cases requiring data structures that cannot be encapsulated in a Normative Type, it is always possible to define a specialized structure for local use. In this case, both the server and the client need to be explicitly aware of the structure and its semantics. General-purpose clients will not be able to meaningfully handle or manipulate such a structure beyond simple display and storage. Nonetheless, this option provides the PVA mechanisms for efficient data transport and opens a door into a PVA-based ecosystem. This provides great flexibility for meeting any unique needs for transporting scientific data.

Integration in the IOC

QSRV is a pvAccess server that can be built into an IOC to enable PVA connections to the EPICS Process Database. Data from records can be served over CA or PVA with essentially the same content. However, the PVA normative type has more fields than the CA structure, including "description", a time stamp "user" tag, an alarm "message" and display formatting hints.

Individual records and their fields are automatically mapped to PVA PVs using suitable normative types. This straightforward "single" PV mapping does not require any specific configuration.

In addition, there is initial support for combining data from different EPICS Process Database records into custom "group" PV structures served by PVA. Compared to updates from separate PVs, the data in such a "group" PV can provide atomic updates.

Integration in Existing Applications

The general intent is to eventually transition all EPICS tools to the PVA protocol while supporting CA as long as possible.

The CS-Studio ecosystem exemplifies how EPICS tools aim to transparently support both Channel Access and pvAccess. As long as PVA data uses Normative Types, CS-Studio can switch between the protocols by changing the PV name prefix between "ca://" and "pva://". While "ca://" is for now the default, sites can already select "pva://" as their local default.

To end users, CS-Studio tools behave the same regardless of the underlying protocol. Data originating from records on an IOC or other Channel Access or PVA servers is

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO01

treated in the same way as long as PVA uses Normative Types for scalars and arrays.

The advantage of PVA starts to emerge for waveforms and images based on the “NTNDArray” data type. This Normative Type combines array values with metadata regarding compression and image encoding. In the past, such data had to be transferred via a collection of separate CA PVs. Using PVA, a single NTNDArray PV provides the complete information for compressed waveforms or images. Because of the standardized semantics of the Normative Type structure, this information is completely understood and used by the CS-Studio image display and waveform plot widgets.

New projects might be able to exclusively use PVA while existing projects will need to undergo a gradual transition from CA to PVA. For a record “X” on an EPICS 7 IOC, some applications will simply use “X” as a PV name, and the tool will then interact via either CA or PVA depending on the site-specific default protocol. Other applications will specifically use “ca://X” or “pva://X” to access the same data via the specified protocol.

An archive system that is configured to sample and store the value of PV “X” may initially use the default CA protocol and get updated to sample and store the PV “pva://X” later on. The CS-Studio Data Browser, a tool to show trends of PV values over time, has been prepared for this transition. When the user views a trace for the PV “pva://X”, the Data Browser will start to show its values based on that fully qualified name. In addition, it queries the archive for potential history data of PVs “ca://X” and “X”, assuming that data for past time periods has been sampled using those names, using different protocols. Combining the results, the Data Browser presents the user with a single trace for the PV that transparently bridges protocol changes.

Data Acquisition

Data Acquisition use cases require efficiency and high bandwidth from their network transport, combined with consistency between data and high accuracy time stamps across measurement, archive and retrieval.

The PVA Normative Types allow for efficient transport of structured data “out of the box”. As mentioned above, the NTNDArray, in addition to the array data, includes sufficient metadata to allow a PVA client to interpret and display the data. Array dimensions, data encoding, data type, etc. are part of a single structure, allowing a client to plot or process the data based solely on the information provided by the PV. Also, for that single structure, consistency at any time can be guaranteed. This enables features like data compression within the structure. E.g., CS-Studio supports lossless LZB compression, which can be enabled/disabled at the server end, while the client dynamically adjusts based on the provided encoding metadata.

PVA is used in production at multiple facilities for data acquisition and imaging. For example, the areaDetector module [4] supports the PVA transport of image data from a wide variety of detectors for x-ray scattering and imaging. PVA is used for the transport of neutron scattering time-of-flight / pixel ID event arrays. PVA is used for beam synchronous acquisition for pulsed accelerators to ensure a consistent data set across the machine for beam diagnostics.

Next-Generation User-Level Libraries

The initial pvAccess server and client implementations in EPICS 7 came in the form of C++ and Java libraries, including pvAccessCPP, pvAccessJava, pvDataCPP and pvDataJava. These implementations successfully demonstrated the desired functionality and are operational in production setups. Their design aimed for the same Application Programming Interface (API): the C++ and Java libraries contained nearly the same class names, method names and data types as permitted by the underlying programming environments. On the downside, these initial implementations were restricted to features that are common to C++ and Java, failing to take advantage of the strengths of each programming environment.

In 2019, a new Java implementation (core-pva) and a new C++ implementation (PVXS) were started, each concentrating on leveraging the advantages of the corresponding programming environment. For example, the original implementations included their own runtime type system. The Java implementation now uses its native type system and “instanceof”, while the C++ implementation makes full use of C++ templates to handle the various data types. Compared to the initial PVA implementations, the core-pva and PVXS APIs are quite different, but more natural to programmers who are fluent in the respective language. Frequent interaction among the PVXS and core-pva implementers asserts network compatibility – not only between the new C++ and Java implementations but also when interacting with the original implementations. The PVA protocol documentation has been reviewed and updated throughout the re-implementation of the protocol.

Both PVXS and core-pva now meet the desired requirements. While the original implementations are in use and still maintained with bug fixes, the new implementations are clearly seen as the way forward, as demonstrated with support for IPV6 and security detailed below. The core-pva Java library is used in CS-Studio. At the time of this writing, the EPICS 7 IOC still defaults to the original C++ PVA libraries, but the PVXS library already offers a complete set of command line tools and an alternate “QSRV” integration into the IOC. The “P4P” Python binding to PVXS allows convenient access from Python. Finally, a PVA gateway based on P4P and thus PVXS allows PVA traffic across different networks.

Middle-Layer Services

EPICS 7 supports high-performance data services, as a consequence of pvAccess supporting complex data structures and being able to request a PV value subject to arguments – implementing in effect a Remote Procedure Call (RPC) access method. This RPC access method is tailored to request/response type operations. It differs from the

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO01

common data access methods (read, write, subscribe) in allowing the payload for request and response to be different and to change for each execution.

Examples include simple PV name resolution by matching query (e.g., regex), retrieval of physics lattice and beam models, and interfaces to relational databases like Oracle for operational data, such as magnet polynomial measurements or device commissioning status. A complex suite of EPICS 7 services for FACET at SLAC interfaces modern Python and MATLAB physics applications to many legacy physics systems such as synchronous orbit acquisition, linac klystron energy profile control, and even operation of an old Fortran-based multi-modal experiment manager. This support was primarily included in EPICS 7 in prediction of big-data siloing and high-performance computing off the control system network, as is now taking shape where ML, large numerical, and multi-particle modeling systems hosted in high-performance clusters interface to the production control system.

In a different case of an EPICS 7 service, conceptually closer to control, the RPC access method is used to implement complex write operations with transactional behavior patterns. In ITER's Magnetics Diagnostics system, which measures magnetic field lines and plasma current, the complete configuration for 74 FPGA-based controllers (5 kB of data each) is written as a single specialized PVA structure to the system's master controller [5]. That controller implements the transactional behavior by verifying data integrity, configuring all FPGAs, verifying their health status and returning a single operation return status to the PVA client.

Increased Adoption and Use

As detailed in recent Collaboration Meetings [6], the EPICS Collaboration continues to grow with new projects and facilities adopting EPICS 7 and existing facilities choosing EPICS 7 for major upgrades. For example, at the Fermi National Accelerator Laboratory, EPICS has been selected as the control system framework for the PIP-II Project, the facility's flagship accelerator project for high-energy physics. At Brookhaven National Laboratory, EPICS has been selected for the Electron-Ion Collider Project, a next-generation collider for nuclear physics research. In both of these cases, EPICS 7 will be replacing an existing in-house developed control system framework, although with an expectation for EPICS integration with some remaining original control system components.

EPICS 7 is being used by numerous other large-scale science facility projects including the STS project at ORNL, the APS-U project at ANL, the ALS-U project at LBNL, the LCLS-II HE and MEC-U projects at SLAC, the ESS facility in Sweden, and the international ITER project. The addition of these large projects to the collaboration ensures long-term support for EPICS and will help to drive and fund the ongoing modernization of the toolkit to meet the needs of new facilities.

Small projects and university labs using EPICS add other benefits to the collaboration: these groups provide a vibrant training ground for students and the next generation of EPICS-aware scientists and engineers. The Fritz-Haber-Institute, for example, is developing a community practice for EPICS to support small-scale experiments and smaller facilities across the institutes of its organizational framework, the Max-Planck-Society. Numerous university laboratories have found the free open-source EPICS toolkit a cost-effective way to develop control systems, benefiting from the broader EPICS community.

ESS started using PVA as the default protocol for the control room applications (CS-Studio/Phoebus) at the start of the 2023 commissioning period. The control system ran under heavy network load with minimal issues. A crucial component of the ecosystem is the P4P module's PVA gateway: it delivers major performance improvements over CA gateways, especially in the ability to handle concurrent connections of mixed-sized data items, from scalars to multi-megabyte arrays, under high load.

Switching to PVA proved to be worth the effort, bringing a number of advantages over CA, with virtually no disruptions due to the migration.

Support for the RTEMS Real-Time OS

The RTEMS real-time OS [7] has been used as an alternative to VxWorks [8] for IOCs that control VMEbus systems, mostly running older EPICS 3.14 versions. Only very few facilities are building new systems based on VMEbus. EPICS 7 is used on RTEMS 4 in production in a few installations, but a reliable assessment of the maturity of that combination cannot be extrapolated from these few instances. In the past, progress in porting EPICS to the newer RTEMS versions 5 and 6 has been slow.

This situation has changed drastically in the last year, following a change in licensing policy for the VxWorks system. Some large facilities (APS, Gemini, Diamond) need to continue to use existing VMEbus hardware and drivers for facilities that are undergoing major upgrades and want to switch the real-time OS to RTEMS. Moving towards a strategic partnership, some laboratories started working with RTEMS developers to achieve full support for running EPICS 7 on RTEMS 6 on common PowerPC-based VME CPU boards, with support for the standard network protocols that EPICS installations use (NFS, DHCP/BOOTP, NTP/PTP). Support for NFSv4 has been tested, and a Board Support Package for the newer MVME2500 (QorIQ) VME CPU board is planned for the end of the year.

With full RTEMS 6 support, EPICS 7 can now be used on other single-board computers such as the Raspberry Pi, Zynq 7000 SoCs and other boards with real-time features.

Future developments planned for RTEMS include support for remote debugging with gdb and the support for dynamic loading.

Improved Documentation

Documentation coverage and quality are persistent issues in distributed collaborations and EPICS is certainly no exception to this. Some years ago, a new website [9] was launched and the development of another site for reference

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BCO01

materials [10] was started. However, the contributions remained few, mostly due to a lack of clear conventions and workflows. Managing the Content Management System of our main website, despite its popularity and nice front end, is challenging for an occasional contributor.

To improve the situation, a recent "document-a-thon" (a workshop series dedicated to improving the documentation) was used to decide on a new structure for the documentation, create a contribution guide and reorganize the existing websites. To make contributing easier for the EPICS community, central parts of the documentation sources are now hosted on GitHub and use the familiar GitHub contribution workflow. Processing, publishing and hosting are provided by the readthedocs.org service.

Not all documentation belongs in the central repository. Add-on modules should host their documentation with the source code and link to it from the central documentation hub. By using the same conventions, tools and styles, integration and common look-and-feel will improve and new users will experience a more efficient learning curve.

HAVE WE MET THE GOALS?

Five years after its initial release, EPICS 7 has achieved all major goals initially planned.

- The EPICS Process Database and the underlying IOC software did not see intrusive changes with the introduction of EPICS 7.
  Existing IOC software continues to work; the PVA server can be added to existing applications and runs in parallel to the CA server.
- Using Normative Types, PVA clients and servers support a superset of the CA functionality. Switching from CA to PVA can be done gradually – client by client, server by server – and smoothly.
- Straight replacement of CA by PVA needs no additional configuration on the IOC and only minimal configuration changes for generic clients.
- The second generation of user-level libraries in C/C++, Java and Python allows the creation of PVA server and client applications using modern, language-specific and well-documented APIs.

While the goals have generally been achieved, a few limitations still exist:

- pvData and pvAccess code requires newer C++ compilers that are not available on older, previously supported targets. The IOC parts of EPICS 7 are fully available and working, but the PVA protocol cannot be used on these targets.
- Links between EPICS Database Records on different IOCs still use the CA protocol. Migrating this functionality to use PVA is in the prototyping stage, but not finished.

NEXT STEPS AND FUTURE PLANS

IP Version 6

IPv6 is the current internet standard, but usage of the older IPv4 is still prevalent. In November 2020, government-funded sites in the USA were instructed to begin transitioning to an IPv6-only network environment within a few years [11]. While legacy control system equipment may continue to operate over IPv4 on isolated networks, EPICS tools running on "office networks" may have to support IPv6 in the very near future.

For TCP communication, the first step of converting to IPv6 is quite easy. Most C/C++ network code will accept hostnames with both IPv4 and IPv6 addresses, once calls to "gethostbyname" are replaced by calls to "getaddrinfo". Java and other programming languages have related network support – updating the address resolution will allow the code to support IPv6.

This has been demonstrated in a branch of the "ether_ip" driver used by EPICS IOCs to communicate with ControlLogix PLCs [12]. Tests with network tools show that the driver can now communicate with IPv6 addresses. At the same time, this example shows the difficulties to expect when updating control systems to IPv6: ControlLogix PLCs cannot be configured with IPv6 addresses yet.

For pvAccess, both PVXS and core-pva support IPv6. The address list used by PVA clients as well as the interface list used by PVA servers may contain explicit IPv6 addresses or hostnames that resolve to IPv6 addresses. By default, the IPv4 implementation uses UDP broadcasts for name resolution. IPv6 no longer supports broadcasts, but IPv6 multicasts with "link local" addresses are functionally equivalent and thus used for PV name resolution on IPv6. TCP name resolution is possible with both IPv4 and IPv6.

Support for IPv6 required no changes to the PVA protocol. While a few protocol messages related to name resolution contain IP addresses, these messages were designed from the start with sufficient space to hold IPv6 addresses.

With PVXS now supporting both IPv4 and IPv6, the PVXS-based PVA gateway can be used to bridge IPv4 or mixed control system networks to IPv6-only networks.

Secure Network Connections

A plan is in motion to update PVXS (C++) and core-pva (Java, in CS-Studio/Phoebus) to support secure network connections based on the industry standard Transport Layer Security (TLS) technology [13]. PVA clients that search for PV names will be able to indicate support for TLS authenticated and encrypted communications. PVA servers that support TLS will be able to accept such search requests and initiate the creation of a secure communication session. PVA servers that support secure connections will prefer TLS over regular unsecured connections. Server authentication will be accomplished by providing an X.509 certificate and optional client authentication will be achieved in the same way.

The proposed plan will use the subject of the supplied client certificate as the client name in the ACF authorization function. This can be used to control write access to

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TH1BC001

PVs. With integration to site authentication schemes (LDAP, AD, Kerberos), the retrieved principal will be used instead, providing greater flexibility when authenticating interactive users.

TLS name resolution may either individually query a list of PVA servers or may contact a PVA Name Server. The Name Server will be updated so as to not only resolve the name requests to PVA servers but also provide certificate status information to the server and allow the servers to use the OCSP stapling to optimize server certificate status verification in the client.

This new PVA network security will require a new configuration. The keys and certificates will need to be provisioned and stored securely. Eventually, the updated pvAccess protocol will provide robust authorization in an end-to-end encrypted, fully authenticated, efficient and manageable framework for control systems access. The implementation is planned to be completely backward compatible, with secure and non-secure clients and servers interoperating seamlessly.

Testers are encouraged to try out the code in the PVXS "TLS" branch and Phoebus, and give feedback.

This work is being undertaken over 2 years primarily by Osprey DCS, SLAC and ORNL, with Osprey DCS and SLAC contributions funded by a grant from the US Department of Energy for this activity.

CONCLUSION

All initial major goals have been achieved. Adoption and use of EPICS 7 and its newly introduced pvAccess protocol are picking up: new installations are using the new protocol from day one; existing installations are beginning to migrate their systems, following a seamless and smooth integration path that leaves no site behind.

EPICS users are gradually discovering the extended possibilities provided by pvAccess, creating new applications for use cases beyond the scope of traditional EPICS systems.

Current projects adding support for IPv6 and Secure Network Connections will cover upcoming IT issues and requirements. An open and extendable system of standard structure types ensures compatibility and portability of applications across installations and users.

After 30 years, EPICS remains the premier open-source toolkit for building control systems for accelerators and large experimental physics projects. With the successful release of EPICS 7, the Collaboration has provided a modernized implementation to ensure leadership for the coming decades.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the large and lasting impact that Marty Kraimer has made to the EPICS project and collaboration from their inception and for as long as he was physically capable. Marty passed away on 31 January 2022.

REFERENCES

[1] List of EPICS Base releases, `https://epics.anl.gov/download/base/`

[2] EPICS 7 release announcement, `https://epics.anl.gov/tech-talk/2017/msg01977.php`

[3] Normative Types Specification, `https://docs.epics-controls.org/en/latest/pv-access/Normative-Types-Specification.html`

[4] EPICS Support for Multidimensional Detectors, `https://areadetector.github.io/`

[5] A. C. Neto *et al.*, "Conceptual architecture of the plant system controller for the magnetics diagnostic of the ITER tokamak", *Fusion Eng. Des.*, vol. 96-97, pp. 887-890, 2015. `doi:10.1016/j.fusengdes.2015.06.063`

[6] EPICS Collaboration Meetings, `https://epics-controls.org/news-and-events/meetings/`

[7] RTEMS Real Time Operating System, `https://www.rtems.org/`

[8] VxWorks Real-Time Operating System, `https://www.windriver.com/products/vxworks`

[9] EPICS Website, `https://epics-controls.org/`

[10] EPICS Documentation Website, `https://docs.epics-controls.org/`

[11] "Completing the Transition to Internet Protocol Version 6", Memorandum M-21-07, Executive Office of the President, Office of Management and Budget, Washington (D.C.), Nov. 19, 2020.

[12] K. U. Kasemir and L. R. Dalesio, "Interfacing the ControlLogix PLC Over EtherNet/IP", in *Proc. ICALEPCS'01*, San Jose, CA, USA, Nov. 2001, paper THDT002, pp.. 481-483. `https://github.com/epics-modules/ether_ip`

[13] G. White, M. Davidsaver and G. McIntyre, "Accelerator Systems Cyber Security Activities at SLAC", presented at the 19th Biennial International Conference on Accelerator and Large Experimental Physics Control Systems, Cape Town, South Africa, Oct. 2023, paper TU2BCO04, this conference.