



MEPhIST-0 integrated control and data acquisition system

N. Efimov^{*}, F. Podolyako, A. Prishvitsyn, S. Krat

National Research Nuclear University MEPhI, 115409, Kashirskoye shosse 31, Moscow, Russia

ARTICLE INFO

Keywords:

Fusion
Tokamaks
EPICS
CSS
Data acquisition
HDF5

ABSTRACT

The growing fusion research industry faces an extensive need for specialists with practical skills. However, the high complexity of the relevant experiments makes it difficult to maintain laboratories in which students can get their first hands-on experience of interacting with relevant devices. One of the options is using small, simplified machines that are designed for low-qualified personnel. This paper presents the architecture and implementation of a control and data acquisition system for an educational tokamak MEPhIST-0. It integrates different plasma diagnostic systems as well as tokamak's control nodes using the EPICS framework. Different approaches to data archiving and access are discussed, and the overall design is presented from the perspective of a primarily educational installation, as well as the preliminary results.

1. Introduction

MEPhIST-0, which stands for the MEPhI spherical tokamak, is an educational tokamak, designed in National Research Nuclear University "MEPhI" (see Fig. 1). The major and minor radii of MEPhIST-0 are 25 cm and 13 cm respectively, with the radii ratio ~ 1.9 which puts it at the upper boundary of what can be named a spherical tokamak. It has the maximum achievable toroidal magnetic field of 0.8 T, the expected discharge duration of 40 ms, the plasma current of ~ 10 kA and the average electron density of 10^{12} cm^{-3} . The detailed description of this facility is presented in [1,2].

The primary mission of the MEPhIST-0 is to produce a cadre of new specialists for the rapidly expanding fusion industry. Following its educational paradigm, MEPhIST-0 is designed to be operated by students during scheduled laboratory studies and experiments for their qualification research. It requires the creation of control, data acquisition and archiving systems that should meet the following requirements:

- To be fully automated for remote control of the tokamak.
- To provide the possibility of operation by low-qualified personnel.
- To demonstrate and represent larger devices engineering solutions.
- To be flexible and easily extensible.
- To ensure personnel safety and danger situation handling.

The high degree of automation follows from the need to minimize the risks to an operator's health during experiments. In this context, remote control refers to the operator working from a control room in the same

building as the tokamak, but without direct access to the tokamak room. The ability to be operated by low-qualified personnel necessitates the presence of an "easy" control mode to give undergraduate students or even schoolchildren an opportunity to participate in sophisticated plasma experiments. An extremely successful setup example with such a fusion educational system is the GOLEM tokamak [3], which can provide different virtual control rooms according to the operator's skill level.

Being an educational device, MEPhIST-0 as well as each of its sub-systems should serve as some kind of demonstration mockup of larger more complex devices, such as T-15MD [4], JET [5] and ITER [6]. This means that the systems created for MEPhIST-0 should bear a resemblance to larger devices. Additionally, the design of this tokamak also assumes its use as a test site for some new devices. For example, the infrared heterodyne interferometer used in the MEPhIST-0 is the prototype that will be the basis for creating a multichannel interferometer for the tokamak T15-MD. Therefore, the control system needs to be scalable, which means the ability to quickly add new diagnostics. Finally, the control system of the MEPhIST-0 should provide personnel safety and danger situation handling via interlocks, emergency contactors and an alarm system.

Thus, this work is devoted to the description of the development and testing of the MEPhIST-0 tokamak's distributed control system, which includes pumping, charging and discharging of the capacitor banks for the magnetic coils, plasma parameters configuration, as well as experimental data generation, acquisition, postprocess, archiving and access.

^{*} Corresponding author.

E-mail address: NEEfimov@mephi.ru (N. Efimov).

<https://doi.org/10.1016/j.fusengdes.2023.113990>

Received 3 August 2023; Received in revised form 4 September 2023; Accepted 7 September 2023

Available online 12 September 2023

0920-3796/© 2023 Elsevier B.V. All rights reserved.

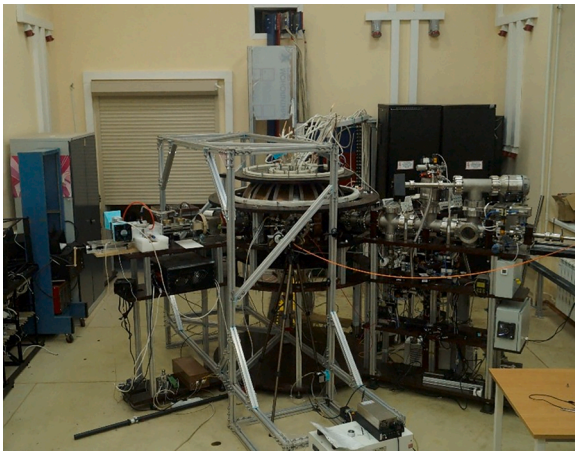


Fig. 1. MEPhIST-0 tokamak for educational and research.

1.1. Overall design of the MEPhIST-0 control system

The architecture design of the MEPhIST-0 control system (CS) is based on the open-source software platform called EPICS (Experimental Physics and Industrial Control System), widely used in various scientific projects such as CERN and LIGO [7,8] and particularly prevalent in tokamaks: ITER [9], EAST [10], JET, MAST [11], KSTAR [12], COMPASS-U [13], GLAST [14] and many others.

EPICS allows one to design a distributed system, where every node acts independently so the CS can be resistant to failure of its parts. Basically, the system structure consists of Input/Output controllers (IOCs), that directly interact with physical devices, the “Channel Access” network communication protocol (or “pvAccess” since EPICS 7 [15]), that acts over TCP/IP and allows centralized access to IOCs, and clients like operator displays, databases, alarm systems, etc.

Each IOC contains a memory resident database with a set of process variables (PVs) of various types with some configurations (display precision, scan period, alarm values, smoothing and others), scanners, which regulates the processing of the records, device drivers for handling access to external devices and an interface between the IOC and other hosts, that transfers PVs states and values. After configuring IOCs for control and data acquisition devices, users can access them from any workstation in the local area network (LAN) using EPICS

protocols.

Fig. 2 presents the general structure of the MEPhIST-0 control, data acquisition, storage, and management system. The lifecycle of the data starts in the right part of the scheme and ends in the left one. Each control or data generation device or group of devices (Fig. 2, segment 1) have its own control nodes (Fig. 2, segment 2) equipped with EPICS base and configured IOCs. These control nodes act as interface converters, transforming communication protocols of devices into the EPICS format. This allows reading and writing of data from devices using a unified EPICS Channel Access protocol (CA). The approach of separate control units primarily aims to enhance fault tolerance, but it also attractive in terms of cable management and data transfer, especially when devices are physically separated by tens of meters.

EPICS, being a cross-platform system, can be installed on various architectures. For control units small “ARM” based single-board computers (SBCs) like Raspberry Pi and more conventional “x86” PCs with Ubuntu or Debian Linux distributions are involved. SBCs appeared to be highly attractive for integrating hardware into EPICS due to their cost-efficiency and compactness. However, their fault tolerance may be limited compared to more common industrial computers. Nevertheless, in our case no failures were observed in the SBCs operation during several months of tests. It also can be seen from the scheme in Fig. 2 that in some cases, when a physical device acts over ethernet, the function of the control node is performed by virtual machines (VMs), based on the “Proxmox” virtual environment. Virtualization technology is used in many segments of the created CS in order to isolate software tools execution from each other, which also increases the system’s fault tolerance.

To exclude the influence of system segments on each other, all communication channels are galvanically isolated. Thus, LAN is designed using gigabit fiber optic lines and gigabit network switches with optical transceivers. The connections of the control nodes to the tokamak devices are also made with a galvanic isolation by fiber optics or optocouplers. As was mentioned above, all devices in LAN can get a direct access to PVs of the control nodes, so the two left blocks on the scheme acts independently.

Almost all procedures related to automated processing, transmission and archiving of data are carried out in the segment 3 of the scheme, which is designed as one compact server rack with 3 servers and the 10 Gb/s Ethernet interface between them. The tokamak can be operated from any local workstation with the installed EPICS or the “Control System Studio/Phoebus” (CSS) platform, that is intended for creation

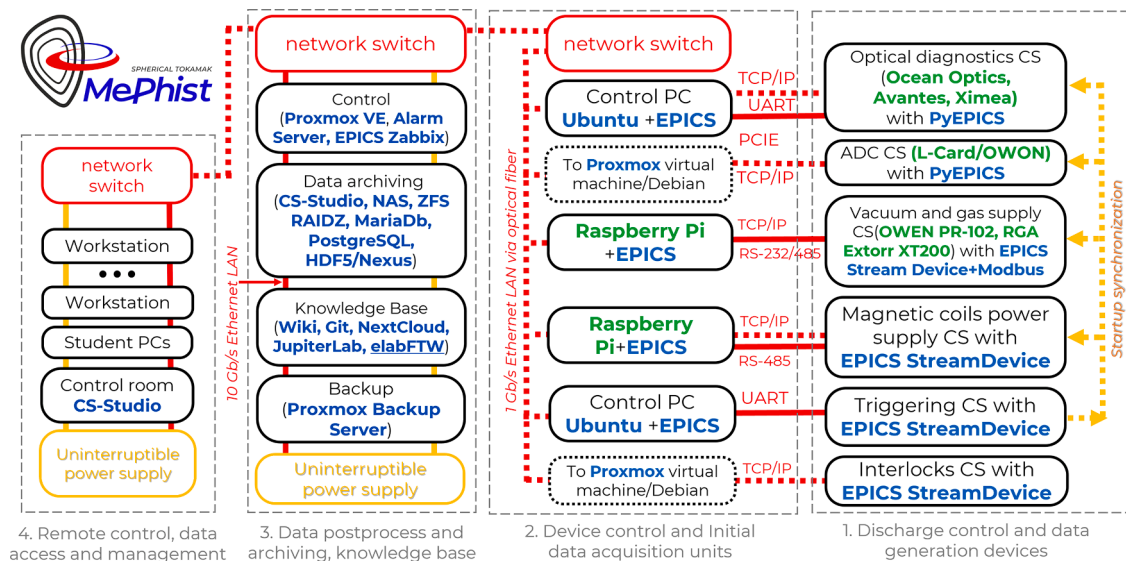


Fig. 2. General structure of the MEPhIST-0 control system. Green color – hardware, blue color – software, red color – interfaces, orange color - power and startup synchronization circuits.

and launch of graphical user interfaces (GUI) of control and data acquisition nodes. In the same way, the live and historic data access can be provided by CSS or web interface on any local computer. The third and fourth segments of the scheme actually may act separately from each other and most of the control functions would work even in case of second's segment failure. In the same way, the data acquisition and archivation will work, even if the control room is offline.

2. Data generation in control devices and plasma diagnostics

Being an educational platform, MEFHIST-0 is equipped with a representative set of basic plasma diagnostic tools: Langmuir probes, a set of electromagnetic sensors (flux loops, Rogowski coils, magnetic probes), a microwave interferometer, a four channel X-ray spectrometer, an optical spectrometer, and a high-speed camera. In the context of this paper, the plasma diagnostics and other sensors are of interest from the point of view of the load on the archiving system.

The acquisition of data, generated by plasma diagnostics, occurs according to the algorithm shown in Fig. 3. Before the initiation of a plasma shot, an operator may change some measurement parameters in the diagnostic's GUI and the CSS will send updated values of PVs to the diagnostic's IOC via the CA protocol (arrow 1 in Fig. 3). Like all other tokamaks, MEFHIST-0 works in a pulsed regime with extremely low duty

cycle. The average duration of the discharge does not exceed ~10 ms now and is not expected to exceed ~40 ms. Thus, start of measurements must be carried out at the strictly defined time, which is done by the external triggering of the plasma diagnostics. For most diagnostics, it can be provided in manual and automatic mode. In the first case an operator chooses which diagnostics should be launched by calling their "RUN" PVs before the start of the triggering system. In the "automatic" mode the diagnostic's IOC keep the "external triggering" regime of their devices enabled all the time and monitor the state of the "start" button that launches the triggering CS. If its state has been changed to 1, IOC will send a data request to a device.

There are several ways for EPICS to communicate with a device. The most straightforward way for devices with the "byte stream" based communication interface is to utilize EPICS "Stream Device" library together with the "Asyn Driver" module. Its choice is most common if the device only has a communication protocol with no application programming interfaces (API). Otherwise, it is possible to engage a special interface for API integration. Since most of the MEFHIST-0 tokamak plasma diagnostic devices have APIs for the Python programming language, they are integrated in EPICS by the "PyDevice" interface [16], which allows arbitrary Python code to be called via EPICS records and to push data to IOCs database via the CA protocol.

The update of the "RUN" PV (or pressing "start" button in the

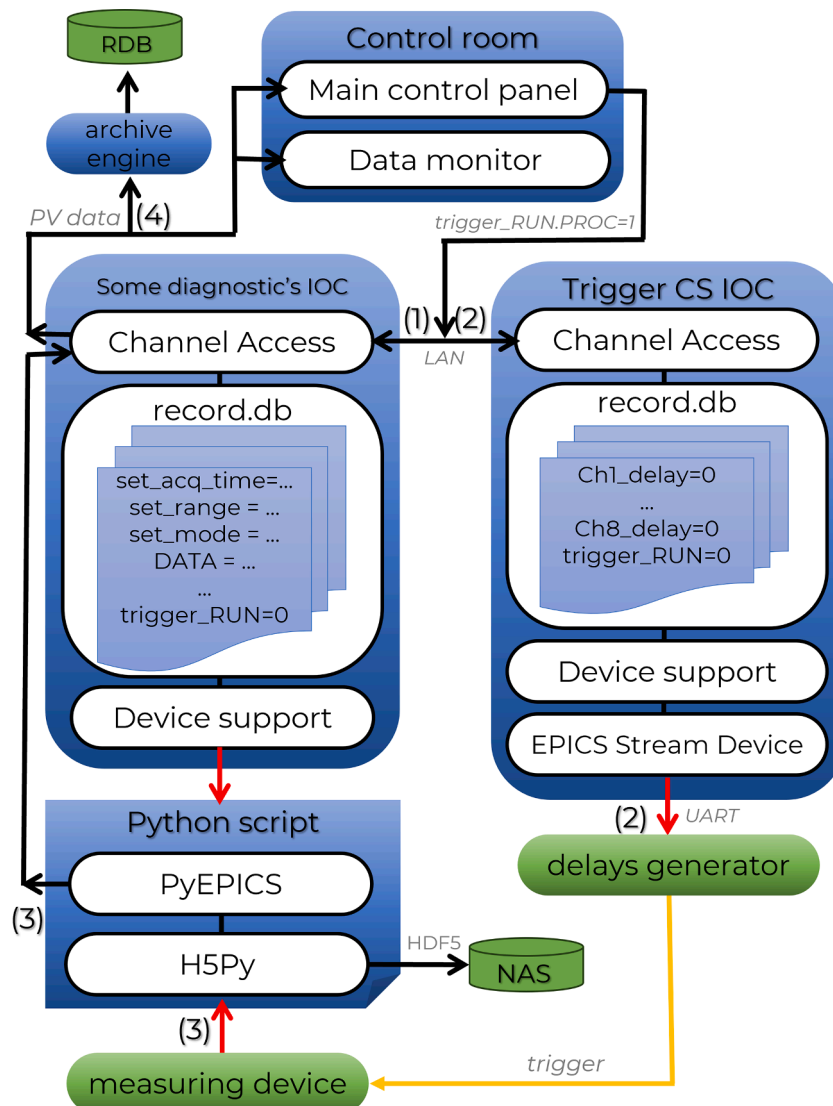


Fig. 3. The scheme of data acquisition.

automatic mode) thereby calls an external python script through the “PyDevice”. In turn the script directly connects and communicates with the device using some API. As a rule, it writes some configs and enables external triggering mode of a device and then just waits for this device to send some data. Once triggered, the device provides some measurements and then sends data to the Python script, that updates values of the corresponding data PVs (arrows 3 in Fig. 3). The triggering of all diagnostics, as well as the switching of the tokamak’s magnetic coils is done by a pulse delay generator, integrated into EPICS via Stream Device, as it is controlled by an Arduino Mega with a text communication protocol. So, by pressing the “start” button in the GUI an operator causes the triggering CS to launch the magnetic coils and diagnostics at the predetermined time (arrows 2 in Fig. 3). Finally, the control, visualization and archiving systems, that monitor PVs states periodically, provide further data processing (arrow 4 in Fig. 3).

2.1. Optical diagnostics

Analysis of plasma by optical methods is vital for any modern tokamak. MEPhIST-0 currently has 2 optical spectrometers, “Avantes AvaSpec” and “Ocean-FX”, which can measure time-integrated spectra in the range of 300–1100 nm with the resolution of 0,5 nm and 16-bit discretization. While the “Avantes” spectrometer utilizes the USB port for data transfer, Ocean-FX has the gigabit ethernet port. In both cases the communication is provided via Python scripts with the use of the “Avaspec.dll” [17] and “python-seabreeze” [18] libraries. The IOC’s database of each spectrometer stores one-dimensional arrays for the wavelengths, intensities of the source, background and postprocessed spectrum. The postprocessing, done with the EPICS “calc” module, mainly subtracts the background from the source spectrum. Thus, each spectrum occupies up to 6400 integer numbers.

The tokamak is also equipped with the fast camera “Ximea”, which has the ultimate framerate of 10 kfps. It is connected to the control unit with the PCIe bus through an optic fiber. The communication is done using the “xiApiPython” library [19] similar to the way shown in Fig. 3. The data in python scripts is stored as a two-dimensional array or several arrays if an obtained image is in color. Then it can be saved as conventional video with the “OpenCV” library [20] or transferred to EPICS IOC frame by frame. Actually, the camera generates the largest amount of data per pulse, since each video converted to h264 format takes about 3 MB.

2.2. Analog-to-digital converters

Most of the MEPhIST-0 diagnostics are analog, so their data acquisition can be unified with a set of analog-to-digital converters (ADC). At the moment there are 4 Rogowski coils, 8 voltage loops, 6 magnetic probes, 1 optical photodiode, 2 Hall sensors, 4 photomultiplier detectors of the X-Ray spectrometer, 2 sensors for measurement forward and reflected power of the microwave pre-ionization and one output of the microwave interferometer in the system. There are also some diagnostics, which are under development, so the total number of necessary channels is around 30. We predominantly use two “L-Card E-502” [21] for signal acquisition. It has one 16-bit ADC and a multiplexer, allowing for the configuration of up to 16 differential channels with an ultimate acquisition frequency of 2 MHz. The communication is also achieved with the Python API. So, the use of 2 ADC with the acquisition time of 30 ms and the maximum sample rate will generate about 120,000 floating point numbers per pulse. In addition to L-Card, to collect data from diagnostics where a larger bandwidth is required (as the magnetic probes in plasma disturbance monitoring mode), digital oscilloscopes with up to 100 MHz sample rate per channel are considered.

2.3. Vacuum, magnetic coils and triggering systems

In addition to data from plasma diagnostics, it is also essential to

archive data from several control and power systems that directly affect plasma parameters. These systems are the vacuum CS, that is also responsible for gas feed, the magnetic coils CS, and the triggering system, that allows to configure the startup time of different magnetic coils and diagnostics. The control of these systems is achieved using I/O modules, that support the Modbus protocol, and the Arduino platform, that can be accessed with ordinary text protocol.

Therefore, in all these cases communication with the physical devices is conducted through native EPICS “Stream Device” and “Modbus” libraries. The data to be archived includes pressure and gas fluence sensors of the vacuum CS, the capacitors charging voltage of the magnetic CS and different channels delay of the triggering CS. The data generation in these cases occurs continuously with the period specified in the corresponding PVs, which is typically between 1 and 5 s for most sensors of these systems. The amount of data generated by these systems per pulse is negligible compared to other diagnostics. Meanwhile, the vacuum system works 24/7, so the total data amount can’t be ignored. Overall, it has about 20 sensors, which corresponds up to ~ 1 million floating point numbers per day at the typical scanning period of 2 s.

3. Technical structure of data archiving system

In the MEPhIST-0 data acquisition system (DAQ) the initial data storage is carried out in random-access memory (RAM) of the control units. The data update is determined by a PVs configuration and can be either periodic or interrupt driven. Meanwhile the primary data archiving systems (DAS), located on a remoted server computer, continuously monitor PVs value changes and perform the data archiving to the permanent storage either on change or periodically.

The technical structure of DAS is shown in Fig. 4. It consists of three server computers HP Proliant DL160 G6 with a quite modest performance. Each has two Intel Xeon X5675 CPUs with 24 cores and 40 Gb of ECC RAM. Two servers are combined into a cluster with the Proxmox Virtual Environment. It allows to deploy multiple virtual machines (VM) on both servers, which is needed to isolate different services from each other. Such approach is actively used in data centers system administration. There are separate VMs for EPICS, where some IOCs are configured, for network attached storage (NAS) based on “Openmediavault” open-source platform [22], for relational database systems, for local domain names server (DNS) and etc.

The long-term data storage is based on the zettabyte file system (ZFS), where a RAIDZ is deployed with three 4 Tb HDD disks, giving overall 8 Tb of free space. The space is shared with DAS subsystems through iSCSI and NFS network protocols by the NAS. As can be seen from the scheme, there is also a backup system on the third server for saving configuration files of the CS, VM images and for the most valuable experimental data. It performs backup once a week and keeps two last backups.

While most archiving services are located on the first server, the second one is supposed to host services, providing a local knowledge base, and supporting infrastructure like local domain name system (DNS), which allows to access local web-services by an easy-to-remember string address instead of a set of IP address numbers. The knowledge base performs two functions: providing maintenance personnel with up-to-date information about the control system operation and status and serving as an educational platform for students with information about tokamak design, data access and analysis, online testing systems and so on. For example, the local wiki system accelerates the integration of new staff and contain guides for students during routine laboratory studies. The presented design has scalable structure, which allows to add new services to the existing system via separate VMs without affecting the operation of current ones.

4. Data acquisition, storage, management and access

A data acquisition system is an essential component of any

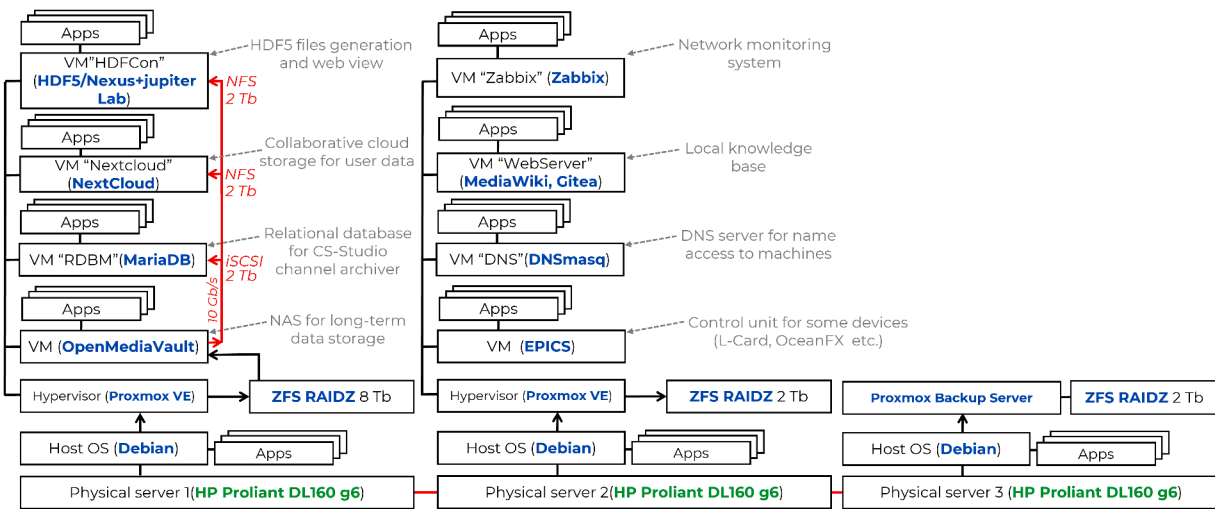


Fig. 4. Technical structure of DAS.

experimental research. It is responsible for aggregation of signals produced by various diagnostic systems, sensors, and other equipment. Such system also may be utilized for signals postprocessing, such as smoothing, background and noise removal. It systemizes and sorts the signals for the more convenient subsequent analysis. Finally, it transfers the data to the long-term storage.

As was mentioned above, the initial data storage is performed in the IOCs database of the control units, and it is updated either periodically or with event handling. The mission of the primary DAS is to collect the data from all units, combine them in some way and save to the permanent storage in the NAS. In general, all data of the tokamak can be divided in two categories: continuous and pulse data. The first one includes the data from the vacuum system, interlocks sensors, some temperature sensors. The second category contains data, which is generated during plasma discharge in the tokamak or related to the discharge parameters (like capacitors charging voltage, working gas type and pressure and presets for trigger signal delays).

There are two common ways of channels archiving. The first one is based on the relational database (RDB) and is suitable for large number of channels and continuous archiving. The second type assumes file storage and is more intended for diagnostic and experimental data. It is designed to handle a substantial amount of data, but with fewer number of channels and with strict separation of data from different discharges. It is quite usual to use several archiving tools with different storage types together. Thus, in KSTAR tokamak most of the experimental data is stored in MDSplus and HDF5 formats, while the machine real time operation data is processed in EPICS [23]. Similarly, the COMPASS-U database utilizes MySQL for metadata, HDF5 for the raw data and MP4 for videos and also uses the ZFS as the storage [13].

Thus, the MEPhIST-0 tokamak DAS also engages several approaches. A RDB is used predominantly for continuous data, logs and for all experimental data, while an HDF5 file storage would be used exclusively for storing data from plasma diagnostics and metadata, describing the plasma shots conditions.

4.1. Relational database

The use of EPICS for control is well-established. Meanwhile, its application for data archival is less developed. In the design of the MEPhIST-0 DAS the ITER project was chosen as the exemplar, being a kind of trend maker for the modern fusion. The ITER CODAC standard suggests the use of the archiving system, embedded in the Control System Studio (CSS) framework [24], that is already used in MEPhIST-0 for control, as it provides easily configurable GUIs, allowing to read and

modify PVs values. For the MEPhIST-0 tokamak the latest variant of CSS, also known as Phoebus, was chosen [25]. Its main difference from the older version, that is planned to be used in ITER, is that it no longer depends on the “Eclipse” platform. CSS is actually a collection of tools, intended for large scale control systems operation and monitor. The archive system of the CSS is called BEAUTY from the “Best Ever Archive Toolset, Yet”.

BEAUTY consists of the archive engine, CSS graphical user interface (GUI), and a relational database. The principal scheme of the archiving process, implemented in MEPhIST-0, is shown in Fig. 5. BEAUTY’s archive engine continuously monitors the states of EPICS records via the CA protocol and provides the aggregation of PVs values into RDB either periodically or when its value exceeds the preliminarily configured dead range, which allows to cut off the noise signal. It also may archive some metadata together with the PV values: timestamps, alarm status, units, limits and etc.

BEAUTY architecture supports configuration of separate archive engines, as an example, for several control groups with individual web monitoring tools, that may act independently. In our case four engines were configured: for the vacuum system, magnetic coils power supply system, ADCs and the optical diagnostics. The configuration of the PV for the specific archive engine is provided via the XML files, which are then loaded in the BEAUTY. The use of several archive engines allows to implement a more flexible approach in data logging. Thus, it makes possible to disable specific engines during some adjustment works, when the data, obtained by the corresponding devices doesn’t make sense. As example, it is seems useless to archive magnetic probes signals during wall conditioning stationary discharges, so it could be disabled during this procedure.

The archive engine provides an automated data saving to the pre-configured RDB like MySQL, Oracle, and PostgreSQL. In our case two test RDB was used: MariaDB [26] and the PostgreSQL+Timescale [27]. The choice of the specific RDB has a lot of pros and cons. In our case it seems not so significant due to relatively small data flow. However, during beta testing of the DAS with MariaDB and PostgreSQL it became clear that RDB choice may alter the way of the data visualization. The CSS has two regimes of data browsing: “raw” and “optimized”. The second one is preferable in case of large amount of data in order to minimize the memory usage. It is appeared that in case of the PostgreSQL data optimization can be provided inside the RDB on the server, while for the MariaDB it is provided in the CSS on the client side, so actually the historical data visualization can be slightly different depending on the architecture, as the RDB and CSS utilize different point sampling mechanisms.

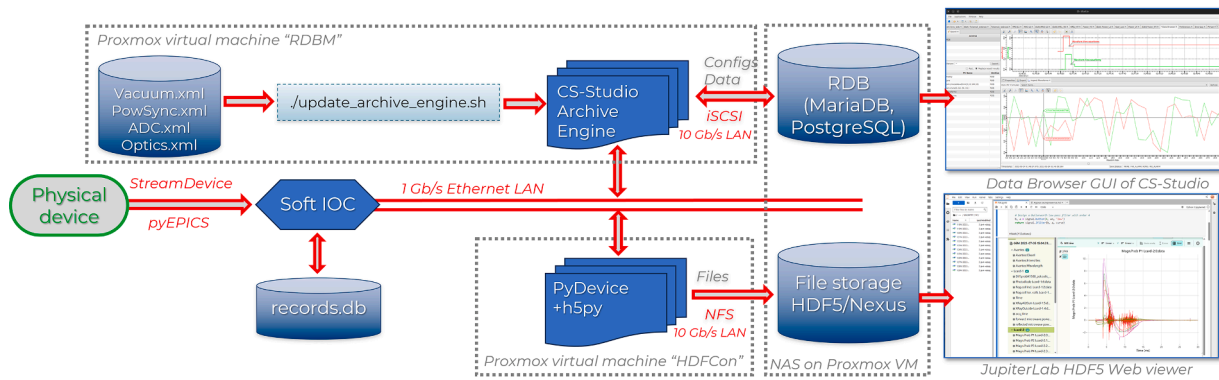


Fig 5. The scheme of MEPhIST-0 data archivation system.

In addition to simple integration into EPICS the use of BEAUTY is attractive due to native GUI “Data Browser” for live and historic data visualization and export, embedded in the same tool, as the control panel of the MEPhIST-0. For example, the Fig. 6 shows the visualization of the vacuum CS data during one working month. The windows on the left of this screenshot shows the loaded databases and PVs, filtered by the “VAC*” condition. The bottom window shows some statistics and also contains tabs for chart configuration and data export.

It seems that CSS is more suitable for visualization of the data over long periods of time, like the pumping curve or dynamics of some device’s temperature. The capabilities of array data visualization, as well as the combination of data for one specific plasma shot are limited and the system doesn’t provide enough flexibility in combining such data for any analysis, forcing the user to do it via export in third party software. Thus, for unexperienced user it would be tricky to find the oscillograms for the specific discharge, as it would need them to find the corresponding timestamp and filter PVs by it. Such approach might be not enough demonstrative and user-friendly for educational needs. In addition, RDB based storage limits the possibility of data sharing without providing a direct access to the internal control network and configuring a specialized software on the client side. That is why another system was developed, which perform archiving of plasma diagnostics simultaneously with the BEAUTY.

4.2. HDF5/NeXus

In fusion research the most common platform for experimental data storage is probably the MDSplus [28]. The project is used in many large fusion projects, the most known of which are: JET (UK), ASDEX-U, Wendelstein 7-X (Germany), EAST, HT-7U (China), KSTAR (South Korea), T-10 (Russia), C-Mod, DIII-D and NSTX-U (USA) [29–31]. As was mentioned above, EPICS platform is used for control in KSTAR tokamak, while MDSplus is intended for data acquisition and storage, and this approach is widespread. MDSplus uses file-based hierarchical data storage instead of RDB, which allows more straightforward data systematization and sharing. However, it seems that this platform performs much more functionality, than is required for our case, so another hierarchical data format called HDF5 [32] was chosen to work together with RDB.

The choice of the HDF5 is determined by its high prevalence in various scientific branches, which leads to the presence of a large number of tools for its processing and visualization. It can be presented as a container of datasets of various formats (text, digits, arrays, pictures and videos) with a metadata for each dataset, which makes it a self-descriptive and hierarchical format. Metadata may include the information about the data measurement conditions, timestamps and comments. The main reason for its choice is the possibility of integration into

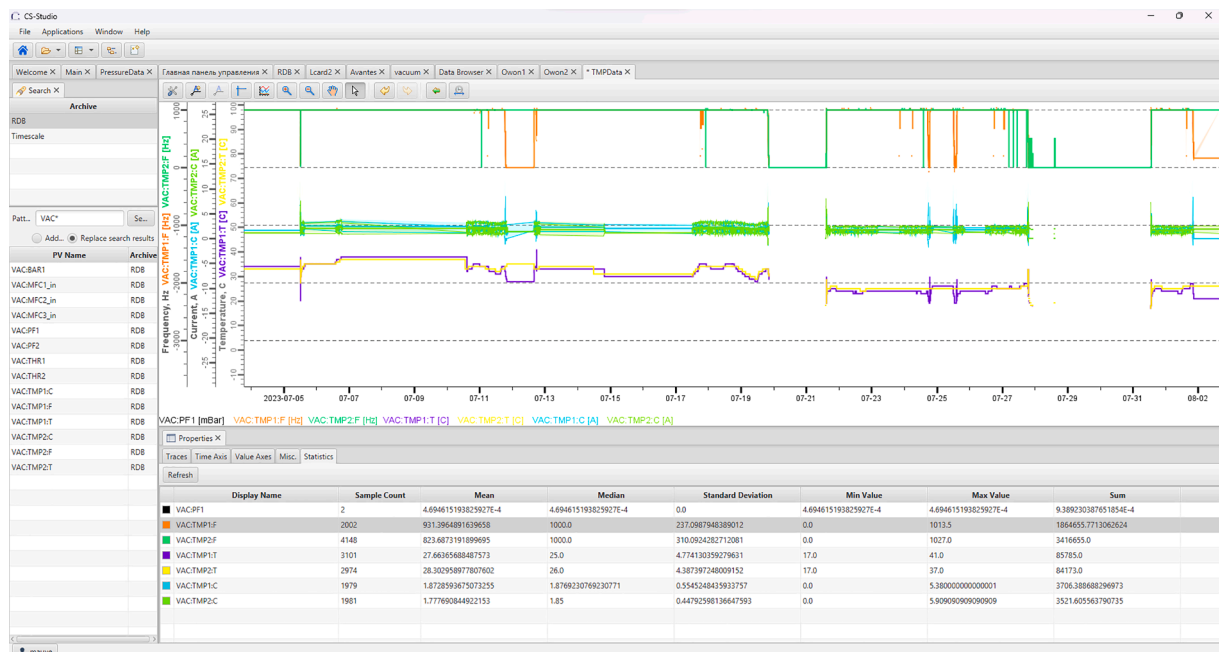


Fig. 6. The interface of the CSS “Data Browser” with the historical data of the MEPhIST-0 vacuum system.

common data analysis systems: OriginLab, Matlab, Scilab, Mathematica, Python, R and others. It allows an easy sharing of experimental data without the need for granting access rights to DAS as for RDB.

Thus, the HDF5 is considered as the main format for storing experimental data, related to plasma shots, in the MEPhIST-0 tokamak. For more convenient data visualization, the NeXus standard is utilized. Actually, it is built on top of the HDF5 format, but demonstrates more standardized approach in hierarchy formation, adding some rules for organizing domain-specific data, links structure, and field names, expressed in a form of base classes. This variation of the HDF5 is especially common in neutron, X-ray, and muon experiments. A detailed description of NeXus is given in [33]. Thus, further mention of the HDF5 format in the text means its NeXus version.

Unfortunately, EPICS doesn't have native HDF5 support, but there is the possibility of its integration via an interface for HDF5 in Python called "h5py" [34], covering all functionality of this format. So, by combining the "PyEPICS" and "h5Py" it is possible to monitor the update of PVs values and archive them in HDF5 file either with scripts, which perform a data and commands transfer between the IOCs and physical devices, as shown in Fig. 3, or with a separate script. The launch of a separate script is done with EPICS: there is a calc PV in the HDF5 IOC, that monitors the state of the "start" button and calls a script, if the value of the PV, corresponding to this button, has been set to 1. The script in each turn waits for some specific time (10 s as usual) and then collects data from the diagnostics via the CA protocol. Then it combines data in HDF5 file with the predetermined structure with "h5Py" library and saves in the file storage, located on NAS, through the network file sharing (NFS) protocol.

It was decided to create separate HDF5 file for separate discharges. A similar approach works in the Golem tokamak's archiving system, where each plasma shot is performed in a separate web page [3]. Meanwhile, in the GLAST tokamak, which is quite close in size and achievable plasma parameters to the MEPhIST-0, division into files occurs with an interval of one day [14]. It is not yet obvious, which approach is better, however, flexible data appending mechanism of the HDF5 allows to change this setting easily in the future. The name of each HDF5 file contains the discharge number and the timestamp. The data is stored in separate groups: vacuum, power & trigger systems, and for each measuring device. The header of each dataset contains metadata about the device and the channel name, which is crucial for multichannel ADC devices, where various sensors are combined.

For data visualization and analysis, the JupiterLab web interactive

development environment was deployed on a VM along with the HDF5 IOC. The HDF5 visualization is provided directly by the "H5Web" library [35]. It can be noticed here that the choice of NeXus over the raw HDF5 is mainly due to its better visualization with the "H5Web" library. An example of file structure and data visualization is shown in Fig. 7. The left panel of the window contains the file manager, allowing the selection of the year/month/day with a list of NeXus files with separate plasma shots. The right panel shows the interface of the H5Web for the discharge "20M" made on 2023.06.23. The left panel of the H5Web interface contains several groups (Avantes, LCard-1, LCard-2, Vacuum, PowSync) with datasets. The right part visualizes the data for the ADC "L-Card2". It can also be seen that names of the ADC channels allow for the identification of the signal source. Thus, the shown curves represent the signal of the loop voltages and the magnetic probes. Similarly, it is possible to open Avantes optical spectrum and the Ximea camera's frames.

The unique feature of the HDF5 and JupiterLab combination is the ability to perform data analysis directly in the web interface, as the H5Web also allows for easy input of each HDF5 dataset in Python. This fits perfectly with the philosophy of the MEPhIST-0 tokamak as an educational platform. For example, the first thing the students did after gaining the access to the HDF5 DAS during beta tests was to create a script, that automatically smoothes, integrates the Rogowski coil signals and plots the current evolution in the coils. The part of the script can be seen at the bottom of the central panel in Fig. 7. It allows to say that such approach is student-friendly, which gives hope that the created system will contribute to the formation of new personnel for fusion in a beneficial way.

At present, the MEPhIST-0 tokamak's experimental data can only be accessed via the laboratory's local network. However, efforts are underway to integrate the developed DAS into the "FusionSpace" platform. This platform will *inter alia* facilitate the exchange of scientific information, including experimental data from connected facilities, software repositories, codes, and standards, through a web portal [36]. Meanwhile, creation the analogue of the Golem's open access "shot database" [3] for our facility is under consideration.

4.3. Beta testing and data load estimation

The created DAS were tested for several months, during which experiments were carried out dealing with the modernization of the tokamak's electromagnetic system, integrating new plasma diagnostics, as

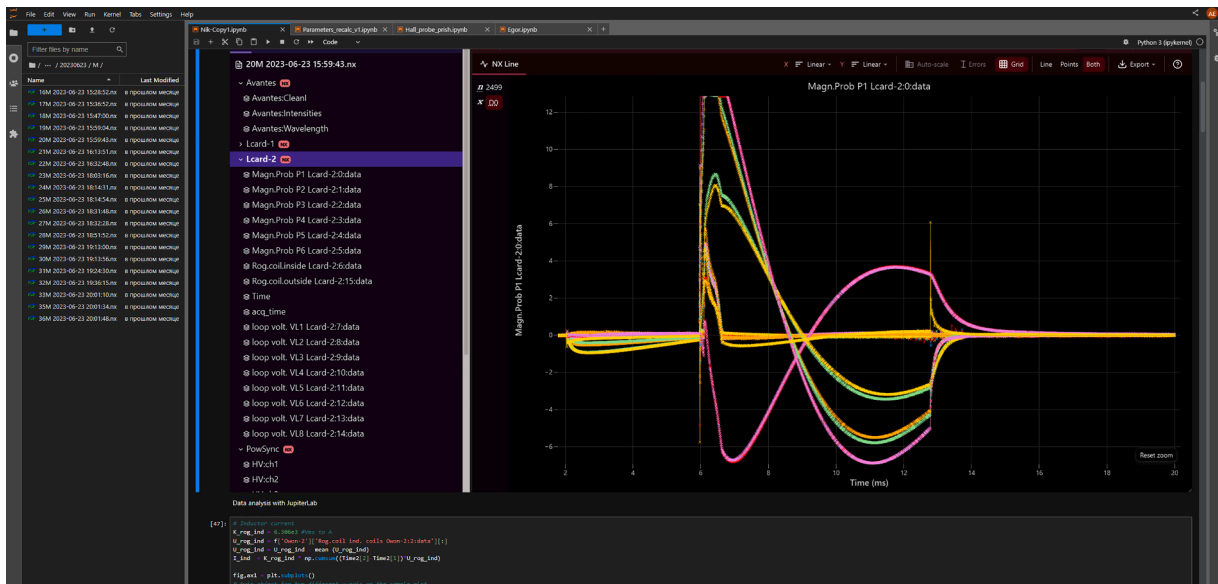


Fig. 7. Web interface of the HDF5 DAS data access and analysis based on JupiterLab and H5Web.

well as ordinary plasma discharges. As expected, the HDF5 DAS system became much more commonly used for experimental data output. Most of the users, who were mainly undergraduate and postgraduate students, created their own Python scripts in the web panel of the JupyterLab from the very beginning of the tests and used them for data visualization after some post-processing and combining. The integration of the DAS within the JupyterLab showcased high efficiency as an educational platform. It enables students to approach the same tasks through diverse methodologies, and combining their efforts facilitates the achievement of optimal outcomes.

For over 2 months of testing, more than 1000 test pulses were carried out along with about 100 plasma discharges. Fig. 8 shows estimations of data sizes for 1 plasma pulse and the data sizes for one month of work for different diagnostics. It should be noted that estimations for RDB are rather rude, as its functionality doesn't provide the sizes of separate rows. So, it is valid to compare the sizes of diagnostics, written in RDB, between each other, and to compare the overall RDB size with HDF5 storage. It can be seen that for both charts "L-card" ADC put the most load. Meanwhile, the overall size of the HDF5 is sufficiently lower, than a sum of all RDB channels. The main cause is that HDF5 storage was usually disabled during test pulses.

Taking into account that an average number of plasma pulses per working day is around 10, it can be estimated, that monthly data growth is no more than 2 GB. Thus, currently used data storage with the total size of 4 TB, based on ZFS RAIDZ, is redundant. Anyway, it is planned to add ADC with a sufficiently higher sample rate (more than 1 MHz per

channel), which will significantly increase the amount of data.

5. Typical operation procedure

Control System Studio (CSS) is used as the primary platform for operating MEFHIST-0 tokamak experiments. The choice of CSS is based on its native EPICS support and its status as a standard ITER CODAC utility for GUIs and data archiving [24], which meets the requirements of being the representative of the more complex devices. The last version of the CSS, that is called "Phoebus", has both a desktop application based on "JavaFX" and the Web version. Once the IOCs of all control units have been configured, access through CSS can be achieved using basic "JavaFX" XML files, that forms GUI scenes. CSS also supports scripting for automated procedures and subsequent data analysis.

The main control panel of the MEFHIST-0 tokamak is shown in Fig. 9. If, for any reason, the pumping system would be turned off, it can be activated via the vacuum panel in a separate tab. The operator is then required to check the status of the DAS in the main control tab, as well as the plasma diagnostics groups necessary for the experiment. The MEFHIST-0 tokamak electromagnetic system is powered by capacitor banks, which need to be charged before each plasma shot. So, the operator needs to set up the charging voltage, open the emergency contactors in the main control panel and enable charging. The process is automated. Meanwhile, the operator can configure another experiment parameters by adjusting the time delays between thyristors opening of various coils batteries and the pre-ionization startup, as well as the gas type and pressure, controlled by the capacitance gauge.

After all presets and charging completion, which can be seen by the corresponding indicator in the main control tab, the operator only has to press the button "start". After the pulse, the Data Acquisition System (DAS) automatically combines and sorts experimental data for the discharge, and then archives it for future access. Some data will automatically be displayed on the CSS tabs: there are separate tabs for the optical spectrometry, electromagnetic sensors and fast camera. After the pulse, the operator must close the emergency contactors as some capacitors may experience reverse polarity due to inductive energy recovery. The process can then be repeated. Thus, the average time for one plasma shot is now ~20 min, most of which is due to charging.

Currently, the fully automatic workflow is not yet completed, as the system is in the beta testing stage. However, it can be relatively easily implemented through scripts in CSS. The same applies to interlocks, which are mechanisms for automatically blocking potentially harmful components of the tokamak. While the system is still in the process of implementing automatic interlocks, alert windows are already functional. Consequently, although CSS is not yet suitable for operators with limited experience, it possesses the technical capabilities to simplify and enhance safety measures as much as possible.

6. Conclusion

The control system, as well as the data acquisition, archive, access, and analysis systems, have been designed, created, and tested. The systems have proven to be highly functional during the testing and operational stages. The control system is based on EPICS and complies with ITER CODAC standards, thus fulfilling the requirements for MEFHIST-0 to serve as a fusion educational platform and provide exemplary practices. The data archiving system, which supports the entire data lifecycle, is implemented using two approaches: the ITER-style BEAUTY system with RDB as the storage, and the HDF5/NeXus files. RDB is found to be most efficient for processing real-time operational data and has the better protection against data loss, while HDF5 storage is better suited for experimental data as it offers easy data access and analysis for inexperienced users. The reliability of the developed system was evaluated by performing more than 1000 shots, no significant problems were identified.

The involved software solutions (EPICS, CSS, HDF5) are intended for

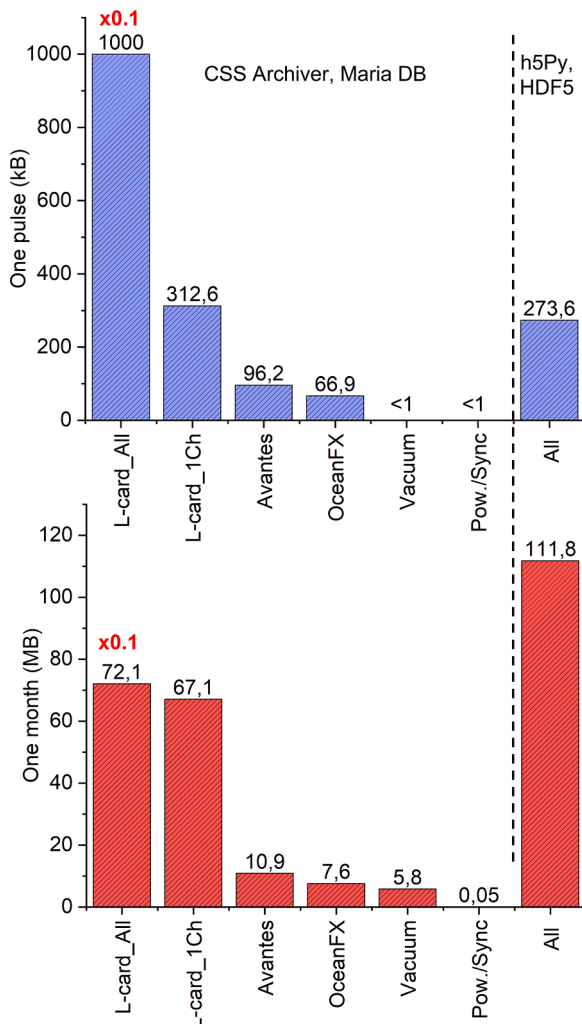


Fig. 8. The size of the archived data for one discharge and for a month of work.

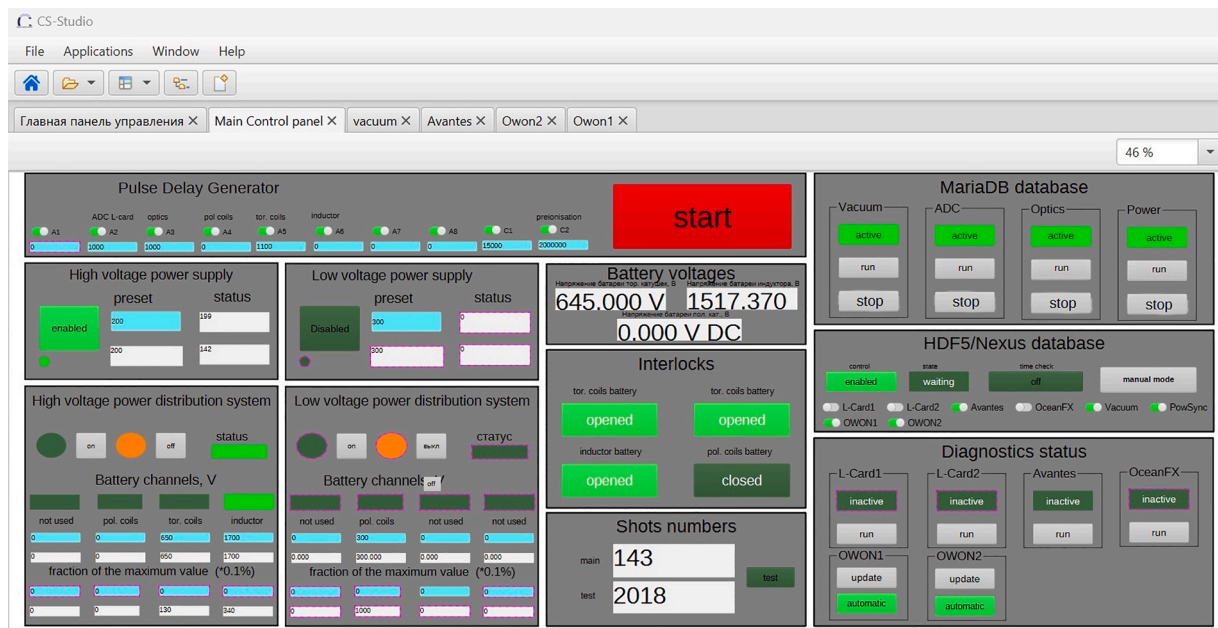


Fig. 9. Main control panel of the MEPHIST-0 tokamak in Control System Studio/Phoenix.

solving sophisticated tasks of automating large research and technological facilities. At the same time, the first experience of attracting physics students and engineering students to work with the created control and data acquisition system showed a high potential for using these tools for educational purposes. It seems that such an approach will, on the one hand, make it possible to train specialists in the field of industrial automation, and, on the other hand, will allow physicists to gain experience in operating a tokamak without the need to immerse themselves in the philosophy of the control system itself.

Further work with the system involves the integration of new devices, as well as the design of interfaces for the implementation of “easy” modes of operation with higher degree of automation, which would be useful for low-skilled users. Our future plans also include the preparation of a web platform, that will serve as an educational course for students from other universities, who will not be able to participate in the experiments in person. It would include our knowledge system, the simplified operator’s web displays and data analysis system. Finally, the possibility of integration the “MDSplus” system in addition to the existing RDB and HDF5 will be considered, so that students can compare different approaches to the data management, prevalent in fusion research.

CRedit authorship contribution statement

N. Efimov: Writing – original draft, Writing – review & editing, Investigation, Software, Visualization. **F. Podolyako:** Writing – review & editing, Investigation, Software, Methodology, Validation. **A. Prishvitsyn:** Investigation, Methodology, Formal analysis, Resources. **S. Krat:** Writing – review & editing, Project administration, Funding acquisition, Resources.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the complex program “Development of equipment, technologies and research in the field of atomic energy use in the Russian Federation for the period up to 2024”, supplementary agreements № 075-03-2021-093/3 from 14.04.2021, № 075-03-2022-102-5 from 01.06.2022, № 075-03-2023-097/4 from 17.04.2023, № 075-03-2023-097/7 from 20.06.2023.

References

- [1] S. Krat, et al., MEPHIST-0 tokamak for education and research, *Fus. Sci. Technol.* (2023) 1–19, <https://doi.org/10.1080/15361055.2022.2149033>.
- [2] S.A. Krat, et al., MEPHIST-0 tokamak toroidal magnetic field system, *Phys. Atom. Nucl.* 84 (12) (2021) 1995–2003, <https://doi.org/10.1134/S1063778821120024>.
- [3] O. Grover, et al., Remote operation of the GOLEM tokamak for fusion education, *Fus. Eng. Des.* 112 (2016) 1038–1044, <https://doi.org/10.1016/j.fusengdes.2016.05.009>.
- [4] A.V. Melnikov, et al., Physical program and diagnostics of the T-15 upgrade tokamak (brief overview), *Fus. Eng. Des.* 96–97 (2015) 306–310, <https://doi.org/10.1016/j.fusengdes.2015.06.080>.
- [5] V. Philipps, Ph. Mertens, G.F. Matthews, H. Maier, Overview of the JET ITER-like wall project, *Fus. Eng. Des.* 85 (7–9) (2010) 1581–1586, <https://doi.org/10.1016/J.FUSENGDES.2010.04.048>.
- [6] R. Aymar, P. Barabaschi, Y. Shimomura, The ITER design, *Plasma Phys. Control. Fus.* 44 (5) (2002) 519, <https://doi.org/10.1088/0741-3335/44/5/304>.
- [7] R. Bork, et al., advligots: the Advanced LIGO real-time digital control and data acquisition system, *SoftwareX* 13 (2021), 100619, <https://doi.org/10.1016/j.softx.2020.100619>.
- [8] M.R. Kraimer, M. Clausen, W. Lupton, C. Watson, Experience with EPICS in a wide variety of applications, in: *Proceedings of the 1997 Particle Accelerator Conference* (Cat. No.97CH36167), IEEE, Vancouver, BC, Canada, 1998, pp. 2403–2409, <https://doi.org/10.1109/PAC.1997.751222>.
- [9] D. Van Houtte, F. Sagot, K. Okayama, K. Blackler, A functional approach for managing ITER operations, *Fus. Eng. Des.* 87 (5–6) (2012) 652–656, <https://doi.org/10.1016/j.fusengdes.2012.01.033>.
- [10] L. Zhu, et al., The upgrade of the EAST poloidal field power supply monitoring system, *Fus. Eng. Des.* 172 (2021), 112757, <https://doi.org/10.1016/j.fusengdes.2021.112757>.
- [11] J. Waterhouse, A. Stephen, N. Petrella, Introduction of ITER CODAC relevant technologies on JET and MAST, *Fus. Eng. Des.* 161 (2020), 111858, <https://doi.org/10.1016/j.fusengdes.2020.111858>.
- [12] K.H. Kim, et al., The KSTAR integrated control system based on EPICS, *Fus. Eng. Des.* 81 (15–17) (2006) 1829–1833, <https://doi.org/10.1016/j.fusengdes.2006.04.026>.
- [13] A. Havránek, et al., Conceptual design of the COMPASS-U control systems, *Fus. Eng. Des.* 170 (2021), 112550, <https://doi.org/10.1016/j.fusengdes.2021.112550>.

- [14] M.A. Javed, M.A. Naveed, A. Hussain, S. Hussain, Integrated data acquisition, storage and retrieval for glass spherical tokamak (GLAST), *Fus. Eng. Des.* 152 (2020), 111424, <https://doi.org/10.1016/j.fusengdes.2019.111424>.
- [15] L. Dalesio, et al., EPICS 7 provides major enhancements to the EPICS toolkit, in: *Proceedings of the 16th Int. Conf. on Accelerator and Large Experimental Control Systems ICALEPCS2017*, 2018, p. 5, <https://doi.org/10.18429/JACOW-ICALEPCS2017-MOBPL01>, pages, 0.98 5MB.
- [16] "PyDevice module. <https://github.com/klemenv/PyDevice>".
- [17] "Python library for Avantes. <https://github.com/tspspi/pyavaspec>".
- [18] "Python module for Ocean Optics spectrometers. <https://github.com/ap-/python-eabreeze>".
- [19] "Python API for XIMEA cameras. <https://www.ximea.com/support/wiki/apis/python>".
- [20] "OpenCV. <https://docs.opencv.org/4.8.0/>".
- [21] "L-Card E-502 ADC manual. https://www.lcard.ru/download/e-502_users_guide_en.pdf".
- [22] "Open Media Vault NAS. <https://www.openmediavault.org/>".
- [23] J.-M. Kwon, et al., Development of a Virtual Tokamak platform, *Fus. Eng. Des.* 184 (2022), 113281, <https://doi.org/10.1016/j.fusengdes.2022.113281>.
- [24] R. Castro, et al., Data archiving system implementation in ITER's CODAC core system, *Fus. Eng. Des.* 96–97 (2015) 751–755, <https://doi.org/10.1016/j.fusengdes.2015.06.076>.
- [25] K. Shroff, K.-U. Kasemir, C. Rosati, G. Weiss, New Java frameworks for building next generation EPICS applications, in: *Proceedings of the 17th International Conference on Accelerator and Large Experimental Physics Control Systems ICALEPCS2019*, 2020, p. 4, <https://doi.org/10.18429/JACOW-ICALEPCS2019-WESH1002>, pages, 1.096 MB.
- [26] "MariaDB. <https://mariadb.org/documentation/>".
- [27] "PostgreSQL. <https://www.postgresql.org/docs/>".
- [28] "MDSplus. <https://www.mdsplus.org/>".
- [29] "MDSplus users. <https://www.mdsplus.org/index.php/Introduction:Sites>".
- [30] T.W. Fredian, J.A. Stillerman, MDSplus. Current developments and future directions, *Fus. Eng. Des.* 60 (3) (2002) 229–233, [https://doi.org/10.1016/S0920-3796\(02\)00013-3](https://doi.org/10.1016/S0920-3796(02)00013-3).
- [31] T. Fredian, J. Stillerman, G. Manduchi, A. Rigoni, K. Erickson, T. Schröder, MDSplus yesterday, today and tomorrow, *Fus. Eng. Des.* 127 (2018) 106–110, <https://doi.org/10.1016/j.fusengdes.2017.12.010>.
- [32] "HDF5. <https://www.hdfgroup.org/solutions/hdf5/>".
- [33] M. Könnicke, et al., The NeXus data format, *J. Appl. Crystallogr.* 48 (1) (2015) 301–305, <https://doi.org/10.1107/S1600576714027575>.
- [34] "h5py library. <https://docs.h5py.org/en/stable/>".
- [35] "H5Web. https://github.com/silx-kit/jupyterlab-h5web/tree/main/jupyterlab_h5web".
- [36] S.S. Portone, E.Yu. Mironova, O.I. Semenov, et al., Infrastructural hardware platform of the common IT space for fusion research (FusionSpace), 49th International Conference on Plasma Physics and CF, Zvenigorod, 2022. doi:10.34854/ICPAF.2022.49.1.174.