ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP110

CONTROL SYSTEM DESIGN OF THE CHIMERA FUSION TEST FACILITY

P. T. Smith, A. J. Greer, B. A. Roberts, P. B. Taylor, Observatory Sciences Ltd, St Ives, UK
M. Roberts, D. Mccubbin, Jacobs Clean Energy Ltd, Warrington, UK

Abstract

CHIMERA is an experimental nuclear fusion test facility which aims to simulate the intense magnetic fields and temperature gradients found within a tokamak fusion reactor. The control system at CHIMERA is based on EPICS and will have approximately 30 input/output controllers (IOCs) when it comes online in 2024. It will make heavy use of CSS Phoebus for its user interface, sequencer and alarm system. CHIMERA will use EPICS Archiver Appliance for data archiving and EPICS areaDetector to acquire high speed data which is stored in the HDF5 format. The control philosophy at CHIMERA emphasises PLC based control logic using mostly Siemens S7-1500 PLCs and using OPCUA to communicate with EPICS. EPICS AUTOSAVE is used both for manually setting lists of process variables (PVs) and for automatic restoration of PVs if an IOC must be restarted.

INTRODUCTION

CHIMERA is being developed for the United Kingdom Atomic Energy Authority (UKAEA) [1] with Jacobs Clean Energy Ltd as principle designer and constructor and Observatory Sciences Ltd developing the SCADA system. The CHIMERA nuclear fusion test facility (see Figures 1 and 2) aims to test various in-vessel components of a nuclear fusion reactor in conditions that replicate the harsh environment in which they will operate [1]. It will recreate the magnetic field gradient, magnetic field transients and heat fluxes that exist within a tokamak reactor. CHIMERA is designed to study a subject under test (SUT) up to a volume of 1.67 x 0.96 x 0.46 m^3. This is the size of the Test Blanket Module (TBM) being developed for the International Thermonuclear Experimental Reactor (ITER) and tested at CHIMERA. To properly test the TBM, a cooling system which runs at up to pressurised water reactor conditions (328 °C, 155 bars) is being built at CHIMERA and will be connected to the TBM during testing.

At the end of phase 1, CHIMERA will be capable of producing a peak static magnetic field of 5 tesla, a ± 0.25 tesla pulsed magnetic field, surface heating of 0.5 MW/m^2, and volumetric heating of 100 kW. CHIMERA phase 2 will introduce a high heat flux continuous-wave laser producing heat fluxes of 200 MW/m^2over 100 mm^2or 20 MW/m^2 over 1500 mm^2. It will also introduce a liquid metal loop which can be used to circulate liquid PbLi around a water-cooled lithium lead (WCLL) blanket [2].

A separate, but equally important aspect of project CHIMERA, is the facility's digital twin. This term describes a suite of software which will make use of CHIMERA test data to develop a sophisticated simulation of the experiments being carried out at CHIMERA. The intention is that this simulation will run in real-time and use live test data from the data acquisition (DAQ) system to predict the outcome of experiments and further improve the simulation. It will also be used to predict upcoming problems with live experiments before they occur [1].

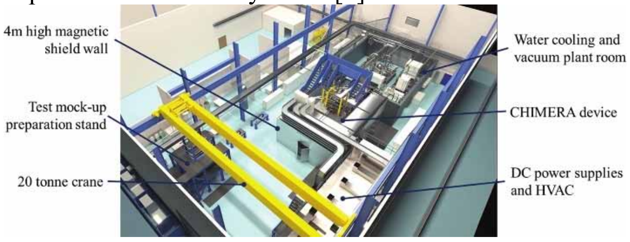

4m high magnetic shield wall
Test mock-up preparation stand
20 tonne crane
Water cooling and vacuum plant room
CHIMERA device
DC power supplies and HVAC

Figure 1: Top down view of the CHIMERA facility. (Courtesy of UKAEA).

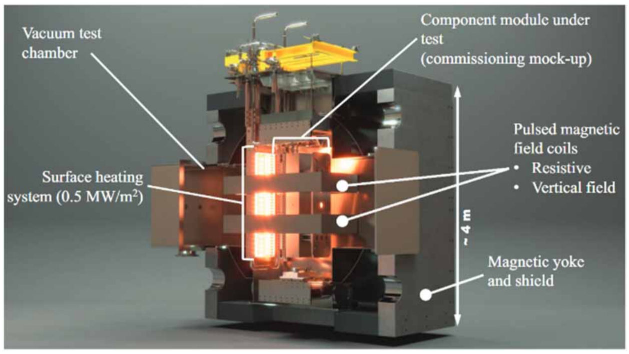

Vacuum test chamber
Component module under test (commissioning mock-up)
Pulsed magnetic field coils
• Resistive
• Vertical field
Surface heating system (0.5 MW/m²)
~ 4 m
Magnetic yoke and shield

Figure 2: Cross-sectional view into the CHIMERA vacuum test chamber. The superconducting magnets are not shown (Courtesy of UKAEA).

CONTROL SYSTEM OVERVIEW

The SCADA system is being developed using the Experimental Physics and Industrial Control System (EPICS) framework. EPICS was chosen in part, due to its open source and flexible nature which will allow this experimental facility to develop over time. The key software can be divided into “high-level” components and “low-level” components. The high-level components are the DAQ system, data archiver, server monitoring system, experiment sequencer, alarm handling system and web server for external access of data. The “low-level” components are 7 IOCs which directly control and monitor various hardware components. All of these SCADA systems communicate over a controls network via TCP/IP, most communications are done with the EPICS channel access (CA) protocol. This paper will focus on giving an overview of the design and current development of the SCADA system prior to the commissioning of the CHIMERA facility.

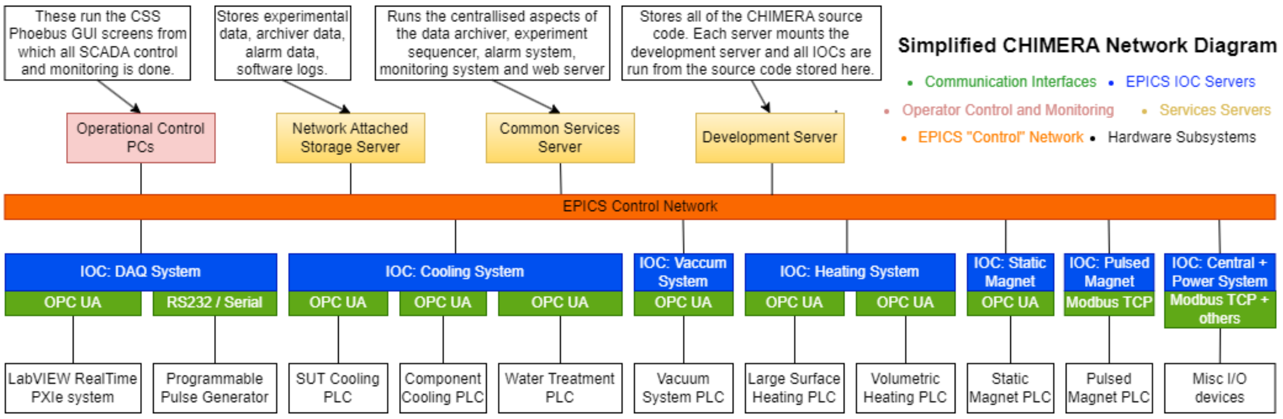

These run the CSS Phoebus GUI screens from which all SCADA control and monitoring is done.
Stores experimental data, archiver data, alarm data, software logs.
Runs the centrallised aspects of the data archiver, experiment sequencer, alarm system, monitoring system and web server
Stores all of the CHIMERA source code. Each server mounts the development server and all IOCs are run from the source code stored here.
Simplified CHIMERA Network Diagram
• Communication Interfaces
• EPICS IOC Servers
• Operator Control and Monitoring
• Services Servers
• EPICS "Control" Network
• Hardware Subsystems
Operational Control PCs
Network Attached Storage Server
Common Services Server
Development Server
EPICS Control Network
IOC: DAQ System
OPC UA
RS232 / Serial
IOC: Cooling System
OPC UA
OPC UA
OPC UA
IOC: Vaccum System
OPC UA
IOC: Heating System
OPC UA
OPC UA
IOC: Static Magnet
OPC UA
IOC: Pulsed Magnet
Modbus TCP
IOC: Central + Power System
Modbus TCP + others
LabVIEW RealTime PXIe system
Programmable Pulse Generator
SUT Cooling PLC
Component Cooling PLC
Water Treatment PLC
Vacuum System PLC
Large Surface Heating PLC
Volumetric Heating PLC
Static Magnet PLC
Pulsed Magnet PLC
Misc I/O devices

Figure 3: A simplified network diagram outlining the different servers which makeup phase 1 of the CHIMERA SCADA system and their place within the network (adapted from a diagram courtesy of Jacobs Clean Energy).

CONTROL PHILOSOPHY

Low level control of hardware is mostly done by programmable logic controllers (PLCs), with high level commands being sent to the PLCs from EPICS. The PLCs will react to these commands and follow their programmed logic to either execute the command or reject it. As the hardware manufacturer is usually also responsible for creating the PLC which controls the hardware, this approach makes development of the PLC logic smoother. This is important as at CHIMERA all operations that are deemed safety critical or that could damage equipment are done within the PLCs and an interconnected series of interlocks between them.

Commands are sent to PLCs using mostly either the OPCUA standard or the Modbus standard. All communications between SCADA and hardware equipment are done over a TCP/IP network. Where a hardware device requires a serial connection, a serial to TCP/IP converter device is used. Each hardware subsystem (vacuum, cooling, etc) can only communicate with its corresponding IOC which will be running on a dedicated industrial computer. The controls network is insulated from external access from the internet as an additional security measure.

OFFLINE INSTALLATION

One requirement for this facility is that the SCADA system must be fully installable and configurable while disconnected from the internet. All SCADA servers run Redhat 8 and so we have created a custom Redhat ISO file which makes use of the Redhat kickstart feature. This allows each server's installation settings to be preconfigured so that the installation of the OS can be fully automated.

All custom CHIMERA source code is installed and built-into a directory on the Development server (see Figure 3), which is then network mounted to every other server. All custom software and IOCs are run from this network mounted directory which keeps code centralised and easy to maintain. If an IOC server were to lose connection to the central build directory, the IOC would continue functioning normally. However, if the IOC needed to be restarted then this would require reconnection of the development server.

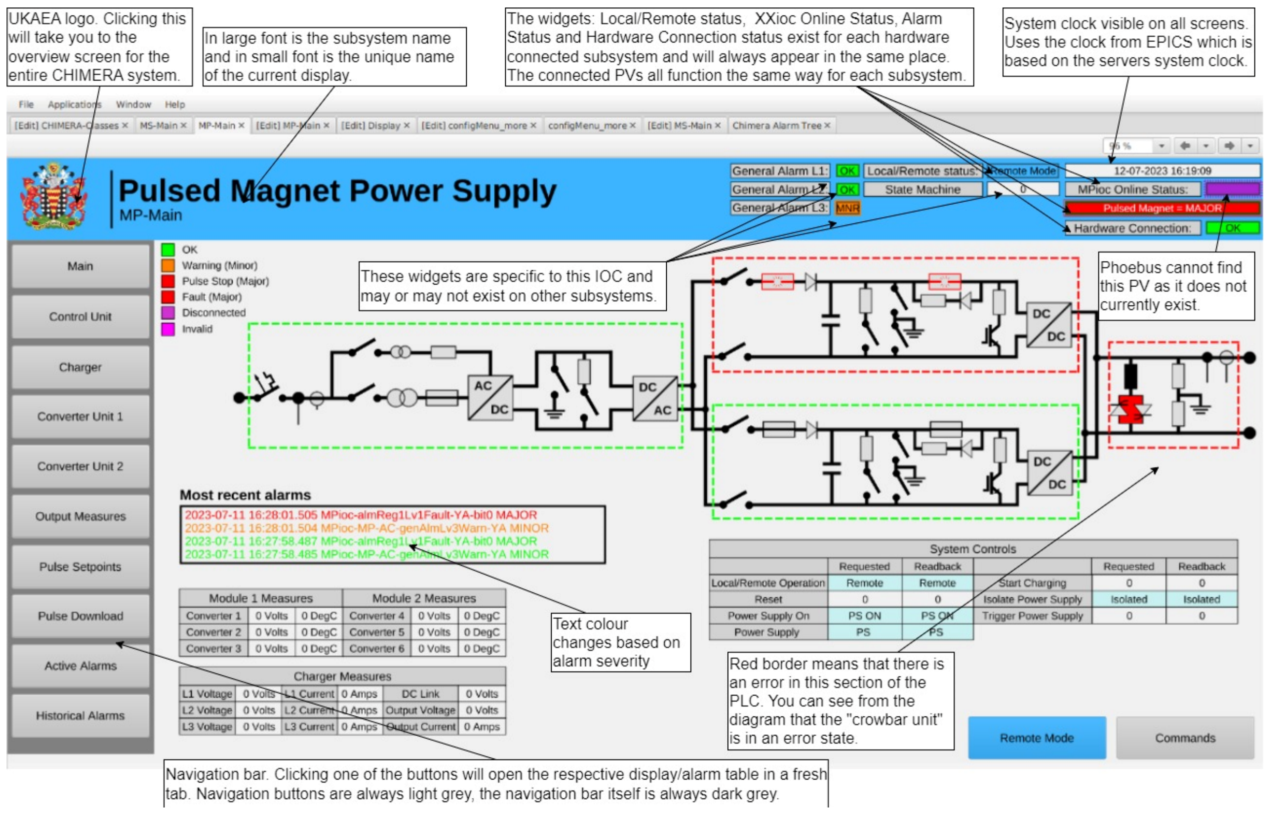
PLC VARIABLE LISTS

PLC variables are just a name given to data structures which are sent or received from PLCs. These are defined in a spreadsheet which is intended to be used as a source of information by both PLC and SCADA engineers. This spreadsheet contains all of the information required to create an EPICS record to communicate with the corresponding PLC variable. This often includes a definition of the OPCUA namespace and nodeID or the Modbus register number which is used to access the variable. It also contains the variables datatype, min/max range and the information required to add the record generated from the variable to the archiving and alarm systems if required.

The signal lists are standardised to allow the creation of a system which is used to automatically propagate updates made to PLC software into the SCADA software. If a PLC engineer updates the signal list to add a new signal, a python script can be run which recreates relevant EPICS database files as well as updating archiver and alarm system data files. The python module epicsdbbuilder [3] is used for the automatic creation of EPICS database files, while the alarm server data file must be made as an xml file to the requirements of the Phoebus alarm-server. Finally, the archiver data file must follow the specifications of the SLAC archiver appliance helper scripts which are used to feed new process variables (PV)s into the archiver.

USER INTERFACE

Almost every subsystem at CHIMERA is controlled and monitored from the Phoebus application [4]. Most SCADA functionality including the archiver, alarm system, experimental sequencer and server monitoring are also operated using Phoebus. Because all of this functionality is available within Phoebus, either by creating custom displays or connecting to standard Phoebus applications, an operator can more easily navigate to different control and monitoring systems rather than having to juggle different applications.



UKAEA logo. Clicking this will take you to the overview screen for the entire CHIMERA system.

In large font is the subsystem name and in small font is the unique name of the current display.

The widgets: Local/Remote status, XXioc Online Status, Alarm Status and Hardware Connection status exist for each hardware connected subsystem and will always appear in the same place. The connected PVs all function the same way for each subsystem.

System clock visible on all screens. Uses the clock from EPICS which is based on the servers system clock.

These widgets are specific to this IOC and may or may not exist on other subsystems.

Phoebus cannot find this PV as it does not currently exist.

Text colour changes based on alarm severity

Red border means that there is an error in this section of the PLC. You can see from the diagram that the "crowbar unit" is in an error state.

Navigation bar. Clicking one of the buttons will open the respective display/alarm table in a fresh tab. Navigation buttons are always light grey, the navigation bar itself is always dark grey.

General Alarm L1:	OK
General Alarm L2:	OK
General Alarm L3:	MNR
Local/Remote status:	Remote Mode
State Machine	0
	12-07-2023 16:19:09
MPioc Online Status:	
	Pulsed Magnet = MAJOR
Hardware Connection:	OK

Most recent alarms

```
2023-07-11 16:28:01.505 MPioc-almReg1Lv1Fault-YA-bit0 MAJOR
2023-07-11 16:28:01.504 MPioc-MP-AC-genAlmLv3Warn-YA MINOR
2023-07-11 16:27:58.487 MPioc-almReg1Lv1Fault-YA-bit0 MAJOR
2023-07-11 16:27:58.485 MPioc-MP-AC-genAlmLv3Warn-YA MINOR
```

Module 1 Measures			Module 2 Measures		
Converter 1	0 Volts	0 DegC	Converter 4	0 Volts	0 DegC
Converter 2	0 Volts	0 DegC	Converter 5	0 Volts	0 DegC
Converter 3	0 Volts	0 DegC	Converter 6	0 Volts	0 DegC

Charger Measures					
L1 Voltage	0 Volts	L1 Current	0 Amps	DC Link	0 Volts
L2 Voltage	0 Volts	L2 Current	0 Amps	Output Voltage	0 Volts
L3 Voltage	0 Volts	L3 Current	0 Amps	Output Current	0 Amps

System Controls					
	Requested	Readback		Requested	Readback
Local/Remote Operation	Remote	Remote	Start Charging	0	0
Reset	0	0	Isolate Power Supply	Isolated	Isolated
Power Supply On	PS ON	PS ON	Trigger Power Supply	0	0
Power Supply	PS	PS			

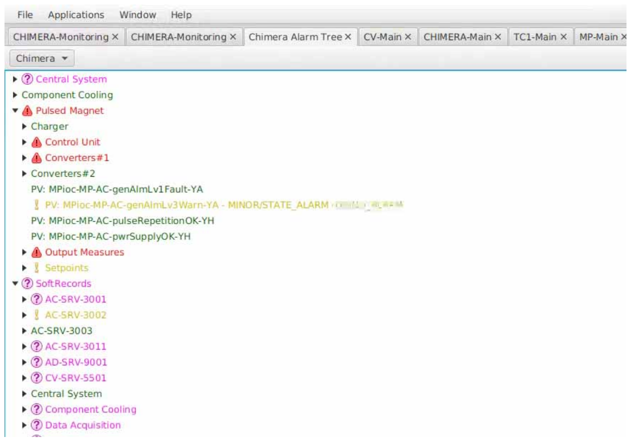
Figure 4: The in-development "Main" screen for the pulsed magnet subsystem, displayed in Phoebus.

To make full use of Phoebus, at CHIMERA we use both Phoebus services and elements of the Phoebus GUI which interface with the services. For example, we use both the Phoebus alarm-server service and the Phoebus alarm table and alarm tree which are graphical apps that form part of the Phoebus GUI.

Each subsystem has a set of displays which are used for control and monitoring, each subsystem has a “Main” screen which includes the most important information for the system and an animated process and instrumentation diagram (P&ID), as shown in Figure 4.

ALARM HANDLING

The Phoebus alarm server and the Phoebus alarm logger are used to detect alarms raised in EPICS PVs, collect them and then store them for retrieval at a later date. The alarm server will detect a new alarm by monitoring a list of pre-defined PVs over CA, when an alarm is detected the alarm server will create a new alarm instance. This is then sent to Kafka, a communication broker, which stores the state of the active alarm and communicates it to the Phoebus GUI [5]. Alarms are organised in a tree like structure; at CHIMERA each branch of the tree is a hardware subsystem with an additional branch for all software-only alarms. Each branch can be further divided into smaller groups of alarms. The alarms and their groups can be viewed in the Phoebus alarm table or alarm tree, see Figure 5.



```
▸ Central System
▸ Component Cooling
▾ Pulsed Magnet
  ▸ Charger
  ▸ Control Unit
  ▸ Converters#1
  ▸ Converters#2
    PV: MPioc-MP-AC-genAlmLv1Fault-YA
    PV: MPioc-MP-AC-genAlmLv3Warn-YA - MINOR/STATE_ALARM
    PV: MPioc-MP-AC-pulseRepetitionOK-YH
    PV: MPioc-MP-AC-pwrSupplyOK-YH
  ▸ Output Measures
  ▸ Setpoints
▾ SoftRecords
  ▸ AC-SRV-3001
  ▸ AC-SRV-3002
  ▸ AC-SRV-3003
  ▸ AC-SRV-3011
  ▸ AD-SRV-9001
  ▸ CV-SRV-5501
  ▸ Central System
  ▸ Component Cooling
  ▸ Data Acquisition
```

Figure 5: The Phoebus alarm tree showing disconnected, Major, Minor and OK alarms within various subsystems.

The alarm logger checks Kafka for new alarms which are then put into long term storage on the NAS (Network Attached Storage) server using Elasticsearch. Elasticsearch can be queried to search for specific historical alarms, this is done by the Phoebus alarm log table and can also be done by making a http request in a web browser. A python Elasticsearch module is used to regularly query Elasticsearch to get the four most recent alarms which are then displayed to the operator.

IOC DESIGN & SERVER MONITORING

There are currently 34 IOCs under development, this includes IOCs to monitor server stats, IOCs to monitor the status of other IOCs, IOCs to communicate with PLCs and some additional IOCs required for various SCADA systems to function properly. These IOCs can be seen in Figure 6 along with the server on which they will run.

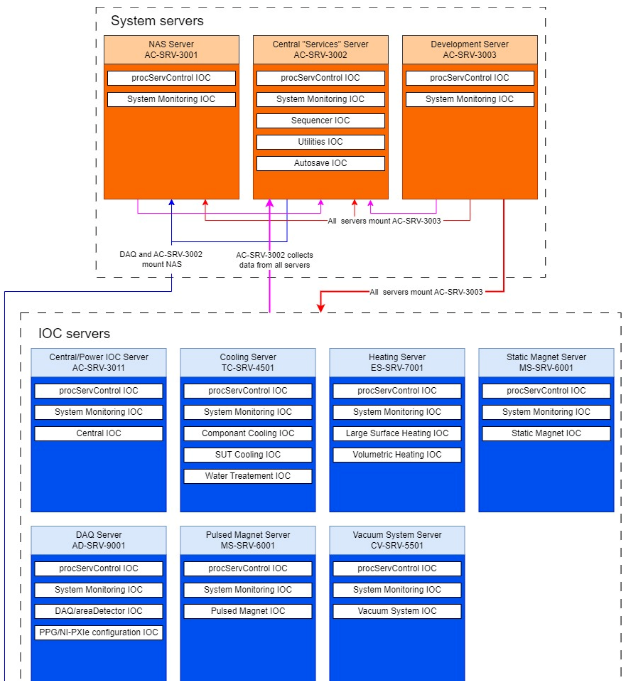


Figure 6: A diagram showing all of the planned servers and the IOCs which will run on them at CHIMERA.

Each server hosts a python "System Monitoring" IOC which will monitor information such as CPU temperatures and storage capacity, it will also raise alarms when certain conditions are met. The python softioc [6] library is used to make these IOCs which will communicate over CA with the rest of the EPICS network. These python IOCs are easy to develop which makes it easier to implement custom code to communicate with a variety of hardware sensors.

At CHIMERA, each IOC is run as a Linux systemd service within a procServ shell. To monitor and control these procServ instances, each server has a procServControl IOC which checks the state of each other IOC running on the server using the EPICS procServControl [7] support module. This support module opens an asyn TCP connection to the procServ shell which is used to report the status of the IOC and to get the output of the IOC shell which is then displayed in Phoebus.

ARCHIVER

EPICS Archive Appliance [8] is used to archive PVs and store changes to the PV for long periods of time. Currently CHIMERA archives over 2000 PVs, this number is likely to grow by a factor of 2 or 3 before completion of CHIMERA phase 1. Archived data is initially stored on an SSD within the "Services" server which runs the archiver. Every day this is backed up to the NAS server where it is stored on a 60TB RAID 6 hard drive array. Some archived PVs are configured to be archived every few seconds, while less active PV's are only archived when their value changes. This behaviour is all configured through the signal lists as previously discussed.

EXPERIMENT SEQUENCER

Experimental "campaigns" at CHIMERA are expected to take days or weeks to complete and require the execution of hundreds or thousands of commands. These commands will be sent automatically by following a predefined sequence of steps which are executed when certain parameters are true. A typical step may involve checking that a permissive PV is true and then writing a value to a hardware connected PV, only if the system is in a valid state. This data will then be sent to the connected PLC. The sequencer also requires the ability to be paused by an operator and then later resumed as well as some error recovery. To define an experimental sequence, a user can submit a suitably formatted and approved spreadsheet. This will be passed into the Phoebus Scan Server which manages the different steps of the experimental sequence.

There are two levels of state machine within the SCADA system, a subsystem state and an overall system state. Some of the possible subsystem states are: Disabled, Ready, Busy, Fault, Local and Shutdown. The overall system states are shown in Figure 7.

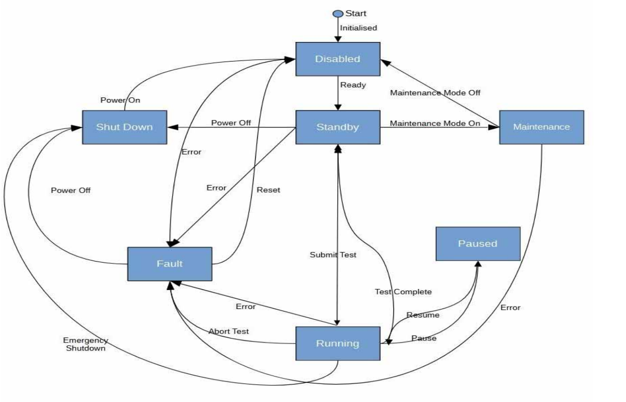


Figure 7: Overall system state diagram

The experiment sequencing system is still early in development, but the current intention is to use an EPICS python IOC to implement the state machines. This will be used in conjunction with the Phoebus scan server which is used to define the steps of the sequence. The progress of the sequencer can be viewed in the Phoebus scan monitor, while each IOCs state machine can be viewed on the GUIs for the relevant IOC as well as on a top-level sequencer GUI. The nature of experiments at CHIMERA means that individual experiments are usually long and costly, meaning that a reliable sequencer is essential. The clearly defined steps and relative simplicity of the Scan Server supports this, while thorough testing of the state machines will ensure its reliability.

DATA ACQUISITION AND RETRIEVAL

Experimental data is acquired and initially processed by a National Instruments PXIe Labview system which uploads it to an OPC UA server. There are approximately 400 data channels uploaded to OPC UA, each representing a hardware sensor. Data will be collected by the PXIe system at 2500 Hz but written into ~ 400 arrays of 250, 64-bit floating-point values at a rate of 10 Hz. Experimental data will be taken in up to 12 second bursts, meaning each OPCUA array is written to 120 times per burst. Records which are setup to monitor these OPC UA arrays will collect the data as it is written and pass it into a custom areaDetector [9] driver. This driver collects all of the data over the 12 seconds and writes it to an areaDetector NDArray as the data arrives. Once the 12 seconds is over, the NDArray is stored as an HDF5 file which is later copied to the NAS server for storage. Several years' worth of experimental data and archiver data will be accessible within the local network through http requests. Some external access to this data is also available to authorised users.

CONCLUSION

As development of the novel CHIMERA facility continues, it is certain that the software requirements will grow and change. Such an experimental facility requires a very flexible SCADA system and approach to software development. EPICS lends itself to this approach and development of systems such as the automated database creation will allow integration of future changes in the hardware into the SCADA system. Software like areaDetector and archive appliance have the capacity for much larger data rates than are currently expected at CHIMERA which leaves the system room to grow. Future work will involve continuing development of the "high level" components in the system and ensuring that they can facilitate changes to the still under development hardware and associated "low level" IOCs. By taking a cautious and flexible approach to development, progress should continue to be made with minimal software changes required once hardware systems are more mature.

An initial deployment of SCADA software is expected in 2024 this will integrate SCADA with the core hardware subsystems of CHIMERA. Once the core subsystems are connected to SCADA, high level components can be better tested and having these in place will aid in the deployment of additional hardware.

REFERENCES

[1] T. R. BARRETT *et al.*, "CHIMERA Fusion Technology Facility: Testing and Virtual Qualification", *Fusion Sci. Technol.*, vol. 79, no. 8, p 1-12, Feb. 2023. doi:10.1080/15361055.2022.2147766

[2] J. AUBERT *et al.,* "Design and Preliminary Analyses of the New Water Cooled Lithium Lead TBM for ITER," *Fusion Eng. Des.*, vol. 160, pp. 111921, Nov. 2020. doi: 10.1016/j.fusengdes.2020.111921

[3] Epicsdbbuilder python module, https://github.com/dls-controls/epicsdbbuilder

[4] CS-Studio Phoebus, https://controlssoftware.sns.ornl.gov/css_phoebus/

[5] CS-Studio Phoebus alarm system documentation, https://control-system-studio.readthedocs.io/en/latest/app/alarm/ui/doc/index.html

[6] Python softioc module, https://github.com/dls-controls/pythonSoftIOC

[7] EPICS procServContol support module, https://github.com/dls-controls/procServControl

[8] The EPICS Archive Appliance, https://slacmshankar.github.io/epicsarchiver_docs/index.html

[9] EPICS areaDetector, https://areadetector.github.io/master/index.html