CTAO System Control Standards

Doc. No: CTA-STD-SEI-000000-0004-1a
2023-02-08

	First/Last Name, Organisation, Role	Digital signature
Prepared by (1)	Vanessa Montes CTAO, Systems Engineer	Vanessa Montes Digitally signed by Vanessa Montes Date: 2023.03.30 15:03:21 +02'00'
Approved by (1)	Nick Whyborn CTAO, Lead Systems Engineer	Nick Whyborn Digitally signed by Nick Whyborn DN: cn=Nick Whyborn, o=CTAO, ou, email=nick.whyborn@cta-observatory.org, c=IT Date: 2023.03.31 11:22:57 +01'00'
Released by	Wolfgang Wild CTAO, Project Manager	Wolfgang Wild Digitally signed by Wolfgang Wild Date: 2023.03.31 11:31:40 +02'00'

Revision History				
Issue	**Rev.**	**Created**	**Reasons / Remarks / Section**	**Author**
1	a	27.10.2020	Initial draft (Draft 01)	V. Montes
1	a	08.04.2022	Included comments from internal review (Draft 02)	V. Montes
1	a	07.07.2022	Included comments from external review (Draft 03)	V. Montes
1	a	08.02.2023	Included comments from SE team and reference to section 6.4 (Draft 04)	V. Montes

Authors	
First/Last Name, Organisation	**Contribution Subject/Chapter**
V. Montes	Author
E. Antolini	Contributor

Table of Contents

1 Introduction

The System Control of CTAO (CS) comprises the hardware and software elements which, in a safe and efficient way, control and monitor both arrays and its different subsystems [AD1]. Considering the challenges associated to the complexity of the observatory's overall design, and to ensure that the desired performance is met, it is fundamental to describe the system's high-level structure and its internal and external interactions.

1.1 Purpose

The purpose of this document is to, along with the documents that describe CTAO's concept for the system control [AD1] and development guidelines [AD2], prescribe the conditions under which the corresponding hardware and software will be developed for the operation of the observatory.

1.2 Scope

This document specifies the standards to be considered, along with the corresponding Requirements Specification documents, throughout the design process of the different subsystems in order to guarantee the cost efficiency, integration and maintainability of the Control System of CTAO.

The applicability of the protocols and standards defined in this document, with respect to the Systems Control Hierarchy as defined in [AD1], is detailed in each section.

2 Applicable and Reference Documents

2.1 Applicable Documents

An 'Applicable Document' is one that is referenced by a 'shall statement' in any section of this document. The following list of documents, of the revision indicated, form a part of the present document to the extent specified herein.

AD1	CTAO System Control Concept, CTA-TRE-SEI-000000-0016, Issue 1, Rev.: a, 2022
AD2	CTAO System Control Development Guidelines, CTA-TRE-SEI-000000-0017, Issue 1, Rev.: a, 2022
AD3	Guide to Industrial Control Systems (ICS) Security SP 800-82 Rev2
AD4	Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations IEC 61508-4-2020
AD5	The Machinery Directive, Directive 2006/42/EC
AD6	Alma Common Software. G. Chiozzi, et al, "CORBA-based Common Software for the ALMA project", in Proc SPIE 4848, 43, 2002. Doi: 10.1117/12.461036
AD7	OPC Unified Architecture Specification. IEC 62541
AD8	Industrial communication networks – Profiles – Part 3-3: Functional safety fieldbuses – Additional specifications for CPF 3. IEC 61784-3-3:2021
AD9	Telecommunications and information exchange between systems – Local and metropolitan area networks – Part 3. IEEE Std 802.3-2018
AD10	Industrial Communication Networks – Fieldbus specifications. IEC 61158
AD11	Industrial Communication Networks – Fieldbus Specifications – Part 6-10: Application layer protocol specification – Type 10 elements. IEC 61158-6-10:2019
AD12	CTAO Timing Standards, CTA-STD-COM-000000-0001, Issue 1, Rev.: a, 2022.
AD13	Network Time Protocol. RFC 5905
AD14	Standard for Ethernet – Layer Management – Section 5, Clause 59. IEEE Std 802.3-2018
AD15	VHDL language reference manual, IEEE Std 1076-2019
AD16	Programmable controllers – Programming languages. IEC 61131-3:2013

AD17 Software Programming Standards. CTA-STD-OSO-000000-0001-1a

AD18 Systems and Software Engineering – Life cycle management – Part 4: Systems engineering planning ISO/IEC/IEEE 24748-4:2016

AD19 Systems and Software Engineering – System life cycle processes ISO/IEC/IEEE 15288:2015

AD20 CTA Software Licensing Policy CTA-STD-OSO-000000-0002-1h

AD21 Quantities and Units. ISO 80000

AD22 Data Elements and interchange formats – Information Interchange – Representation of dates and times. ISO 8601-1:2019

AD23 Documentation Control Plan CTA-PLA-MGT-000000-0009-1b (Draft)

AD24 CTAO RAM Calculation Methodology Guideline CTA-INS-SEI-000000-0001-1b

AD25 Systems and software engineering - Software life cycle processes ISO/IEC/IEEE 12207:2017

AD26 Safety of machinery - Safety-related parts of control systems Part 1: General principles for design ISO 13489-1:2015

AD27 Safety of machinery - Safety-related parts of control systems – Part 2: Validation ISO 13489-2:2012

2.2 Reference Documents

A “Reference Document” is aimed for general guidance and does not need to be applied.

RD-1 Array Control and Data Acquisition Quality Assurance Plan, CTA-PLA-ACA-303000-0002, Issue 2, Rev.: b, 2021.

RD-2 CTA ACADA Software Development Life Cycle, CTA-STD-ACA-303000-0001, Issue 2, Rev.: d, 2017.

3 Definitions and Conventions

3.1 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document:

ACADA	Array Control and Data Acquisition
ACS	Alma Common Software
AD	Applicable Document
CS	Control System
CORBA	Common Object Request Broker Architecture
COTS	Commercial off-the-shelf
CSS	Cascading Style Sheets
CTA	Cherenkov Telescope Array
CTAO	Cherenkov Telescope Array Observatory
EtherCAT	Ethernet for Control Automation Technology
FDB	Functional Block Diagram
FSoE	FailSafe over EtherCAT
HMI	Human Machine Interface
HTML	HyperText Markup Language
ICD	Interface Control Document
ICS	Industrial Control System
IDL	Interactive Data Language
IP	Internet Protocol
IPS	Integrated Protection System
LCS	Local Control System
LCU	Local Control Unit
MTU	Maximum Transmission Unit
NI	National Instruments
NTP	Network Time Protocol
OPC-UA	Open Platform Communications Unified Architecture
PLC	Programmable Logic Controller
PTP	Precision Time Protocol
SIL	Safety Integrity Level
ST	Structured Text
TAI	Temps Atomique International / International Atomic Time
UDP/IP	User Datagram Protocol
UTC	Universal Time Coordinated
VHDL	Very High Speed Integrated Circuit Hardware Description Language (VHSIC-HDL)

3.2 Definitions

The following definitions, as described in [AD1], are referred to throughout the document:

Control Unit: computing unit(s) in charge of executing the control and monitoring functions related to the associated item (e.g., drive systems, conditioning system, etc). (Adapted from [AD3])

Controller: a device or program that operates automatically to regulate a controlled variable [AD3].

Field/Plant: The physical elements necessary to support the physical process. This can include many of the static components not controlled by the ICS; however, the operations of the ICS may impact the adequacy, strength, and durability of the plant's components [AD3].

Fieldbus: A digital, serial, multi-drop, two-way data bus or communication path or link between low-level industrial field equipment such as sensors, transducers, actuators, local controllers, and even control room devices. Use of fieldbus technologies eliminates the need of point-to-point wiring between the controller and each device. A protocol is used to define messages over the fieldbus network with each message identifying a particular sensor on the network [AD3].

Local Control System: Control and safety units, control software, local communication infrastructure required to guarantee functional operation of a given equipment and all the other support elements needed for integration, verification, and maintenance activities. [Customized]

Safety Units: computing unit in charge of executing the safety related functions of the associated item. [Customized]

Safety Integrity Level: discrete level (one out of a possible four), corresponding to a range of safety integrity values, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest [AD4].

4 General

All controllable items shall implement a finite state machine approach at all levels of control layers [AD5].

All the Array Elements shall implement the Element Manager software component as defined in Sec 3.1 of [AD1].

5 Communication Infrastructure Standards

The technologies that shall be used for communication between Level 1 System CTA-X Controllers (see Figure 3 in [AD1]) and other components, are:

- ACS [AD6]
- OPC-UA [AD7]
- PROFIsafe [AD8]

For communication between, and/or within the array elements, the following technologies shall be used:

- Ethernet [AD9]
- EtherCAT [AD10]
- FSoE [AD8]
- PROFIBUS [AD9]
- PROFINET IO [AD11]
- PROFIsafe [AD8]
- Unicast and Multicast UDP/IP [AD9]
- OPC-UA [AD7]
- Modbus [AD10]

Exceptions to the protocols and standards presented in the previous list shall be presented and analyzed on a case-by-case basis, their implementation shall be subjected to CTAO's approval.

6 Interface Standards

The communication protocols can be categorized according to the following channels:

- Data Network
- Internal Control Network
- Time Distribution Network
- Safety Network

This chapter establishes the interfaces to the communication channels listed before, and therefore, the standards that shall be common to all the subsystems. In particular, this section focuses on the description of physical interfaces and their corresponding protocols, additional specifications that refer to determined conditions of each of the subsystems are described in the corresponding Interface Control Document (ICD).

6.1 General Considerations

For all products that are to be part of the CTAO network, parameters associated with the network configuration (for example, broadcast and multicast routing, IP addresses, port numbers, and MTUs) must be defined and managed by CTA personnel.

The use of jumbo packages shall be avoided, unless there are circumstances that justify making the exception. In this case, not only it shall be approved in advance by the CTAO, but it shall also be included in the document that defines the interface for said subsystem.

6.2 Data Network

The data network provides the infrastructure to transmit control, monitoring and scientific data between ACADA (CORBA based software which uses ACS as system framework) and, Array Element Managers and Supervisors. The data interface in each case shall be documented in the corresponding ICD.

6.2.1 Physical Layer

The following standards shall be used 100BASE, 1000BASE or 10GBASE (see [AD9]).

6.2.2 Data Link Layer

The standard to be used shall be Ethernet (see [AD9]).

6.2.3 Application Layer

To communicate to control units TCP/IP and UDP/IP shall be used [AD9].

6.3 Internal Control Network

The control network provides the infrastructure to monitor, sense and control data between supervisors and field units. The data interface of each LCS component to the control network shall be documented in the corresponding ICD.

6.3.1 Physical Layer

The following standards shall be used 100BASE, 1000BASE or 10GBASE (see [AD9]).

6.3.2 Data Link Layer

The standard to be used shall be Ethernet (see [AD9]).

6.3.3 Application Layer

To communicate to control units TCP/IP and UDP/IP shall be used [AD9]. OPC-UA shall be used to communicate to PLC units and potentially to other control units [AD7].

6.4 Time Distribution Network

The Time Distribution Network transports the data stream for the time synchronization of the Array Elements with respect to the central clock, depending on the required level of accuracy (according to [AD12]), it is divided in the two following subnetworks.

6.4.1 Standard Time Distribution Network

6.4.1.1 Physical Layer

The physical layer used for the Standard Distribution Network shall be the same as the one used for the Data Network.

6.4.1.2 Data Link Layer

The standard to be used shall be Ethernet [AD9].

6.4.1.3 Application Layer

NTP shall be used for clock synchronization in distributed networks [AD13].

6.4.2 Precision Time Distribution Network

6.4.2.1 Physical Layer

The following standard shall be used 1000BASE-BX10 [AD14]

6.4.2.2 Data Link Layer

The standard to be used shall be Ethernet [AD9].

6.4.2.3 Application Layer

White Rabbit PTP shall be used for clock synchronization in distributed network for systems which require nanosecond accuracy.

6.5 Safety Network

The safety network is meant to provide the appropriate infrastructure for the integration of control and safety units to the overall interlock and safety system.

6.5.1 Physical Layer

The following standards shall be used 100BASE-TX [AD9]

6.5.2 Data Link Layer

The standard to be used shall be Ethernet [AD9]

6.5.3 Application Layer

This network shall use a combination of PROFISafe [AD8] for interlock management and OPC-UA [AD9] for safety alarm transmission related data.

Within Local Control Systems EtherCAT and FSoE can also be implemented, as long as the appropriate interface between the LC and the Safety Network is defined and documented.

7 Programming Languages

7.1 Local Control Units

The delivered code shall meet the specifications defined in one of the following standards:

- VHDL [AD15]
- FBD and ST [AD16]
- Instruction List (IL) [AD16]
- Ladder Diagram (LD) [AD16]
- Sequential Functional Chart (SFC) [AD16]
- C/C++ [AD17]
- Java Standard Edition [AD17]
- Python [AD17]
- JavaScript [AD17]

Exceptions to the protocols and standards presented in the previous list shall be presented and analyzed on a case-by-case basis, their implementation shall be subjected to CTAO's approval.

The use of LabVIEW is to be considered for cases where LCUs cannot be programmed otherwise (e.g., NI LCUs), and therefore is to be avoided when it comes to standalone analysis and/or control applications which can be run in Linux and/or MS environments. Similarly, the use of Verilog is to be considered for exceptional cases, such as, ASICS and FPGAs programming.

7.2 Local Safety Units

Languages contained in [AD16], which are compatible with failsafe I/O, shall be used for Local Safety Units.

7.3 Human-Machine Interfaces

The following languages/platforms shall be considered for the development of HMIs for both control and safety units:

- TCP/IP [AD9]
- Java Script [AD17]
- Python [AD17]
- CSS
- HTML

The interface and programming standards described in previous sections shall be considered for the communication between the HMIs and the control and safety units.

8 Development Standards

For system development, the system's life cycle process considerations shall be compliant with [AD18] and [AD19].

In addition to this, design decisions shall be compliant and compatible with the Software Programming Standards document [AD17] and Software Licensing Policy [AD20].

9 Notational Standards

Quantities, systems of quantities, units, quantity and unit symbols, and coherent unit systems shall follow the International System of Quantities and the International System of Units according to [AD21].

The presentation format to be used for time shall be Coordinated Universal Time (UTC) according to [AD22], with the exception of Cherenkov telescopes data, in this case time stamping shall be expressed in International Atomic Time (TAI) timescale with epoch set to 1970-01-01 00:00:00 TAI.

10 Tools

10.1 Control Engineering

The tools used for analysis and simulation purposes for the control systems shall be common adopted ones (see examples in section 2.7 of [AD2]), well recognized and reviewed in advance in close collaboration with CTAO. The results of the performed analysis, and the models used to verify the performance for the control system against stablished requirements, shall be considered as a deliverable.

10.2 Version Control

During the lifecycle of a control system the use of a version control tool shall be implemented. For products' delivery, the use of CTAO's official GitLab (https://gitlab.cta-observatory.org) is expected to be considered, by the latest, for the Acceptance of said product. In the case of modifications or upgrades of the CTAO control system, the responsible person for these changes shall deliver version control repository.

11 Implementation

(Information related to runtime platforms and development environments shall be incorporated in the document's next version).

12 Documentation

Delivered documentation shall be compliant with the Documentation Control Plan [AD23].

For each control item (including third party products and/or COTS), the original documentation shall be delivered as part of the final documentation package.

13 RAM

13.1 Reliability

The reliability is defined as probability that an item can perform a required function under given conditions for a given time interval.

The estimation of hardware reliability shall be done by using failure rates Failure Rate or Mean Time Between Failures values (reliability measurement) directly from the manufacturer/supplier, or from specific reliability libraries and database, or by using the scientific prediction standards (see [AD24]).

The software reliability shall be addressed during the development phase by application of the following quality assurance and other processes:

1) Software Quality Assurance
   The software development shall be carried out under an applicable quality assurance plan (see, for example, the ACADA Quality Assurance Plan [RD-1]). This quality assurance plan shall be tailored to the specific software project and provided to CTAO for approval.
2) Programming Standards
   The CTAO Software Programming Standards [AD17] are applicable to software developed for CTAO except embedded software and firmware written for controlling devices like Field-Programmable Gate Array (FPGA), Programmable Logic Device (PLD), Programmable Logic Controller(PLC), microcontrollers, etc. Following [AD17] not only makes the code more maintainable but also helps reduce errors.
3) Software Development Lifecycle
   The software development lifecycle processes that will be used in the development of this specific product shall be defined and documented. This describes the key phases in the development of the software indicating the inputs, the deliverables, the resources involved and the methods to be used [AD25]. (For an example see [RD-2]).
4) Software Testing
   For software testing, the recommended indicator is the ‘line test coverage”. The ‘test coverage’ is measured in percentages, demonstrating how much of the software was tested by the software testing tooling at a given time. Within the various indicators that can be measured as ‘test coverage’, CTAO selected the ‘line test coverage’ as the main indicator (i.e. a 60% line test coverage means that 60% of the lines of code inside a software project have been run via tests before the delivery, higher percentage of test coverage gives higher confidence that there are no bugs, since when a tests runs over a piece of code, it is assumed it would trigger bugs contained within). A minimum line test coverage value shall be demonstrated prior to the acceptance of a software release (determined in agreement with IKC).
5) Static Code Analysis
   Static code analysis consists of the analysis of computing programs without executing them. Such analysis allows to identify possible vulnerabilities, potential bugs, and departures from the coding conventions that cannot be identified via software tests. For static code analysis, CTAO recommends using static code analysis frameworks such as the SonarQube framework and/or "linters". These tools shall be configured to adhere to the standard coding guidelines

and the prescriptions specified in the CTA Software Programming Standards document [AD17].

6) Code Peer Review
   CTAO also recommends any new code is peer reviewed before its incorporation into an existing software project. Like for the software testing, certain quality gates obtained from the static code analysis shall be passed prior to the acceptance of a software release.

It should be noted that the approach described above does not attempt to quantify the software failure rate of a product. For the RAM analysis, it is assumed that the software failure rate of the final product will be negligible if the appropriate quality processes are applied.

13.2 Availability

Availability is defined as the probability that the instrument can perform its intended function(s) during 'observation time' (RAM calculation methodology guideline, [AD24]).

The proposed availability estimation method for hardware items preferred by CTAO is the Monte Carlo simulation of the individual systems/subsystems. Once modelled in Reliasoft BlockSim module with required parameters a minimum number of simulations are recommended to estimate the availability. A spare parts count for individual blocks can be obtained following those simulations (for more details refer to CTAO RAM calculation methodology document [AD24]).

As noted in the previous section, within the RAM analysis, the software failure rate is assumed to be negligible in which case it will have no impact on the overall availability of the system.

13.3 Maintainability

Maintainability is a characteristic of design and installation. This characteristic is the measure of the ability of an item to be retained in, or restored to, a specified condition when maintenance is performed by personnel having specified skill levels and using approved procedures and resources.

A good design for maintenance shall follow closely the subsequent principles and practices:

1. Standardization: the goal is to ease the maintenance. Minimize the number of software tools etc. to be used. Reduce costs by minimizing the type and number of spare parts needed in the inventory.
2. Modularization: Create a set of modular units (LRUs) and code: inspectable, testable, and replaceable.
3. Interchangeability: Define interfaces and consider compatibility with other similar functioning parts/software.
4. Accessibility: Ensure easy access to remove and replace if needed by the maintenance crew considering their knowledge and skills. This consideration should also be applied to the deployment of software & firmware upgrades.
5. Malfunction annunciation: Consider minimizing the need for inspection, tools and diagnostic tasks and the time/cost of the corrective maintenance tasks.

6. Fault isolation: Consider designing the system to be as informative as possible such that it not only signals a failure mode, but also narrows down the possible failure mechanisms. Fail-safe principles shall be applied, and, where possible, self-diagnosis and self-recovery are highly recommended.
7. Identification: Name the products with unique identifiers, to help streamlining documentation, procedures, and maintenance tasks. It should be possible to remotely identify the versions of hardware, firmware and software that are deployed.
8. For software development, use tools and procedures configured to adhere to the standard coding guidelines and the prescriptions specified in the CTA- Software Programming Standards document [AD17].

14 Safety

The control system shall raise an alarm when there is an indicator or parameter approaching safety or performance critical criteria previously defined.

In the case of loss of communication with the CTA-X supervisor, the control system shall be able to maintain safe operation of the local control subsystem.

The control system fieldbuses shall be implemented separated from the safety system fieldbuses, therefore preventing any of the two affecting each other in the case of hardware failure.

For the design and integration of safety related parts, this includes both hardware and software components, standard [AD26] shall be followed.

Regarding procedures and conditions to be followed when validating safety related functions and parts, in addition to following the design principles previously mentioned, standard [AD27] shall be followed.