







MINISTERO DELL'ISTRUZIONE, DELL'UNIVERSITÀ E DELLA RICERCA

NORTH-WEST UNIVERSITY YUNIBESITI YA BOKONE-BOPHIRIMA NOORDWES-UNIVERSITEIT

The Mini Array Telescope Control System

Federico Russo – INAF OAS Bologna for the ASTRI Project

AIV SW MEETING























ASTRI-Code Generator

	GET_AMC_MAXTEMPWARNING				
Node ID: TBD	ns=4;s=MAIN.A	ns=4;s=MAIN.AMC_MAXTEMPWARNING			
Description:	Read the set ter	Read the set temperature limit for the Nanotec boards			
Data size/type:	INT64				
Sampling Interval:	On request				
Units:	°C				
Range:	ALARM LOW ALARM HIGH				
Operation States:	MAINTENANCE				

	GET_AMC_I	M1TEMPW	ARNING			
Node ID: TBD	ns=4;s=MAIN	ns=4;s=MAIN.AMC_M1TEMPWARNING				
Description:	Read the act	Read the actual temperature of the Nanotech boards for M1				
Data size/type:	ARRAY <int64> [36]</int64>					
Sampling Interval:	10					
Units:	°C					
Range:	ALARM LOW		ALARM HIGH	MAXTEMPWARNING		
Operation States:	ONLINE		•			

ICD	
ocument	

Excel	File

			۵	D	0	D			0		
		4	A	D	U	U		Г	G	<u> </u>	
			Name of command	Actione	Short name	ADS variable	OPC_UA node	OPC UA Data type	CMD/MODE value	Sampling Interval (s) / ON CHANGE	Default va
											1
	GET AMC M2TEMPWARN	GET	AMC_MAXTEMPWARNING	AMC	MAXTEMPWARNING		ns=4;s=MAIN.AMC_MAXTEMPWARNING	INT64		On request	
Node ID: TBD		GET	_AMC_M1TEMPWARNING	AMC	M1TEMPWARNING		ns=4;s=MAIN.AMC_M1TEMPWARNING	ARRAY <int64> [36]</int64>			
Noue ID. TOD	IIS=4,S=MAIN.AMC_M2TEN	GET	_AMC_M2TEMPWARNING	AMC	M2TEMPWARNING		ns=4;s=MAIN.AMC_M2TEMPWARNING	ARRAY <int64>[3]</int64>			-
Description:	Read the actual temperature	GET	AMC_MAXPOWRCONSUMP	AMC	MAXPOWRCONSUMP		ns=4;s=MAIN.AMC_MAXPOWRCONSUMP	INT64		On request	
Data size/type:	ARRAY <int64>[3]</int64>										
Sampling Interval:	10	GET	_AMC_M1POS	AMC	M1POS		ns=4;s=MAIN.AMC_M1POS	ARRAY <double> [36]</double>			
		GET	AMC_M2POS	AMC	M2POS		ns=4;s=MAIN.AMC_M2POS	ARRAY <double> [3]</double>			
		GET	_AMC_M1POSABS	AMC	M1POSABS		ns=4;s=MAIN.AMC_M1POSABS	ARRAY <double> [36]</double>		On request	
		GET	_AMC_M2POSABS	AMC	M2POSABS		ns=4;s=MAIN.AMC_M2POSABS	ARRAY <double>[3]</double>		On request	
	1	0 GET	AMC_M1ABSLIMITS	AMC	M1ABSLIMITS		ns=4;s=MAIN.AMC_M1ABSLIMITS	ARRAY <double>[72]</double>		On request	
	1	1 GET	_AMC_M2ABSLIMITS	AMC	M2ABSLIMITS		ns=4;s=MAIN.AMC_M2ABSLIMITS	ARRAY <double>[6]</double>		On request	
	1	2 GET	_AMC_M1ACTSTATUS	AMC	M1ACTSTATUS		ns=4;s=MAIN.AMC_M1ACTSTATUS	ARRAY <int64> [36]</int64>			

F. Russo, AIV SW Meeting (19-20 Feb 2024)







A trip among the TCS software levels





- 1. The actual state of the TCS is progressively growing following the Use Cases document.
- 2. After the implementation of the next use case it follows the implementation of a test to validate it. The test is described in the **TCS Software validation Plan**
- 3. A test report is produced and linked to the pipeline run.
- 4. The test name from the Software Validation Plan usually gets the name from the Use Case it tests. If more tests are planned for the same Use Case a count number is appended to the name.



- 1. The actual state of the TCS is pr
- 2. After the implementation of the n The test is described in the **TCS** 3139 3140
- 3. A test report is produced and lin 3143
- 4. The test name from the Software ³¹⁴⁵ more tests are planned for the sa ³¹⁴⁶

- 3138 astri.tcs.TelescopeManagerImpl.TelescopeManagerImplTest
- 3139 astri.tcs.MountSupervisorImpl.MountSupervisorImplTest
 - 40 astri.tcs.TcsMasterImpl.MasterCompClientTest
 - 41 astri.tcs.PMCsupervisorImpl.PMCsupervisorImplTest
- 3142 astri.tcs.EIE.EIE500Test
 - 143 astri.tcs.script.GoTo_Operational
- 3144 astri.tcs.script.GoTo_Park
 - 5 astri.tcs.script.GoTo_ra_dec
 - 6 astri.tcs.script.GoTo_az_el
- 3147 astri.tcs.script.Track_ra_dec_time
- 3148 astri.tcs.script.GetMonitorData
- 3149 astri.tcs.script.SwitchOnOffSQM
- 3150 astri.tcs.script.SwitchOnOffCCamera
- 3151 astri.tcs.script.SwitchOnOffCCameraThermal
- 3152 astri.tcs.TcsTelManagerImpl.TcsTelManagerImplTest



- 1. The actual state of the TCS is progressively growing following the Use Cases document.
- 2. After the implementation of the next use case it follows the implementation of a test to validate it. The test is described in the **TCS Software validation Plan**
- 3. A test report is produced and linked to the pipeline run.
- 4. The test name from the Software Validation Plan usually gets the name from the Use Case it tests. If more tests are planned for the same Use Case a count number is appended to the name.
- 5. To do: show the match of the Junit name with the test name



1. The actual state of the TCS is progressively growing following the Use Cases document.

2.	After the	<pre>@Suite.SuiteClasses({</pre>				
	The test i	<pre>//The test that will be runned with</pre>	the script .tcsRunTest	located in th	ne tcs main	directory
		<pre>// Relevant UC or TC are commented n</pre>	ext to the JUnit list			
		<pre>TelescopeManagerImplTest.class, //</pre>	T-9.1.10-010			
3.	A test re	MountSupervisorImplTest.class,//	T-9.1.10-5201			
	•	<pre>MasterCompClientTest.class,//</pre>	T-9.1.10-030			
		<pre>PMCsupervisorImplTest.class,//</pre>	T-9.1.10-5401			
4.	The test r	EIE500Test class,//	TBD			
	more test	GoTo_Operational.class,//	T-9.1.10-0301			
		GoTo_Park.class,//	T-9.1.10-040			
		<pre>GoTo_ra_dec.class,//</pre>	T-9.1.10-270			
5.	To do: sh	GoTo_az_el.class,//	T-9.1.10-260			
		<pre>Track_ra_dec_time.class,//</pre>	T-9.1.10-280			
		GetMonitorData.class,//	T-9.1.10-380			
		SwitchOnOffSQM.class,//	T-9.1.10-0101			
		SwitchOnOffCCamera.class,//	T-9.1.10-0102			
		SwitchOnOffCCameraThermal.class,//	T-9.1.10-0103			
		TcsTelManagerImplTest.class,//	T-9.1.10-5202			



//The test that will be runned with the script .tcsRunTest located in the tcs main directory // Relevant UC or TC are commented next to the JUnit list

GoTo_Operat:	ional.class,//	T-9.1.10	-0301
GoTo_Park.cl	lass,//	T-9.1.10	-040
GoTo_ra_dec	.class,//	T-9.1.10	- 270
GoTo_az_el.	lass,//	T-9.1.10	- 260
Track_ra_ded		T-9.1.10	- 280
TcsTelManage	erImplTest.class,//	T-9.1.10	- 5202
})	•		
more test	GoTo Operational.class,//		T-9.1.10-0301
	GoTo_Park.class,//		T-9.1.10-040
	GoTo_ra_dec.class,//		T-9.1.10-270
5. To do: sh	<pre>GoTo_az_el.class,//</pre>		T-9.1.10-260
	<pre>Track_ra_dec_time.class,//</pre>		T-9.1.10-280
	GetMonitorData. <mark>class</mark> ,//		T-9.1.10-380
	SwitchOnOffSQM.class,//		T-9.1.10-0101
	<pre>SwitchOnOffCCamera.class,//</pre>	-	T-9.1.10-0102
	_SwitchOnOffCCameraThermal.c	lass,//	T-9.1.10-0103
	IcsIelManagerImpliest.class	,//	1-9.1.10-5202

r١

TCS software Features Status



Actual implementation status of the TCS regarding Mount S.C.A.D.A.:

- The life cycle for the state machine of the telescope is operative.
 - tested with internal Hardware Simulator (I.E. MountDeviceConnector is a mock component with state change capability).
 - tested with external Hardware Simulator (I.E. MountDeviceConnector is connected via opc-ua to a EIE PLC mount simulator).
 - In addition to accept commands from HMI or CCS command line scripts exist to execute **goToPark** and **goToOperational**.

• Motion of the telescope is operative for AZ EL and RA DEC pointing.

- tested with internal Hardware Simulator (I.E. MountDeviceConnector is a mock component with motion capability).
- tested with external Hardware Simulator (I.E. MountDeviceConnector is connected via opc-ua to a EIE PLC mount simulator).
- In addition to accept commands from HMI or CCS command line scripts exist to execute goToAzEI, goToRaDec.

• Motion of the telescope is operative for tracking RA DEC coordinates.

- tested with internal Hardware Simulator (I.E. MountDeviceConnector is a mock component with motion capability).
- tested with external Hardware Simulator (I.E. MountDeviceConnector is connected via opc-ua to a EIE PLC mount simulator).
- In addition to accept commands from HMI or CCS command line scripts exist to execute TrackRaDec.

• Motion of the telescope is operative for Jog Motions.

- tested with internal Hardware Simulator (I.E. MountDeviceConnector is a mock component with motion capability).
- Stop motion command is always possible.

TCS software Features Status



Actual implementation status of the TCS regarding Mount S.C.A.D.A.:

- Switch ON/OFF Cherenkov Camera, Camera Thermal and SQM of the telescope is operative.
 - tested with internal Hardware Simulator (I.E. MountDeviceConnector is a mock component with switch on/off capability).
 - In addition to accept commands from HMI or CCS command line scripts exist to execute switchOnCCamera, switchOffCCamera, switchOffCCameraThermal, SwitchOnSQM, SwitchOffSQM.

• Monitoring for Warnings, Errors and Alarm

- tested with internal Hardware Simulator (I.E. MountDeviceConnector is a mock component with possibility to fake error).
- tested with external Hardware Simulator (I.E. MountDeviceConnector is connected via opc-ua to a EIE PLC mount simulator).
- In addition to accept commands from HMI or CCS command line scripts exist to execute **PrintTCSErrors**.

• Monitoring for detailed information regarding Mount and Telescope Health, C.Camera, Amc, Pmc, SIII

- tested with internal Hardware Simulator (I.E. MountDeviceConnector is a mock component with possibility to generates default info).
- tested with external Hardware Simulator (I.E. MountDeviceConnector is connected via opc-ua to a EIE PLC mount simulator).
- In addition to accept commands from HMI or CCS command line scripts exist to execute GetMonitorData.
- Actually just Void information for Cherenkov Camera, Amc, Pmc, Siii

TCS software specific tools



The TCS software is meant to be run for a single telescope, so in the final configuration 9 TCS's instances will be run in the same time.

- Automatic script that, starting from the TCS "single" CDB, is able to generate a final CDB that configure ACS to work with all the 9 TCS's instances.
- Specific TCS Component Factory that hides from the developer the internal logic used to manage multiple telescopes with a single TCS source code.
- The TCS Component Factory has a build in functionality that allow two levels of debug:
 - Level one uses mocks that simulate the Hardware used in the TCS (no need for OPC-UA servers simulators).
 - Level two (is still a beta version) is able to bypass the ACS manager and can be used to access directly the java code for the implemented component (no need for ACS at all in order to test the code).
 - These algorithms can be programmed to simulates simple behaviours for external components or hardware devices (useful for Junit programming).
- TCS includes multiple scripts used to
 - Setup the ACS environment needed by the TCS.
 - Compile and install the software.
 - Install the needed libraries.
 - run the testCases (one or all the testCases).
 - run the Junits.
 - Run the opc-ua server simulators (one or all the simulator used by the TCS)









	Federico Russo authored 1 week ago	.3.0
	$rightarrow v2.3.0 \ v$ interface-control-idl / telescope-control-	system / idl / + ~
	Name	Last commit
	🕒 MountSupervisor.idl	
	🕒 Supervisor.idl	
	TCSControlExceptions.idl	
	TCSControlExceptions.xml	
	TcsHMIData.idl	
(🕒 TcsMaster.idl	
	🕒 TcsTelManager.idl	
	TelManAndTcsMastDataTypes.idl	
	TelescopeManager.idl	
	🕒 astrocommon.idl	

	Federico Russo authored 1 week ago	rol-idl release 2.3.0
	□ v2.3.0 v interface-control-idl / tele	escope-control-system / idl / + ~
	Name	Last commit
	🕒 MountSupervisor.idl	
	🕒 Supervisor.idl	
TCS	TCSControlExceptions.idl	
	TCSControlExceptions.xml	
	🕒 TcsHMIData.idl	
	🕒 TcsMaster.idl	
	🕒 TcsTelManager.idl	
	🕒 TelManAndTcsMastDataTypes.idl	
	🕒 TelescopeManager.idl	
	🕒 astrocommon.idl	



TelescopeStatusData getTelescopeStatusData();

struct TelescopeStatusData {

string TelescopeName; // The name of the telescope e.g., "M11"/ Instant timestamp; // The timestamp of the current status data FEDECHECK if present in ACS string cameraConfigId; // The name of the Camera config ID, if defined string structureConfigId; // The name of the Structure config ID, if defined tcs::SkyEquatorialTarget skyEquatorialTarget; //Current Ra-Dec as read by the Structure tcs::HorizonDirection horizonDirection; //Current Alt-AZ as read by the Structure TcsTelescopeStatusEnum telescopeState; //The state of the telescope unsigned long long sbId; //ID of the Scheduling Block, if defined unsigned long long obsId; //ID of the Observation Block, if defined

};

TcsMonitoringData getTcsMonitoringData();

//Telescope Status Windows: boolean isCCameraCurrentOn; boolean isPmcCurrentOn; boolean isAmcCurrentOn; boolean isSI3CurrentOn;

boolean isTelescopeRemoteMode; //Remote or Local information

// OPCUA STATUS FOR LOCAL CONTROL SOFTWARE

boolean isCCameraLcsOpcUaRunning; //status of the opc-ua connection TBD boolean isPmcLcsOpcUaRunning; //status of the opc-ua connection TBD

//WARNINGS, ERROR AND ALARM INFO //INTERLOCKS STATUS //Information collected by CCameraSupervisor typedef sequence<double> sipmTemperature; // 37 values typedef sequence<double> hvValues; // 37 values for each camera typedef sequence<double> hvCurrentValues; // 37 values for each camera

Telescope Control System Interfaces Mini-Array



TelescopeStatusData getTelescopeStatusData();

struct TelescopeStatusData {

string TelescopeName; // The name of the telescope e.g., "M11"/ Instant timestamp: // The timestamp of the current status data FEDECHECK if present in ACS string cameraConfigId; // The name of the Camera config ID, if defined string structureConfigId; // The name of the Structure config ID, if defined tcs::SkyEquatorialTarget skyEquatorialTarget; //Current Ra-Dec as read by the Structure tcs::HorizonDirection horizonDirection; //Current Alt-AZ as read by the Structure TcsTelescopeStatusEnum telescopeState; //The state of the telescope unsigned long long sbld; //ID of the Scheduling Block, if defined unsigned long long obsld; //ID of the Observation Block, if defined

};

TcsMonitoringData getTcsMonitoringData();

//Telescope Status Windows: boolean isCCameraCurrentOn: boolean isPmcCurrentOn: boolean isAmcCurrentOn; boolean isSI3CurrentOn;

boolean

boolean

// OPCU

c-ua connection TBD

boolean isPmcLcsOpcUaRunning; //status of the opc-ua connection TBD

🎦 TcsHMIData.idl

//WARNINGS. ERROR AND ALARM INFO

https://www.ict.inaf.it/gitlab/astri/scada/interface-control-idl/-/blob/main/telescope-controltypedef sequence<double> sipmTemperature; // 37 values system/idl/TcsHMIData.idl?ref_type=heads

typedef sequence<double> hvValues; // 37 values for each camera typedef sequence<double> hvCurrentValues; // 37 values for each camera



Derived from ACADA (CTA) IDL:

structure.midl















- Work has been done with the ADAS and OOQS teams in order to define the work flow and the interfaces between the systems.
- Things are moving fast and just yesterday some modifications happened to upper level documents
- TCS UC Live Document: https://docs.google.com/document/d/1849MkzoepUefilrgXnTdqV9t7MfARYClgbRsRjBh0c/edit

ГCS

Internal interfaces (TCS Act As Client)

ADAS

TCS simplified internal Workflow



sd ValidateMotionJob Please note that JobManager is a class of the **TelescopeManager TelescopeManagerimpl** MountSupervisorImpl MountDeviceConnectorimpl PMCsupervisorImpl PMCDeviceConnectorImpl JobManager CentralControl executeSerialJob (SCADA_DM): JobStatus jobManager= createJob (SCADA DM): TcsJob :TcsJob ValidateJob(TcsJob) alt OPT1 IsJobAcceptable(TcsJob) GetStatus() :TCUStatus IsJobAcceptable(TcsJob): boolean GET_PMC_STATUS(LongHolder): int :int :PMCStatus :true :true



Questions?



