MRO FTT/NAS & FLC ICD

FLC to ISS ICD

MRO-ICD-CAM-1200-0113

The Cambridge FTT Team

rev 1.21

22 August 2017

Cavendish Astrophysics

Cavendish Laboratory
JJ Thomson Avenue
Cambridge CB3 0HE
UK

ICD Description

Specific entry from the ICD N^2 Table contained in FDR document.

ICD Number	Sub-systems		Org	Owner	Brief description and preliminary contents
MRO-ICD-CAM-1200-0113	FLC		MRAO	JSY	Interface between FLC control software and ISS, based on the Generic System Interface framework

Change Record

Revision	Date	Author(s)	Changes
0.1	2012-05-02	JSY/EBS/MF	First version
1.0	2013-12-18	JSY	Update for first software release
1.1	2014-01-10	JSY	Added notes on implementation of variants. Minor corrections to other text
1.2	2014-05-22	JSY	Added boolean system properties (were omitted in error). Added fault names
1.3	2014-06-04	JSY	Added AuxTip, AuxTilt, AllowAuxTipTiltIn (FTT only)
1.4	2014-11-27	JSY	Added OpticalGainX, OpticalGainY, AvgCentroidErr
1.5	2015-03-02	JSY	Changes for frame-type-specific background images: new/removed system properties e.g. DarkFullExpTime; modified commands e.g. getDarkFrame; new/removed monitor points e.g. DarkFullFrameSet. Added CameraAirCooling (implemented May 2014)
1.6	2015-03-19	JSY	Added CCDSetTemp
1.7	2015-05-20	JSY	Removed UseCamera and CameraAirCooling monitor points (no need to implement in GSI interface as they directly correspond to system properties). Added getter commands for unit tests
1.8	2015-06-04	JSY	Added startUcamControl/Display commands and port number system properties for the dlmsg interface
1.9	2015-06-08	JSY	Added commands and status item for control of air flow
1.10	2015-06-10	EBS	Removed getter commands. Changed ForceShutdown monitor point to CameraEnable, removed HeaterPower. Replaced obsolete setHeatPower command with setHeatOn and setHeatOff. Added LabJack fixed parameters. Added AirFlowTooLow fault constraints.
1.11	2015-06-25	EBS	Minor edit to track software name changes.
1.12	2015-06-30	JSY	Additional environment controller monitor points (implemented some time ago).
1.13	2015-11-13	EBS	Track environment controller's new ability to shut down camera controller.
1.14	2016-02-23	JSY	Additional system properties, monitor points and faults related to shutdown/power outlet control.
1.15	2017-02-09	EBS	Added temperature and humidity calibration parameters for environment controller.

1.16	2017-05-31	EBS	Added new publish and subscribe commands, changed power board fixed parameters.
1.17	2017-06-16	EBS	Changed LabJack interface names. Removed HeaterPower. Added airflow valve fault.
1.18	2017-06-26	EBS	Track hardware changes to environmental controller fixed parameters and monitor points.
1.19	2017-06-26	EBS	Fixed uncapitalised environmental controller monitor point names.
1.20	2017-07-12	EBS	Track changes to environmental controller monitor point ordering.
1.21	2017-08-22	JSY	Added flow rate calibration parameters for environment controller.

Notification List

The following people should be notified by email that a new version of this document has been issued:

MROI: **Allen Farris**
Andres Olivares

MRAO: **Chris Haniff**
John Young
David Busher
Martin Fisher

Table of Contents

Scope

This document describes the software interfaces of the FLC to the ISS. These comprise the Generic System Interface and the additional network requirements of the software. Interfaces internal to the FLC software and hardware interfaces required by the software are not described.

Acronyms and Abbreviations

API Applications Programming Interface
CSV Comma Separated Values
FLC First Light Camera
FTT Fast Tip-Tilt
FTTA Fast Tip-Tilt Actuator
GSI Generic System Interface
GUI Graphical User Interface
ISS Interferometer Supervisory System
MMCDB MROI Monitor and Configuration DataBase
MROI Magdalena Ridge Observatory Interferometer
NAS Narrow-field Acquisition System
NMT New Mexico Tech
NTP Network Time Protocol
TBC To be confirmed
TBD To be determined
UT Unit Telescope
UTCS Unit Telescope (Mount) Control System
UTM Unit Telescope Mount

Documents

Drawings

Reference Documents

RD1 Technical Requirements: Fast Tip-Tilt/Narrow-field Acquisition System (INT-403-ENG-0003) – rev 2.2, May 20th 2010

RD2 Technical Requirements: First Light Camera (INT-403-TSP-0107) – rev 1.0, May 20th 2010

RD3 Interface spreadsheet for FTTCamSystem – `http://www.mrao.cam.ac.uk/research/OAS/pmwiki/uploads/MROIFastTipTilt.InterfaceControl/FTTNAS-ISS-Spreadsheets.tar.gz`

RD4 Interface spreadsheet for FTTEnvSystem – `http://www.mrao.cam.ac.uk/research/OAS/pmwiki/uploads/MROIFastTipTilt.InterfaceControl/FTTNAS-ISS-Spreadsheets.tar.gz`

RD5 UTCS Design Description (MRO-TRE-OSL-6100-004) – issue 15

RD6 The UTCS to ICS Interface (MRO-TRE-OSL-6100-007) – issue 15

Applicable Documents

AD1 Interface to an MROI System INT-409-ENG-0020 – version 1.5, January 31st 2013

1 Introduction

This ICD, in conjunction with the interface spreadsheets RD3 and RD4, defines the interfaces between the FLC control software and the MROI Interferometer Supervisory System (ISS).

A separate ICD describes the interfaces between the FTT/NAS and the ISS. However, it is important to understand that FLC and FTT/NAS software are not separate implementations (except that there are separate FLC and FTT/NAS editions of the control GUI). Rather there is a single software package, for which a well-defined subset of the functionality satisfies the FLC technical requirements (RD2). The FLC subset has been implemented and tested first, and so the FLC ICD defines the more mature fraction of the functionality (system properties, commands, and monitor points). This subset includes all of the functionality needed to commission the Unit Telescope.

The single implementation has been designed to work with either the FLC or FTT/NAS hardware, with appropriate changes to the values of the system properties.

The ISS is responsible for coordinating the actions of the many quasi-autonomous systems that comprise the interferometer control system. In particular, the ISS is responsible for:

- Starting and configuring the systems;
- Sequencing and coordinating the actions of multiple systems in order to collect science data efficiently;
- Collecting monitor data and storing it in a central database;
- Managing faults and error conditions;
- Shutting down the systems in an orderly fashion at the end of observing (where applicable).

The ISS consists of several components which function in concert to provide supervisory control of the interferometer:

Database Manager Mediates access to the central MROI Monitor and Configuration Database (MMCDB).

Executive Starts and configures the systems. Configuration parameters are obtained from the MMCDB.

Supervisor Commands the systems once they have been initialized.

Data Collector A distributed system of Data Collectors collect monitor data published by the systems and import the data into the MMCBD.

Publish-Subscribe System A service provided by the Database Manager and Data Collectors that allows any system to subscribe to specific monitor data published by any other interferometer systems.

Fault Manager Responds to fault notifications published by systems.

When the various ISS components need to communicate with individual interferometer systems, they do so through a network interface which uses a custom communications protocol over TCP/IP sockets. Systems implement the server side of this communications link, and can accept connections from multiple remote clients. Clients can send commands and receive responses by connecting to one TCP port, and receive monitor data and the results of long-running commands by connecting to another port. Code for encoding and decoding network messages on both sides of this interface is generated from spreadsheets that define the particular commands and monitor variables that each kind of system implements, as well as the types of fault that can be generated. The code-generation framework, known as the Generic System Interface (GSI), is supplied by MRO. There are versions of the GSI that generate Java and C code.

As agreed with MRO, we use different variants of each interface spreadsheet to suit the various deployment scenarios. These are outlined in Sec. 3.1 below. To keep the total number of spreadsheets manageable, there is no separate FLC interface spreadsheet (this is a change from the approach suggested at PDR). Instead, the accompanying spreadsheets define all of the commands and monitor points anticipated for the final FTT/NAS software implementation. For convenience, the FLC ICD lists only the subset that satisfies the FLC requirements. The other commands and monitor points are implemented such that they can be executed/published, however in early releases their functionality may be limited.

Thus the steps needed to interface a system (or collection of related systems) to the ISS are:

- Create a spreadsheet based on an MRO-supplied template that defines:
  - The system classes;
  - The commands that instances of each class accept, their parameters, and whether they should run in a separate thread (such commands are called "asynchronous" commands);
  - The monitor points that instances of each class publish and their data types;
  - The types of fault that may be published.
- Write a suitable `main()` function that initializes the network interface.
- Write custom state change actions that will be executed during transitions between states of the standard model for GSI-based systems (for example actions to initialize and shut down the system).
- Write custom methods that implement the commands defined in the spreadsheet.
- Optionally, write custom getter methods that return new values of the monitor points defined in the spreadsheet (these are only needed if the monitor points are to be polled at intervals initially defined by the spreadsheet and adjustable by the client; otherwise systems may publish monitor data at times of their choosing using generated publish methods).
- Run the GSI code generation tool on the spreadsheet to generate code that encodes or decodes network messages corresponding to the defined commands and monitor points.
- Compile the custom methods and link these against the generated code and a static code library.
- The result will be an executable that implements the network interfaces to the ISS and calls the appropriate custom code to actually execute commands and get monitor points.

This ICD accompanies the six interface spreadsheets (three variants of each of two systems) in order to provide some additional context for the definitions they contain.

Besides providing the network interface, the GSI also provides a standalone execution environment, in which commands may be executed from a script or by calling their C methods directly, rather than being sent from a remote client. The direct method call approach allows any command to be executed whereas the scripting implementation does not support all of the possible datatypes.

2 Requirements

Many of the top-level requirements that apply to the FLC (RD1) will only be satisfied if the control software operates correctly. The MROI Generic System Interface is an integral part of that control software, even when it is being used in standalone mode with no network connection to the ISS.

The network interface to the ISS is required to realize the target acquisition functionality, as telescope steering corrections/offloads are sent to the Unit Telescope Mount's control system (UTCS) via the ISS (this design choice was at MRO's request), and certain information published by the UTCS that is needed by the FLC will be obtained via the ISS Publish-Subscribe System (PSS).

The FLC software requires the following interfaces external to the FLC hardware:

- Ethernet network support (100BASE-T or faster), for communication with the ISS and between the software components. However, the software can also work in stand-alone mode, where all the components run on one computer.
- A network interface to the ISS, based on the GSI framework outlined above. This enables the FLC to communicate with the ISS (and the local UTM) over the network using the GSI. The software makes use of the following GSI facilities:
  - Control.
  - Monitor point publishing.
  - Error and exception handling.
  - The publish/subscribe system. Necessary for telescope communication.
- An external networked computer running Linux, for the control GUI.
- An external networked computer for the analysis GUI. The preliminary analysis GUI requires an installation of Python.
- A Network Time Protocol (NTP) server running on the local network, for accurate timestamping of FLC data.

3 Design

The software consists of four independent components (Figure 1):

1. The system controller is responsible for controlling the first light camera and using camera data to perform source image acquisition.
2. The environment controller is responsible for maintaining a thermally safe camera environment and collecting environmental FLC data.
3. The control GUI permits interactive control, monitoring, and logging of data from the system controller and the environment controller when equivalent ISS facilities are not available or not convenient. There are distinct FLC and FTT/NAS editions of the control GUI. The FLC edition has a simpler user interface and can log image centroids to CSV files, as requested by AMOS, as well as FITS.
4. The analysis GUI analyses and graphs data logged by the control GUI.

The system controller and the environment controller communicate with the ISS over the network using the GSI protocol. They also communicate with the control GUI over the network using the dlmsg protocol, a messaging system developed by Cambridge.

The control GUI logs and saves data generated by the system controller, the environment controller and itself. These data can be analysed later by the analysis GUI.

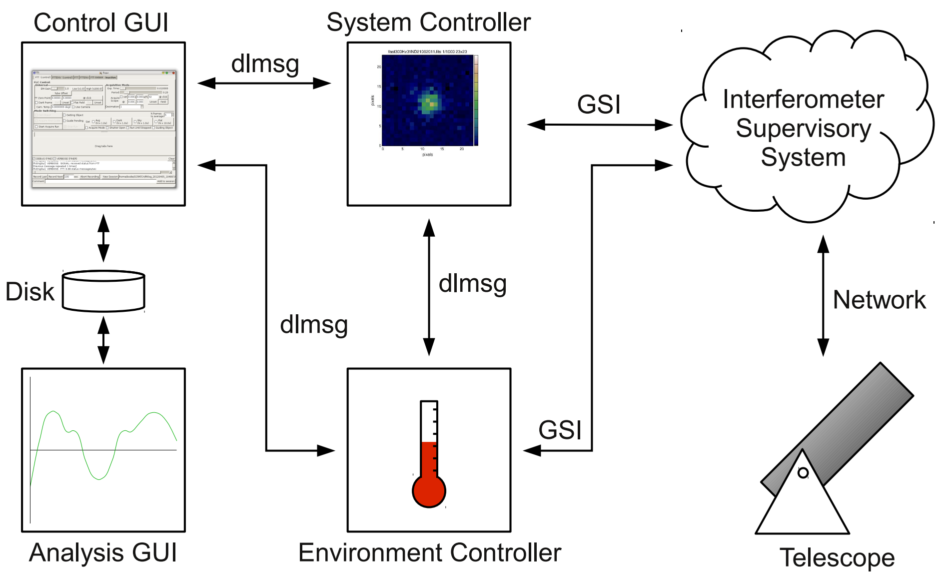

Figure 1: ISS-centric overview of the FLC software.

The network requirements depend on how the FLC software is deployed. The system is designed to be very flexible in this respect. Some possible scenarios are:

- The four components all run on the FLC computer and communicate via loopback interfaces. In this case, no network is required, this is a complete standalone system, however telescope communication will not be available.
- The environment controller runs on an independently powered embedded processor while the other components run on the FLC computer. This may be regarded as the minimum "safe" configuration as it provides awareness of the environment even when the FTT camera is completely powered down. If both computers are located within thecontrol rack, this scenario only requires network functionality within that rack.
- The system controller runs on the FLC computer, the environment controller runs on the embedded processor, and the control GUI and analysis GUI run remotely, for example on a laptop computer in the telescope enclosure or in the interferometer control room. This requires an operational network but no networked GSI interface is required.
- The system controller runs on the FLC computer, the environment controller runs on the embedded processor, and the control GUI and analysis GUI run remotely. The ISS also has access to the controllers via the GSI. Depending on the configuration, either the control GUI or the ISS has control of the system, the other can only monitor the system and issue commands that do not change the system state.
- The system controller runs on the FLC computer, the environment controller runs on the embed-

ded processor, and they both communicate with the ISS via the networked GSI. The control and analysis GUIs are not needed. This is the expected final configuration of the system.

The system controller also needs to communicate with the local telescope to send steering corrections and receive the current rotation matrices and (if applicable) guide target offsets. This communication relies on availability of the GSI publish-subscribe system.

3.1 System variants

The software comprises several variants of each of the system controller and environment controller, for use in the deployment scenarios outlined above and for testing the system. Each variant:

- has a unique executable whose name follows the convention outlined below;
- uses one of several interface spreadsheets to define the accepted commands;
- implements the network interfaces to the control GUI only if needed for the intended deployment scenario(s).

Matching filenames are used for the executable and interface spreadsheet, comprising the components outlined below.

1. A prefix indicating the role of the executable (the same spreadsheet is used irrespective of the executable prefix).

   utest_ Executes a unit test suite.

   test_ Executes a standalone test script.

   start_ Creates, initializes, and starts a system instance in standalone mode without using a script. During initialization, system properties are read from a file specified on the command line.

   (no prefix) System server application.

2. The basic name of the system (FTTCamSystem or FTTEnvSystem).

3. A suffix describing any network interface to the control GUI.

   (no suffix) No control GUI interface.

   _ucamcontrol Full control GUI interface; GSI interface spreadsheet excludes commands that change the system state.

   _ucamdisplay Limited control GUI interface that does not allow clients to change the system-state.

Each interface spreadsheet defines a unique name for the system, comprising the last two components of the executable filename.

The particular variants that will be implemented are summarised in Table 2. Note that test executables are not installed by `make install`. The ucamdisplay spreadsheet differs from the base "no suffix" spreadsheet by the addition of a command to start the (limited) GUI interface. The ucamcontrol spreadsheet omits all of the state change commands of the base system, and adds a command to start the GUI interface. The three spreadsheets also use different names for the system as mentioned above. In all other respects the spreadsheets are identical for the three variants.

Executable	Spreadsheet (.ods)	Description
utest_FTT[X]System	FTT[X]System	Standalone unit test by direct method calls
FTT[X]System	FTT[X]System	System server without control GUI interface
utest_FTT[X]System_ucamcontrol	FTT[X]System_ucamcontrol	Standalone unit test via control GUI interface
start_FTT[X]System_ucamcontrol	FTT[X]System_ucamcontrol	Standalone with control GUI interface
FTT[X]System_ucamcontrol	FTT[X]System_ucamcontrol	System server with control GUI interface
start_FTT[X]System_ucamdisplay	FTT[X]System_ucamdisplay	Standalone with display-only GUI interface
FTT[X]System_ucamdisplay	FTT[X]System_ucamdisplay	System server with display-only GUI interface

Table 2: System variants. Each entry in the table describes a pair of systems, the system controller FTTCamSystem and the environment controller FTTEnvSystem.

3.1.1 C Implementation

The need to have several alternative system names for FTTCamSystem and FTTEnvSystem introduces a number of difficulties for sharing code between variants in the C implementation. In brief, the following differences between C code generated from different spreadsheets must be accommodated:

- The generated source files have different filenames.
- The C names of the system object differ.
- The state change action methods have different names (e.g. `initializeFTTCamSystem_ucamdisplayAction()`) (because the method names incorporate the system name).

Our solution to these difficulties involves post-processing the generated C code, the steps being as follows:

1. Generate C header and source files for the server and Java source for the client from the interface spreadsheet (e.g. `FTTCamSystem_ucamcontrol.ods` with system name “FTTCamSystem_ucamcontrol”).
2. Perform a global replacement of the system name from the spreadsheet (e.g. “FTTCamSystem_ucamcontrol”) with “FTTCamSystem” in the generated C header and source files. The Java client source is not modified.
3. Rename the C header and source files to `FTTCamSystem.h` and `FTTCamSystemInterface.c` .
4. Use the post-processed C header and source files to build the executables for that spreadsheet.
5. Repeat the process for the next spreadsheet, overwriting the previous post-processed C header and source files with new ones.

The post-processing described above is more or less equivalent to using a common system name for all variants on the server side, but different ones on the client side according to the spreadsheet used. The code-generation could in principle be modified to avoid the need for this post-processing.

A further issue is that the ucamcontrol variants omit the state-change commands from their spreadsheet, but prototypes for these command methods are still needed to build the DLMsg interface. We deal with this by introducing a (non-generated) header file `FTTCamSystemIncludes.h` containing the required declarations.

4 Interface Control

The GSI interfaces are defined precisely in the interface definition spreadsheets. The following is a summary of the interfaces for the system controller and the environment controller.

4.1 System controller interface

Some context is helpful for understanding the system controller interface. When the system is in the standard "Operational" state it can operate in either of two modes:

1. Idle. The camera is not reading out images, but commands are still accepted and monitor points are published once the relevant commands (startPublish and startUcamControl or startUcamDisplay have been received.
2. Acquisition. The camera is reading out full frame images using the manufacturer's conventional clocking scheme. This state is intended for acquisition of celestial objects.

4.1.1 System properties

The system properties are configuration values set up when the program is started, either by the ISS if available or via the GSI stand-alone execution environment if it is not. They are not expected to change during execution, unless specifically stated. All of these properties will be stored as string data types in the central configuration database.

Since there is no separate FLC system controller implementation (as explained above), it is necessary to supply dummy values for the additional FTT/NAS system properties enumerated in MRO-ICD-CAM-1100-0112. The values of the additional system properties have no impact on the operation of the commands listed in this ICD.

University of Cambridge dlmsg interface These parameters define system connections from the Cambridge University control GUI and to the environment controller when the dlmsg interface is used.

- **UcamSysCmdPort** Command server port number, to accept commands from the control GUI. The control GUI "SupervisorPort" configuration parameter must match.
- **UcamSysDataPort** Data server port number, to send data to the control GUI. The control GUI "ServerPort" configuration parameter must match.
- **UcamConnectToEnv** Boolean ("true" or "false"). When true, the system will connect to the environment controller and watch its ***CameraEnable*** monitor point. If that value should become false, the system will commence an orderly shutdown.
- **UcamEnvIpAddr** The Internet Protocol address of the environment controller. It is prudent to use a numerical address in case no name server is available (for example, during network failure outside the control rack).
- **UcamEnvDataPort** Environment controller data server port number; value must match FTTEnvSystem property UcamEnvDataPort.

Emulation The system can be configured to work if the EMCCD camera is not present.

- **UseCamera** Boolean ("true" or "false"). Enable/disable use of the real camera (otherwise an internal simulation is used).

Electron multiplying gain These are either preset limits on allowed electron multiplying gain, or preset values that can be used to acquire standardised data.

- **MinEMGain** Minimum allowed electron multiplying gain.
- **MaxEMGain** Maximum allowed electron multiplying gain.
- **LowEMGain** Standardised low electron multiplying gain.
- **HighEMGain** Standardised high electron multiplying gain.

Exposure time These values are either preset limits on allowed exposure time, or preset values that can be used to acquire standardised data.

- **MinExpTime** Minimum allowed exposure time.
- **MaxExpTime** Maximum allowed exposure time.
- **AvgFullFrameExpTime** Exposure time used to acquire full frames for averaging (see getAvgFrame).
- **DarkFullFrameExpTime** Exposure time used to acquire dark full frames for averaging (see getDarkFrame and getSkyFrame).
- **FlatExpTime** Exposure time used to acquire flat fields for averaging (see getFlatField).

Periods Limits on camera readout period in acquisition mode.

- **MinAcquirePeriod** Minimum allowed frame period in acquisition mode.
- **MaxAcquirePeriod** Maximum allowed frame period in acquisition mode.

Acquisition tolerance Some commands transition automatically between acquisition and idle mode when an object is acquired. This parameter determines when the change is made.

- **AcquireTolerance** Threshold that determines when an object is considered to be acquired.
- **AcquireFrameAvg** Number of frames to average when calculating AcquireTolerance.

Hardware constants These values are intrinsic to the camera hardware and the optics. They are needed for calculating the pixel scale and field size.

- **MaxPixelValue** Maximum pixel value allowed by the camera hardware.
- **PixelSize** Length of the edge of each pixel.
- **DetectorWidth** Width of the sensitive area of the CCD.
- **DetectorHeight** Height of the sensitive area of the CCD.
- **CameraFocalLength** Focal length of the local first light camera lens.
- **OpticalGainX** Factor to account for optical geometry and reflections in X axis.
- **OpticalGainY** Factor to account for optical geometry and reflections in Y axis.
- **UTMagnification** Magnification of the unit telescope.
- **Nas0** A calibrated constant that describes the rotation between the pixel frame and the azimuth/elevation frame.
- **CameraAirCooling** Boolean (“true” or “false”). Enable/disable use of the camera’s internal fan for cooling. Enable this if an external chilled fluid supply is not available.

- **CCDSetTemp** The desired temperature set-point for the EMCCD camera.

Seeing estimation These are the parameters used to estimate the seeing and related parameters from image data.

- **SeeingAvgTime** Time over which *AvgSeeingFWHM* and *AvgPhotonRate* monitor points are calculated.

Offload period This is the (reciprocal of the) maximum rate at which UTM offloads are sent to the telescope.

- **OffloadPeriod** Period of UTM offloads sent

4.1.2 Control

Commands can be broadly divided into three categories:

- Configuration modifiers. These commands modify the configuration of the system controller.
- Readout commands. These commands initiate camera readout in acquisition mode. Some continue camera readout until told to stop, others progress through a preset sequence of readout tasks.
- Information requests. These commands return information about the system settings or tell the system to start publishing to the ISS or to subscribe to the environment controller "cameraEnable" monitor point.

The commands themselves are as follows:

University of Cambridge control GUI interface These commands enable the communications interface with the Cambridge control GUI.

- **startUcamControl** Enable full control via University of Cambridge dlmsg protocol. (Only available in ucamcontrol variants).
- **startUcamDisplay** Enable viewing via University of Cambridge dlmsg protocol, with no state changes allowed. (Only available in ucamdisplay variants).

Configuration modifiers These commands modify the configuration of the system controller.

Electron multiplying gain These commands can be called in any mode. They modify the electron multiplication gain setting of the camera.

- **setEMGain** Set electron multiplying gain to specified value.
- **setToLowEMGain** Set electron multiplying gain to the LowEMGain preset value.
- **setToHighEMGain** Set electron multiplying gain to the HighEMGain preset value.

Tube offset These commands can be called in any mode. They tell the system controller whether the local telescope is in tube offset mode or not. This information is needed to calculate the camera pixel scale.

- **setTubeOffsetOn** Inform the system that the telescope is in tube offset mode.
- **setTubeOffsetOff** Inform the system that the telescope is not in tube offset mode.

Zero point This command can be called in any mode. It sets the target image position on the EMCCD.

- **setTipTiltZeroPoint** Set the target position (in EMCCD pixel coordinates, prior to application of telescope dispersion and off-axis offsets) for acquisition.

Reference object selection These commands work in acquisition mode. They select the "reference object", the image of the celestial object to be acquired, which may not be the science target.

- **setAcquireScopeRect** Set reference object to brightest object within predefined rectangle.
- **setAcquireScopeField** Set reference object to brightest object in entire field.
- **unsetAcquireScope** Unset reference object.

Camera frame timing These commands set the camera frame timing to be used in acquisition mode.

- **setAcquirePeriod** Set the time between EMCCD acquisition frames.
- **setAcquireExpTime** Set exposure time for each acquisition frame.

Decimation Decimation is a method of reducing network bandwidth and storage requirements. If the decimation is n, then only one camera frame in every n is published. Other data taken at the camera frame rate are bundled together so that n frames' worth are published at the same time. Hence there is one image published for every bundle published.

- **setAcquireDecimation** Set the decimation n for acquisition frames.

Background subtraction These commands control the use of previously acquired or generated images to de-bias the centroids calculated from the pixel data. The system stores separate dark frames for each of three frame types: full frames, conventional readout subframes, and custom readout subframes.

- **setDarkFrame** Use the provided image for dark subtraction when capturing the specified frame type. The integer-typed first parameter specifies the frame type: 0 = full frame; 1 = conventional readout subframe; 2 = custom readout subframe. The latter two frame types are not relevant to the FLC command subset.
- **unsetDarkFrame** Do not do dark subtraction for the specified frame type.
- **setFlatField** Use the provided image for flat field correction.
- **unsetFlatField** Do not do flat field correction.

Readout commands These commands initiate or stop camera readout in acquisition mode. Some continue camera readout until told to stop, others progress through a preset sequence of readout tasks.

Observational readout These commands are intended for use in acquiring objects.

- **startAcquireRun** Change to acquisition mode and read out full frame continuously.
- **getObject** Change to acquisition mode, acquire and return when object is acquired.
- **acquireAndGuide** Change to acquisition mode and acquire until stopRun is called.
- **stopRun** Stop any ongoing readout.

Background subtraction These commands acquire sets of images and average them. The images can be used later to de-bias centroids calculated during readout. The system properties specify a total of seven standardised exposure times: two exposure times (for average frames and dark/sky frames) for each of three frame types (full frame, conventional readout subframe, custom readout subframe) as well an exposure time for flat fields (which are always full frames).

- **getAvgFrame** Returns a standardised camera frame of the specified type (0 = full frame; 1 = conventional readout subframe; 2 = custom readout subframe) that is an average of the specified number of frames.
- **getDarkFrame** Like getAvgFrame but with the camera shutter closed.
- **getSkyFrame** Like getAvgFrame but taken on the sky.
- **getFlatField** Like getAvgFrame but taken against an (assumed) uniformly illuminated field.

Information requests These commands return information about the system settings or current state, or tell the system to start publishing to the ISS or to subscribe to the environment controller "cameraEnable" monitor point.

- **getFixedParams** Retrieve settings that are constant over the program run time.
- **startPublish** Start publishing monitor points (DLMsg messaging is not affected).
- **startSubscribe** Subscribe to the environment controller's "cameraEnable" monitor point.

4.1.3 Monitor points

Monitor points for the system controller can most conveniently be grouped by frequency of generation. Note that several combinations of these groups can be being published, depending on the state of the system.

Information about the status of the system controller is sent at a rate that gives "instant" feedback on a human timescale (currently set to 10 Hz). Monitor points in this category include:

TubeOffsetMode Boolean. True if the system has been told the telescope is in tube offset mode.

CameraReadout Boolean. True if the camera is reading out.

ShutterOpen Boolean. True if the shutter is open.

CameraTemp Floating point. The camera CCD chip temperature.

ExpTime Floating point. The current exposure time.

EMGain Integer. The electron multiplying gain, on an arbitrary scale set by Andor.

AcquirePeriod Floating point. The time between frames to be used in acquisition mode.

AcquireExpTime Floating point. The duration of exposures to be used in acquisition mode.

AcquireReadoutTime Floating point. The time required to read out an image in acquisition mode.

AcquireMaxExpForPeriod Floating point. Maximum possible exposure time available for *AcquirePeriod*.

AcquireDecimation Integer. The decimation rate, used to reduce the number of acquisition mode images sent.

AcquireRectX Integer. X pixel coordinate of acquisition rectangle.

AcquireRectY Integer. Y pixel coordinate of acquisition rectangle.

AcquireRectWidth Integer. Pixel width of acquisition rectangle.

AcquireRectHeight Integer. Pixel height of acquisition rectangle.

OffloadGain Floating point. Multiplier for telescope offload values prior to being sent for telescope correction.

TipTiltZeroPointX Floating point. X pixel coordinate of target closed loop servo position, prior to application of telescope dispersion and off-axis offsets.

TipTiltZeroPointY Floating point. Y pixel coordinate of target closed loop servo position, prior to application of telescope dispersion and off-axis offsets.

ObjectivePointX Floating point. X pixel coordinate of target position, after application of telescope dispersion and off-axis offsets.

ObjectivePointY Floating point. Y pixel coordinate of target position, after application of telescope dispersion and off-axis offsets.

RotT11 Floating point. Element T_{11} of the rotation matrix that converts between Nasmyth and RA/Dec coordinates. This depends on the telescope pointing.

RotT12 Floating point. Element T_{12} of the rotation matrix.

RotT21 Floating point. Element T_{21} of the rotation matrix.

RotT22 Floating point. Element T_{22} of the rotation matrix.

AcquireScopeSet Boolean. True if acquisition scope is set.

AcquireActive Boolean. True if acquisition mode is active

GettingObject Boolean. True if the getObject command is executing.

AcquiringAvg Boolean. True if the system is in the process of acquiring data for an average frame.

AcquiringDarkAvg Boolean. True if the system is in the process of acquiring data for a dark average frame.

AcquiringSkyAvg Boolean. True if the system is in the process of acquiring data for a sky average frame.

AcquiringFlatAvg Boolean. True if the system is in the process of acquiring data for a flat field.

GuidingObject Boolean. True if the system is in acquisition mode, has successfully acquired an object, and is continuing to keep it acquired.

RunUntilStopped Boolean. True if executing a (synchronous) command that causes the camera to read out indefinitely, such as startAcquireRun.

GuidePending Boolean. True if executing a command that will eventually cause the system to guide in acquisition mode (currently only acquireAndGuide).

DarkFullFrameSet Boolean. True if a dark frame (which could be a sky frame) is being used to do dark subtraction for full frames.

FlatFieldSet Boolean. True if a flat field is being used to do flat field corrections.

High bandwidth data is sampled at the current camera frame rate, which is expected to be as high as 34 Hz in acquisition mode. If the current decimation rate is n, then data from every n frames is chunked together into time series prior to publication to improve the network efficiency. The exception is the camera images themselves, only one in every n is published to save bandwidth and storage. If the system is idle, none of this information is published.

High bandwidth monitor points include:

CentroidPixX Floating point time series. X coordinate of calculated image centroid.

CentroidPixY Floating point time series. Y coordinate of calculated image centroid.

CentroidValid Boolean time series. True if the system thinks that the calculated centroid value is valid.

OffsetX Floating point time series. The X angle between the current centroid and the target position.

OffsetY Floating point time series. The Y angle between the current centroid and the target position.

InstantSeeingFWHM Floating point. Spatial seeing estimate (expressed as a conventional long exposure image FWHM) from the latest frame.

InstantSeeingErr Floating point. Estimated error in *InstantSeeingFWHM*.

AvgSeeingFWHM Floating point. Spatial seeing averaged over SeeingAvgTime.

AvgSeeingErr Floating point. Estimated error in *AvgSeeingFWHM*.

AvgCentroidErr Floating point. RMS distance of the centroid from the tip-tilt zero point.

InstantPhotonRate Floating point. Detected photons/sec for the tip-tilt reference object from the latest frame.

AvgPhotonRate Floating point. Photon rate averaged over SeeingAvgTime.

Frame Integer 2D array. Camera image.

Finally, telescope steering corrections ("offloads") are published at the rate at which they are sent to the telescope, which is either a fixed rate (expected to be 1 Hz) or the camera frame rate if that is lower. The offload values and their timestamps must be supplied to the dedicated FTTA offload socket interface described in Sec. 8.11 of RD5. It is also MRO's responsibility to configure this interface appropriately and to activate it prior to use of the target acquisition mode of the FLC. We will provide details of how the offload interface should be configured in a later revision of this ICD. If the system is idle, none of this information is published.

The telescope steering correction monitor points are;

OffloadX Floating point. X value of UTM offload.

OffloadY Floating point. Y value of UTM offload.

4.1.4 Subscriptions

Subscriptions which the FLC must access from the UT via the publish subscribe system are listed below with a short description, the status word and either the matrix coefficient names or units in the case of a scalar value, as described in RD6.

- Telescope rotation matrix for conversion between Nasmyth and RA/Dec coordinates:
  `tarray t00,t01,t10,t11`

- Outer axis position
  `outer degs`

To represent RA/DEC coordinates on user displays the rotation correction between the detector coordinate frame and the Nasmyth focal plane must be determined, and the rotation matrix between the Nasmyth focal plane and RA/DEC must be known. The former requires the combination of the Nas0 parameter (available from the FLC system properties) and the outer axis position variable `outer` which must be subscribed to. The latter requires the rotation matrix `tarray` to be subscribed to. The Nas0 system property must be consistent with its equivalent in the UTCS.

If the Cambridge University dlmsg interface is not in use, one subscription is required from the FLC environment controller:

- **CameraEnable** Boolean. True if the environment controller is allowing the camera to run.

4.1.5 Faults

CameraTooHot Shut down camera and program if the camera temperature gets too hot.

4.2 Environment controller interface

4.2.1 System properties

The system properties of the environment controller define software and hardware interfaces and the environmental conditions within which the camera has permission to operate.

University of Cambridge control GUI interface These parameters define the port numbers on which the system listens for connections from the the control GUI. The values must match the corresponding control GUI configuration parameters.

- **UcamEnvCmdPort** Command server port number.
- **UcamEnvDataPort** Data server port number.

LabJack interface These parameters identify each of the LabJacks used for analog and I^2C interfacing, so they can be automatically identified when plugged into the host computer's USB ports.

- **UseInterface** Boolean ("true" or "false"). When true, the environment controller will use the Labjacks. Otherwise, software emulation is used.
- **LabJack1Id** Identifier (serial number on base) of LabJack "1" (closest to Raspberry Pi in chassis).
- **LabJack2Id** Identifier (serial number on base) of LabJack "2".

Temperature Temperature conditions.

- **EnclosureAirMaxTemp** Maximum allowed air temperature within the camera enclosure.
- **EnclosureAirMinTemp** Minimum allowed air temperature within the camera enclosure.
- **EnclosureAirMinPassiveTemp** Minimum allowed air temperature within the camera enclosure for the camera to be enabled.
- **CoolantMaxBelowAmbient** The coolant entry temperature must be at least this close to the enclosure air temperature.

- **CoolantMinAboveDewPoint** The coolant entry temperature must be warmer than the dew point by this amount.
- **EnclosureSurfMaxAboveAmbient** The external enclosure surface must be less than this amount above the ambient temperature.
- **EnclosureSurfMaxBelowAmbient** The external enclosure surface must be less than this amount below the ambient temperature.
- **EnclosureSurfMaxPassiveBelowAmbient** The external enclosure surface must be less than this amount below the ambient temperature before the camera can be enabled.

Humidity Humidity conditions.

- **EnclosureMinHumidity** Minimum allowed humidity within the camera enclosure.
- **EnclosureMaxHumidity** Maximum allowed humidity within the camera enclosure.

Flow rate Flow rate conditions. These only produce warnings if the conditions are not met. The camera still has permission to operate.

- **MinCoolantFlow** The minimum coolant flow needed.
- **MinAirFlow** The minimum air flow needed.

Calibration Parameters used to improve sensor accuracy. For each listed sensor, the raw output is multiplied by the gain and an offset is added to the result. Hence the calibration is a simple linear remapping.

- **EnclosureAirTempGain** Enclosure air temperature gain adjustment.
- **EnclosureAirTempOffset** Enclosure air temperature offset adjustment.
- **EnclosureHumidityGain** Enclosure humidity gain adjustment.
- **EnclosureHumidityOffset** Enclosure humidity offset adjustment.
- **CoolantOutTempGain** Coolant exit temperature gain adjustment.
- **CoolantOutTempOffset** Coolant exit temperature offset adjustment.
- **CameraMountTempGain** Camera mount temperature gain adjustment.
- **CameraMountTempOffset** Camera mount temperature offset adjustment.
- **CameraCaseTempGain** Camera case temperature gain adjustment.
- **CameraCaseTempOffset** Camera case temperature offset adjustment.
- **CoolantInTempGain** Coolant entry temperature gain adjustment.
- **CoolantInTempOffset** Coolant entry temperature offset adjustment.
- **HeaterTempGain** Heater temperature gain adjustment.
- **HeaterTempOffset** Heater temperature offset adjustment.
- **EnclosureTableTempGain** Table temperature near enclosure gain adjustment.
- **EnclosureTableTempOffset** Table temperature near enclosure offset adjustment.
- **EnclosureSurfTempGain** Enclosure surface temperature gain adjustment.

- **EnclosureSurfTempOffset** Enclosure surface temperature offset adjustment.
- **EnclosureMountTempGain** Enclosure mount temperature gain adjustment.
- **EnclosureMountTempOffset** Enclosure mount temperature offset adjustment.
- **AmbientTempGain** Ambient temperature gain adjustment.
- **AmbientTempOffset** Ambient temperature offset adjustment.
- **AmbientHumidityGain** Ambient humidity gain adjustment.
- **AmbientHumidityOffset** Ambient humidity offset adjustment.
- **BaseplateTableTempGain** Table temperature near baseplate gain adjustment.
- **BaseplateTableTempOffset** Table temperature near baseplate offset adjustment.
- **BaseplateTemp1Gain** Baseplate temperature 1 gain adjustment.
- **BaseplateTemp1Offset** Baseplate temperature 1 offset adjustment.
- **BaseplateTemp2Gain** Baseplate temperature 2 gain adjustment.
- **BaseplateTemp2Offset** Baseplate temperature 2 offset adjustment.
- **BaseplateTemp3Gain** Baseplate temperature 3 gain adjustment.
- **BaseplateTemp3Offset** Baseplate temperature 3 offset adjustment.
- **BaseplateTemp4Gain** Baseplate temperature 4 gain adjustment.
- **BaseplateTemp4Offset** Baseplate temperature 4 offset adjustment.
- **BaseplateTemp5Gain** Baseplate temperature 5 gain adjustment.
- **BaseplateTemp5Offset** Baseplate temperature 5 offset adjustment.
- **BaseplateTemp6Gain** Baseplate temperature 6 gain adjustment.
- **BaseplateTemp6Offset** Baseplate temperature 6 offset adjustment.
- **BaseplateTemp7Gain** Baseplate temperature 7 gain adjustment.
- **BaseplateTemp7Offset** Baseplate temperature 7 offset adjustment.
- **CoolantFlowGain** Coolant flow rate gain adjustment.
- **CoolantFlowOffset** Coolant flow rate offset adjustment.
- **AirFlowGain** Dry air flow rate gain adjustment.
- **AirFlowOffset** Dry air flow rate offset adjustment.

Shutdown and power outlet control Parameters that affect how the environment controller shuts down the FLC system, including removal of power to the camera and the FLC computer.

- **FTTShutdownEnable** Boolean ("true" or "false"). When true, the environment controller is permitted to shutdown the camera system software and the FLC computer.
- **FTTPowerControlEnable** Boolean ("true" or "false"). When true, the environment controller is permitted to use the ethernet-switched mains power board within the control rack to cut the power to the camera and the FLC computer.

- **FTTShutdownSeconds** The time between detection of an adverse environmental condition and shutdown of the camera system software and the FLC computer.
- **FTTPowerdownSeconds** If **FTTPowerControlEnable** is true, this is the time from the end of the **FTTShutdownSeconds** countdown until the mains power is removed from the camera and the FLC computer.
- **FTTShutdownUser** The username and internet protocol address of the user on the FLC computer whose account is used to initiate a shutdown of that computer.
- **FTTPowerIPAddr** The internet protocol address of the ethernet-switched power board.
- **FTTPowerIPPort** The internet protocol port of the ethernet-switched power board.
- **FTTPowerLogin** The string used to log in to the ethernet-switched power board.
- **FTTPowerComputerOutletNumber** The outlet number on the ethernet-switched power board that supplies power to the FLC computer.
- **FTTPowerCameraOutletNumber** The outlet number on the ethernet-switched power board that supplies power to the camera Peltier cooler.

4.2.2 Control

The environment controller has only one operational state, in which it publishes environment measurements, (optionally) controls a heater in the camera enclosure and (optionally) enables or disables camera operation. The command set is as follows:

University of Cambridge control GUI interface These commands enable the communications interface with the Cambridge control GUI.

- **startUcamControl** Enable full control via University of Cambridge dlmsg protocol. (Only available in ucamcontrol variants).
- **startUcamDisplay** Enable viewing via University of Cambridge dlmsg protocol, with no state changes allowed. (Only available in ucamdisplay variants).

Heater commands These commands control the camera enclosure heater or allow it to work automatically.

- **setHeatAuto** Allow the environment controller to control the heater state automatically.
- **setHeatManual** Control the heater manually.
- **setHeatOn** When manual heater control is enabled, turn the heater on.
- **setHeatOff** When manual heater control is enabled, turn the heater off.

Camera control commands These commands control whether the system controller uses the camera, or allow the environment controller to do so.

- **setCameraAuto** Allow the environment controller to shut down the FLC if it detects adverse environmental conditions.
- **setCameraManual** Allow manual camera shutdown. Cancels any shutdown countdown initiated in auto mode.
- **setCameraDisable** When manual camera control is on, initiate the FLC shutdown procedure.

Air flow control commands These commands activate the solenoid valve that turns the flow of dry air on or off.

- **setAirFlowOn** Turn on air flow to the camera enclosure.
- **setAirFlowOff** Turn off air flow to the camera enclosure.

Information requests These commands retrieve the environment controller fixed parameters or start publishing.

- **getFixedParams** Retrieve settings that are constant over the program run time.
- **startPublish** Start publishing monitor data to the ISS (this does not affect DLMsg messages).

4.2.3 Monitor points

The Generic System Interface monitor points for the environment controller generate scalars, booleans and Durations at a constant frequency, expected to be 1 Hz. The majority of these are environment data but there are a few status monitor points too. Environment controller monitor points include:

EnclosureAirTemp Floating Point. Air temperature inside the camera enclosure.

EnclosureAirHumidity Floating Point. Humidity inside the camera enclosure.

EnclosureAirDewPoint Floating Point. Calculated dew point inside the camera enclosure.

CoolantOutTemp Floating Point. Temperature of coolant as it exits the camera enclosure.

CoolantInTemp Floating Point. Temperature of coolant prior to entering the camera enclosure.

CameraCaseTemp Floating Point. Temperature of the camera case.

HeaterTemp Floating Point. Temperature of the heater inside the camera enclosure.

CameraMountTemp Floating Point. Temperature of the camera mount.

EnclosureTableTemp Floating Point. Temperature of the optical table near the enclosure.

EnclosureSurfTemp Floating Point. Temperature of the external enclosure surface.

EnclosureMountTemp Floating Point. Temperature of the external enclosure mount.

CoolantFlowRate Floating Point. Coolant flow rate.

AirFlowRate Floating Point. Flow rate of dry air blown onto the camera window.

LabJack1Temp Floating Point. Internal temperature of LabJack1.

AmbientTemp Floating Point. Air temperature outside the camera enclosure.

AmbientHumidity Floating Point. Humidity outside the camera enclosure.

AmbientDewPoint Floating Point. Dew point outside the camera enclosure.

BaseplateTableTemp Floating Point. Temperature of the optical table near the baseplate.

BaseplateTemp1 Floating Point. Baseplate temperature1.

BaseplateTemp2 Floating Point. Baseplate temperature2.

BaseplateTemp3 Floating Point. Baseplate temperature3.

BaseplateTemp4 Floating Point. Baseplate temperature4.

BaseplateTemp5 Floating Point. Baseplate temperature5.

BaseplateTemp6 Floating Point. Baseplate temperature6.

BaseplateTemp7 Floating Point. Baseplate temperature7.

LabJack2Temp Floating Point. Internal temperature of LabJack2.

HeaterAuto Boolean. True if the heater is being controlled by the application.

HeaterOn Boolean. True if the heater is on.

AirFlowOn Boolean. True if air flow to the camera enclosure is on.

CameraAuto Boolean. True if the camera enable is being controlled by the application.

CameraEnable Boolean. True if the environment controller is allowing the camera to run.

PowerControl Boolean. True if the environment controller has established control of the ethernet-switched power board.

ShutdownPending Boolean. The environment controller has detected an adverse environmental condition and has initiated the shutdown countdown.

PowerdownPending Boolean. The FLC shutdown is complete and the powerdown countdown is underway.

TimeToShutdown Duration. The time until the FLC is shut down.

TimeToPowerdown Duration. The time until power to the FLC computer and camera is cut.

ComputerPowerOn Boolean. Is the FLC computer power on?

CameraPowerOn Boolean. Is the camera power on?

4.2.4 Subscriptions

No subscriptions.

4.2.5 Faults

EnclosureTooHot Warn if enclosure is too hot.

EnclosureTooCold Warn if enclosure is too cold.

EnclosureTooDry Warn if enclosure is too dry.

EnclosureTooHumid Warn if enclosure is too humid.

Condensation Warn if coolant temperature is below dew point.

Coolant Warn if coolant temperature is more than 5 degrees below ambient.

CoolantFlowTooLow Warn if coolant flow is too low.

AirFlowTooLow Warn if air flow is too low. This fault is only enabled if the air flow is on and has been on for a fixed period of time (currently five seconds). This is to prevent warnings when the air flow valve is shut, and when the air flow valve has just been opened (before the air has reached the air flow sensor).

AirFlowValveFault The air flow valve driver is signaling a fault, which could be driver overheating or overcurrent or undercurrent.

HeaterControlFailed Warn if an error occurred when switching the heater on or off.

PowerInterfaceFailed Warn if an error occurred when querying or setting the state of an ethernet-switched power outlet.