

|  |  |
| --- | --- |
| Software  Requirements  Specification | Diamond Light Source Ltd  Harwell Science and Innovation Campus  Chilton, Didcot  Oxfordshire OX11 0DE  United Kingdom  T +44 (0)1235 778000  F +44 (0)1235 778499  [www.diamond.ac.uk](http://www.diamond.ac.uk) |

I12-MIRO-CTRL-REQ-0001

EPICS Interface for Miro 310M High Speed Camera

Prepared by: Thomas Connolley

Approved by: Mark Heron Controls Group Leader <Date>

Issue: 1.1

Date: 20/06/2016

Abstract:

This document provides a software requirement specification for an EPICS interface to the Vision Research Miro 310M high speed camera used on Beamline I12.

[Blue text, such as this, is provided to guide the author and reviewer. In most cases it should not be deleted. You may wish to delete this text instance however as original author.

To use this template:

1. Set the values of Title, Document Reference and Revision Number properties via File | Properties
2. Select all the text (CTRL-A). Press F9 to update the selected text.
3. Edit the date in the header. With the header in edit mode select all the text (CTRL-A). Press F9 to update the selected text.
4. Add entry to Version History]

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Date | Changes |
| 1.0 | Thomas Connolley | 16/05/2016 | First draft |
| 1.1 | Thomas Connolley | 20/06/2016 | Revised specification of camera functions required. In summary, access to the functions described in the Ethernet Protocol is required. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Project Personnel

|  |  |
| --- | --- |
| Project Originator | Michael Drakopoulos |
| Project Leader | Thomas Connolley |
| Controls Engineering Lead | Andrew Foster |

Contents

[1 Introduction 4](#_Toc451163077)

[1.1 Document Organisation 4](#_Toc451163078)

[2 System Overview 5](#_Toc451163079)

[2.1 Product Functions 5](#_Toc451163080)

[2.2 User Characteristics 5](#_Toc451163081)

[2.3 Design and Implementation Constraints 5](#_Toc451163082)

[2.4 User Documentation 5](#_Toc451163083)

[2.5 Assumptions and Dependencies 5](#_Toc451163084)

[3 External Interface Requirements 6](#_Toc451163085)

[3.1 User Interfaces 6](#_Toc451163086)

[3.2 Hardware Interfaces 6](#_Toc451163087)

[3.3 Communications Interfaces 6](#_Toc451163088)

[4 Functional Requirements 7](#_Toc451163089)

[5 Analysis Diagrams 8](#_Toc451163090)

[6 Other Non-Functional Requirements 9](#_Toc451163091)

[6.1 Performance Requirements 9](#_Toc451163092)

[6.2 Safety Requirements 9](#_Toc451163093)

[6.3 Security Requirements 9](#_Toc451163094)

[6.4 Software Quality Attributes 9](#_Toc451163095)

[6.5 Business Rules 9](#_Toc451163096)

[7 Other Requirements 10](#_Toc451163097)

[Appendix A: To Be Determined List 11](#_Toc451163098)

[Appendix B: Glossary 12](#_Toc451163099)

[Appendix C : References 13](#_Toc451163100)

# Introduction

[Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem. ]

Since becoming operational in 2009, Beamline I12 has used commercial CMOS high speed cameras as part of its X-ray imaging system. The original camera was a **Vision Research Phantom v.7.3**, replaced in 2015 by a **Vision Research Miro 310M**. Neither camera has been integrated with the DLS standard EPICS control system. In use, the camera is set up and operated using Vision Research’s “Phantom Camera Control” (PCC) software, which is a Windows application running on beamline server i12-pcoedge01. Important hardware integration for timing, triggering and gating has been carried out by I12 staff using ZEBRA and digital I/O. Further integration and automation of experiments, particularly for high speed tomography, is hindered by the lack of an EPICS interface. For example, the camera’s exposure time and frame rate has to be interactively set by the user in PCC, when such parameters could be calculated and set automatically by a Python or GDA script. Similarly, specifying the save location for recorded data has to be done interactively. The current setup leaves room for human error, restricts automation and limits the throughput of experiments. To make further progress with high speed camera integration and experiment automation, an EPICS interface to the camera is required.

The scope of the project is to provide an EPICS interface for the Miro 310M high speed camera with associated EPICS EDM screens. The project will focus on “Day 1” requirements to give the essential control functions, with the ability to add to this functionality at a later date. So as not to rule out future functionality, the EPICS interface should as far as possible provide access to the full functionality enabled by the Ethernet Protocol “PH16 Camera Protocol” [[[1]](#endnote-1)] specified by the document provided by Vision Research.

## Document Organisation

[Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.

This document is intended for developers and testers (Controls Group staff and contractors), and end users (Beamline Scientists and Beamline Staff).

The document details what functions the EPICS interface must provide and how it is to behave, focusing on “Day 1” requirements.

# System Overview

## Product Functions

 [Summarise the major functions the product must perform or must let the user perform. Details will be provided in Section 4, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top- level data flow diagram or object class diagram, is often effective. ]

The EPICS interface must provide a user interface to essential camera setup, operation and image data download functions, which will be a subset of the functionality provided by the supplier’s PCC software. The EPICS interface should as far as possible provide the full functionality enabled by the Ethernet Protocol “PH16 Camera Protocol” specified by the document provided by Vision Research.

The required “Day 1” functionality is as follows:

### Camera Settings

* Get/set recorded bit depth
* Get/set number of memory partitions (Default is 1)
* Put camera into acquisition mode
* Send software trigger signal to camera
* Send an abort recording signal to stop acquisition if required.

### Cine Settings

* Get/set cine number (i.e. the partition into which the cine should be recorded)
* Get/set cine name
* Get/set resolution
* Get/set sample rate in frames per second
* Get/set exposure time in microseconds.
* Get/set Extended Dynamic Range in microseconds
* Command camera to perform a “Current Session Reference” (CSR)
* Get/set image range and trigger position
* Delete current cine from on-board memory (e.g. prior to starting a new recording)

### Flash Memory

* Same image cine from camera buffer to flash memory
* Get flash memory size
* Get remaining flash memory available
* Get number of files in flash memory
* Get names of files in flash memory
* Erase flash memory

### Advanced Settings

* Get/set external sync type (Internal, External)
* Get/set frame delay
* Get camera temperature
* Get sensor temperature
* Get/set trigger to rising or falling edge
* Get/set trigger filter time
* Get/set ready signal ends at: Trigger or Recording end
* Get/set what the Aux pin is set to (Strobe, Event or Memory Gate)

### Image Saving

The commercial PCC software enables saving in a variety of file types, but not hdf. If possible, an hdf writer shall be provided, so that an image sequence downloaded from the camera is saved to HDF format, along with the camera metadata.

* Select image file to download to central storage (e.g. the one in the camera buffer or one stored in Flash Memory)
* Get/set image download file path
* Get/set file type (default to HDF if possible)
* Get number of stored images in on-board cine file
* Get/set output image range (Full cine or user defined)
* Set 16 bit depth
* Get/set output file name

## User Characteristics

[Identify the general characteristics of the intended users of the product. Classes of user may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy. ]

The software will be used by Beamline Scientists, Beamline Controls and Data Acquisition Staff, and beamline users, either directly or through GDA.

## Design and Implementation Constraints

[Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software). ]

## User Documentation

[List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards. ]

Documentation suitable for users and EPICS/GDA programmers is to be provided. The preferred format for this is a page or pages in the I12 (JEEP) section on DLS’s “Confluence” system.

## Assumptions and Dependencies

[List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan). ]

## Assumptions

* The Ethernet Protocol as described in “PH16 camera protocol” [i] can be used to communicate with the camera. Setting of all camera parameters and functions is possible using just the Ethernet Protocol.
* Following discussions with Andy Foster and Allan Greer, The Software Development Kit (SDK) provided by Vision Research is NOT required for developing an EPICS interface.
* The EPICS developer on the project will have access to the web-based Vision Research “Phantom Zone” and “Developer Zone” in case they need to ask for additional information.
* Live streaming of images from the camera is NOT essential, because the camera has a hardware PAL video-out signal which can be fed to the control room. However, if live streaming can be implemented without laborious programming effort, it should be considered.
* A 1 Gbit copper Ethernet link to the camera is adequate. (10 Gbit copper Ethernet may be technically feasible but is unlikely to work due to the cable lengths).
* 10 Gbit Ethernet may be used as a future upgrade, possibly using copper to fibre conversion at the camera.
* Backwards compatibility to the Phantom 7.3 camera is not required.
* The camera will be available on a Diamond network to allow remote access for software development and off-beamline testing.
* The EPICS IOC will not be able to run at the same time as the PCC software, particularly if the EPICS interface is provided for a Linux platform.
* Functions in the Ethernet Protocol specific to a colour camera are not required.

## Dependencies

* The Ethernet Protocol as described in “PH16 camera protocol” [i].

# External Interface Requirements

## User Interfaces

[Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification. ]

The User Interface will be provided within the existing EPICS synoptic.

Access to EPICS PVs is required via the command line (caget/caput).

A GDA XML interface file is required.

## Hardware Interfaces

[Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used. ]

A schematic of the hardware as currently implemented is shown in Figure 1. User interaction with the camera is through the PCC software running on the detector server. All communication between the camera server and the camera is over Ethernet. The protocol is described in a document provided by Vision Research [i].

Setting up the camera is done through PCC. For example, setting the frame rate, exposure time and resolution. Once imaging parameters are set, a “Current Session Reference” (a black reference) can be performed and is strongly recommended.

The PCC software is also used to set parameters like the source of the frame synchronisation (F-SYNC) signal, which is usually either the camera’s internal clock or an external pulse source on BNC coaxial connector.

The camera has 12 GB of fast memory which continually takes images from the sensor as a First-In, First-Out (FIFO) buffer. Images are stored in a proprietary .cine format. The buffer can be partitioned by the user if necessary.

When a user is ready to record images, the camera is put into Acquire mode. Any existing .cine files in the buffer are erased (with a prompt for confirmation by the user) and acquisition starts. The camera records images into the 12GB memory on a FIFO basis until a trigger signal is received, which can be delivered by software or hardware. On receiving the trigger signal, the image currently being acquired is marked as Image 0. Images before the trigger have negative numbers; images after the trigger have positive numbers. The number of images to store after the trigger is specified by the user. Once the end of the buffer is reached, recording stops. The user can then download the image sequence for saving to storage elsewhere (e.g. Beamline central storage). Or, the image sequence can be saved to on-board flash memory for later download. Before saving, the user can interactively view and edit files, for example to only download the images of interest rather than the whole contents of the buffer.

After image download or saving to flash, the user can set up for their next acquisition sequence, which may or may not have the same parameters as the previous one.

It is possible to set up an autosave to on-board flash memory, with automatic re-start of recording once saving is complete.

As well as hardware frame synchronisation and trigger signals, the camera can be supplied with a hardware memory gating signal on the AUX pin. When the gate is high, images are recorded in the 12 GB buffer. When the gate is low, the sensor is still read out but the images are not saved. The gating feature is used a lot on the beamline when performing multiple tomography scans, with defined gaps between each scan. Such “gap scans” have been implemented with Python code and hardware position compare to the tomography stage using Zebra electronics.

## Communications Interfaces

[Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms. ]

See Figure 1 for a schematic of the current camera setup on Beamline I12.

Following preliminary discussions of Version 1.0 of this specification, it is anticipated that the camera interface will run on a server dedicated to the camera, in which case it will be under a Linux operating system. A similar communications interface is anticipated, with the camera on a separate Ethernet connection to the server, while the server itself is on the beamline network. Having the camera directly on the beamline network could result in unacceptably high network traffic.



Figure 1: Schematic of hardware layout for MIRO camera as currently implemented on I12. Hardware I/O signals Trigger, F-sync and Memory Gate come from other beamline hardware, usually Zebra or digital I/O.

# Functional Requirements

4.1 Enter name of group of related requirements

The functional requirements of the EPICS interface are described, by analogy with the functionality provided by the “PCC” software. Manuals for the “Console” software and Ethernet protocol are available. For full functionality, the commands available in the Ethernet Protocol must be exposed to the user in the EPICS interface.

| Title | Description | Priority |
| --- | --- | --- |
| Ethernet | The Ethernet interface with the camera will be supported. | Must |
| HDF output | Saving downloaded image sequences and metadata in HDF format | Highly desirable |
| Live view | View of live image from camera | Optional |
| Camera Settings | Get/set recorded bit depth  Get/set number of memory partitions (Default is 1)  Put camera into acquisition mode  Send software trigger signal to camera  Send an abort recording signal to stop acquisition if required. | Must |
| Cine Settings | Get/set cine number (i.e. the partition into which the cine should be recorded)  Get/set cine name  Get/set resolution  Get/set sample rate in frames per second  Get/set exposure time in microseconds.  Get/set Extended Dynamic Range in microseconds  Command camera to perform a “Current Session Reference” (CSR)  Get/set image range and trigger position  Delete current cine from on-board memory (e.g. prior to starting a new recording)  Select autosave to on-board flash memory  Set restart recording options if autosave is set | Must |
| Flash Memory | Save image cine from camera buffer to flash memory  Get flash memory size  Get remaining flash memory available  Get number of files in flash memory  Get names of files in flash memory  Erase flash memory | Must |
| Advanced Settings | Get/set external sync type (Internal, External)  Get/set frame delay  Get camera temperature  Get sensor temperature  Get/set trigger to rising or falling edge  Get/set trigger filter time  Get/set ready signal ends at: Trigger or Recording end  Get/set what the Aux pin is set to (Strobe, Event or Memory Gate) | Must |
| Image Saving | Select image file to download to central storage (e.g. the one in the camera buffer or one stored in Flash Memory)  Get/set image download file path  Get/set file type (default to HDF if possible)  Get number of stored images in on-board cine file  Get/set output image range (Full cine or user defined)  Set 16 bit depth  Get/set output file name  Option of automated file saving at end of recording  Get/set restart recording options | Must |

# Analysis Diagrams

[Add a Context diagram, state charts or any other diagrams that help to clarify the functional behaviour of the system – not the design.]

To be determined (if required).

# Other Non-Functional Requirements

## Performance Requirements

[If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features. ]

To be determined.

## Safety Requirements

[Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. ]

TBD

## Security Requirements

[Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied. ]

The high speed camera will be connected to a dedicated camera server on the I12 beamline. No specific security beyond the normal beamline network and user logon protocols is required.

## Software Quality Attributes

[Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning. ]

### Adaptability

Once core software functions are established and tested, the software should be adaptable for more complicated tasks.

### Availability

The software should be available as part of the main EPICS EDM interface for the I12 beamline.

### Correctness

The software must correctly set and readback process variables to the high speed camera.

### Reusability

It would be useful if the software could be installed and used on other camera servers on I12, and other beamlines, particularly B16 or I13. Portability may require a portable camera server which can be assigned an appropriate IP address for other beamlines.

### Robustness

Robustness of the software is essential. Failure of a test or experiment due to problems with the EPICS-camera interface must be less than 1 per 1000 hours of user beamtime.

### Testability

The EPICS interface will be tested for functionality and benchmarked against the functionality provided by the PCC software supplied with the camera.

### Usability

Usability will be aided if the appearance and layout of the interface is clear and structured according to different types of parameter. For example, those functions most frequently used should be grouped together. EPICS EDM is sufficiently flexible that this can be achieved.

## Business Rules

[List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules. ]

No specific business rules apply.

# Other Requirements

[ Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalisation requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project. ]

TBD

Appendix A: To Be Determined List

[ Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure. ]

Appendix B: Glossary

[ Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS. ]

TBD – To be decided.

Appendix C : References

[ List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location. ]

1. [] “ PH16 camera protocol” Vlad Iordachescu, Radu Corlan, Version 2.3 February 2, 2014. Downloaded from Vision Research May 2016. S:\Technical\Engineering\Beamline Eng\I12 (JEEP)\manuals\MIRO-310M-Camera\Ethernet\_Protocol\v16proto-2.3.pdf [↑](#endnote-ref-1)