 

**SPICA-VIS**

**Instrument Control Software**

Authors: Philippe Berio and Denis Mourard

Doc. No. : SPICA-VIS-0002

Issue: 2

Date: 10/07/2020

**CHANGE RECORD**

|  |  |  |  |
| --- | --- | --- | --- |
| **ISSUE** | **DATE** | **SECTION** | **COMMENTS** |
| 1 | 13/02/2020 | All | Creation PB |
| 1.1 | 09/03/2020 | Spectro/Sources/CHARA | Blue addition DM |
| 1.2 | 15/06/2020 | 4 and 5 | New motors added and server sections filled PB |
| 2 | 10/07/2020 | All | Ready for Design review |

Table of Contents

[1. Scope 4](#_Toc43822739)

[2. Motorized devices 4](#_Toc43822740)

[2. Detectors 5](#_Toc43822742)

[3. Sources 5](#_Toc43822743)

[4. Power control 5](#_Toc43822744)

[5. Software 6](#_Toc43822745)

[5.1. Context 6](#_Toc43822746)

[5.2. Devices 6](#_Toc43822747)

[5.2.1. spica\_pdc\_server 6](#_Toc43822748)

[5.2.2. spica\_adc\_server 6](#_Toc43822749)

[5.2.3. spica\_periscope\_server 7](#_Toc43822750)

[5.2.4. spica\_ip-ctrl\_server 8](#_Toc43822751)

[5.2.5. spica\_tt-tracker\_server 8](#_Toc43822752)

[5.2.6. spica\_delayline\_server 9](#_Toc43822753)

[5.2.7. spica\_shutter\_server 10](#_Toc43822754)

[5.2.8. spica\_grating\_server 11](#_Toc43822755)

[5.3. Detectors 11](#_Toc43822756)

[5.3.1. spica\_scidet\_server 11](#_Toc43822757)

[5.3.2. spica\_ipdet\_server 12](#_Toc43822758)

[5.4. Power 13](#_Toc43822759)

[5.4.1. spica\_power\_server 13](#_Toc43822760)

# Scope

The scope of this document is to describe the low-level control Software of SPICA-VIS. It presents the list of motorized devices and detectors and the way they are operated.

# Motorized devices

The naming convention of the modules and the motors is presented in the product tree (see SPICA-VIS-0004).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Modules | Motors | Number | Type | Controller | Client/Server |
| SFO (SPICA Feeding Optics) | FOP | 1 | DC motor | VEGA controller | spica\_periscope\_server |
| SFO (SPICA Feeding Optics) | TT-nTP-nn∈[1,6] | 12 | Newport 8816 | Newport 8742 | spica\_ip-ctrl\_server |
| PDC (Polarization Delay Compensator) | PDC-nn∈[2,6] | 5 | Rotation Stage with Stepper Motor  | R256 – Lin Engineering | spica\_pdc\_server |
| ADC (Atmospheric Dispersion Compensator | AF-nAS-nn∈[1,6] | 12 | Rotation Stage with Stepper Motor | R256 – Lin Engineering | spica\_adc\_server |
| BAS (Beam Adjustment System) | PH-nPV-nn∈[1,6] | 12 | Newport 8814 | Newport 8742 | spica\_ ip-ctrl \_server |
| BAS (Beam Adjustment System) | FT-nFP-nn∈[1,6] | 6 | PI | DNA Cube | spica\_tt-tracker\_server |
| SFI (Spatial Filter Input mode) | DL-nn∈[2,6] | 5 | Newport MFA-PP  | Newport SMC100PP | spica\_delayline\_server |
| BSL (Beam Selector) | SH-nn∈[1,6] | 6 | ThorlabsMFF101/M |  | spica\_shutter\_server |
| BCS (Beam Control System) | BCC | 1 | Newport MFA-PP  | Newport SMC100PP | spica\_ipdet\_server |

Table 1: Motorized Devices of the injection table of SPICA-VIS

1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | Names | Number | Type | Controller | Client/Server |
| Grating wheel |  |  |  |  | spica\_grating\_server |
| Wavelength adjusting |  |  |  |  | spica\_grating\_server |

Table 2: Motorized Devices of the spectrograph table of SPICA-VIS

Most of the controllers listed in Table 1 have a serial connection. We plan to operate them through the network thanks to serial device servers ([Nport5610](https://www.moxa.com/en/products/industrial-edge-connectivity/serial-device-servers/general-device-servers/nport-5600-series/nport-5610-16)).

SPICA/VIS operations will also require communicating with different specific CHARA servers:

* STS
* LDC
* MIRCX/SPICA-FT

# Detectors

SPICA-VIS uses 2 detectors:

* Science Detector: ANDOR 1024x1024. It will probably be operated with the USB interface. Some tests will be done as soon as possible in order to be sure that the Cameralink interface is not necessary.
* Pupil/Image Control Detector: ANDOR IXON 897 512x512. During observations, it will be operated up to 200Hz in order to optimize the flux injected into the single mode optical fibers. This frequency cannot be achieved with the standard USB interface. It is necessary to use the Cameralink interface. The data from the detector are transferred to the computer via a dedicated Cameralink frame grabber (Bitflow Neon – aleady tested on CESAR experiment).

# Sources

SPICA-VIS will use two main sources:

* retrofeed: white light source for retro-feeding the single mode fibers and for the recording of the reference positions on the Image and Pupil Control Detectors.
* spectral lamp: We plan to use one or two wavelength calibration light sources (<https://www.oceaninsight.com/products/light-sources/calibration-sources/wavelength-calibration-sources/>) connected through multimode fibers to the V-groove.

# Power control

SPICA-VIS devices will be connected to remotely controlled plugs. The list is presented below.

# Software

## Context

The device and camera control software should be implemented as CHARA client/server applications. A low level (or engineering) GUI should be developed for each client/server application. Each SPICA-VIS server will be registered to CHARA sockman.

## Devices

## spica\_pdc\_server

This server accepts 15 commands (3 commands for 5 beams, the crystal plate is fixed for beam 1):

* HOME: go back to the home position of the rotation stage
* MOVE *angle*: go to the angle position
* MOVE *step*: go to the step position

The layout of the GUI is presented in Figure 2.



Figure 1: spica\_pdc\_server GUI

## spica\_adc\_server

This server accepts 36 commands (3 commands per prism and per beam):

* HOME: go back to the home position of the rotation stage
* MOVE *angle*: go to the angle position
* MOVE *step:* go to the step position

The layout of the GUI is presented in Figure 3.

 Figure 2: spica\_adc\_server GUI

## spica\_periscope\_server

This server accepts 2 commands only:

* UP: go to the SPICA position. The light from the telescopes or STS goes to SPICA.
* DOWN: go to the CHARA position. The light from the CHARA source goes towards the telescopes.

The layout of the GUI is presented in Figure 1.



Figure 3: spica\_periscope\_server GUI

## spica\_ip-ctrl\_server

This server accepts 24 commands, 1 command per axis (24 axes). Each command corresponds to a relative movement:

* MOVE value

The layout of the GUI is presented in Figure 4.



Figure 4: spica\_ip-ctrl\_server GUI

## spica\_tt-tracker\_server

This server accepts 12 commands, 1 command per axis (12 axes). Each command corresponds to a relative movement:

* MOVE value

The layout of the GUI is presented in Figure 5.



Figure 5:spica\_tt-tracker\_server GUI

## spica\_delayline\_server

This server accepts 15 commands, 3 commands per axis (5 axes):

* MOVE value: relative movement
* MVABS value: absolute movement
* HOME: go back to home position

The layout of the GUI is presented in Figure 6.



Figure 6: spica\_delayline\_server GUI

## spica\_shutter\_server

This server accepts 18 commands, 3 commands per shutter (6 axes):

* OPEN
* CLOSE
* STATUS

The layout of the GUI is presented in Figure 7.



Figure 7: spica\_shutter\_server GUI

## spica\_grating\_server

As the mechanical design of the spectrograph is not started, it is difficult to define exactly this server. Hereafter, we define the minimal high-level commands, but we will certainly add some low-level commands in order to command each motorize axis directly.

This server accepts 3 commands:

* LOW: insert the Low-Resolution dispersive optics
* MEDIUM value: insert the Low-Resolution dispersive optics and select the central wavelength
* HIGH value: insert the High-Resolution dispersive optics and select the central wavelength

The layout of a minimal GUI is presented in Figure 9.



Figure 9: spica\_grating\_server GUI

## Detectors

## spica\_scidet\_server

This server grabs the frame recorded by the science detector and copy them into a shared memory. This shared memory is connected to the scientific RTD and to a process dedicated to save the frames into FITS files. Both processes are described in the document SPICA-VIS-0003 (SPICA-VIS-observingsoftware). In addition, this server accepts commands to configure the detector:

* START: start acquisition
* STOP: stop acquisition
* SHUTTER OPEN: open the internal shutter of the ANDOR detector
* SHUTTER CLOSE: close the internal shutter of the ANDOR detector
* GET DIT: get the current DIT (Detector Integration Time)
* GET SIZE: get the windowing (LOW, MED or HIGH)
* GET TEMP: get the detector temperature
* GET GAIN: get the detector gain
* SET DIT: set the current DIT
* SET SIZE: set the windowing (LOW, MED or HIGH)
* SET TEMP: set the detector temperature
* SET GAIN: set the detector gain

The layout of the GUI is presented in Figure 10. The UPDATE button allows to read and display the current DIT, GAIN, Temperature and Size.



Figure 10: spica\_scidet\_server GUI

## spica\_ipdet\_server

This server grabs the frame recorded by the Image/Pupil detector and copy them into a shared memory. This shared memory is connected to the Image/Pupil RTD, to a process dedicated to save the frames into FITS files and to a process dedicated to servo the position of the bright speckle on to a reference pixel corresponding to the core of the optical fiber. The 3 processes are described in the document SPICA-VIS-0003 (SPICA-VIS-observingsoftware). In addition, this server accepts commands to configure the detector:

* START: start acquisition
* STOP: stop acquisition
* SHUTTER OPEN: open the internal shutter of the ANDOR detector
* SHUTTER CLOSE: close the internal shutter of the ANDOR detector
* GET DIT: get the current DIT (Detector Integration Time)
* GET SIZE: get the windowing (Full Frame or Fast TipTilt windowing)
* GET TEMP: get the detector temperature
* GET GAIN: get the detector gain
* SET DIT: set the current DIT
* SET SIZE: set the windowing (Full Frame or Fast TipTilt windowing)
* SET TEMP: set the detector temperature
* SET GAIN: set the detector gain
* PUPIL: insert pupil optics
* IMAGE: remove pupil optics

The layout of the GUI is like the one of spica\_scidet\_server (see Figure 11). One additional button will be inserted in order to insert/remove pupil optics.

## Power

## spica\_power\_server

This server accepts 12 commands:

* R ON: Switch on retrofeed source
* R OFF: Switch off retrofeed source
* S ON: Switch on spectral calibration source
* S OFF: Switch off spectral calibration source
* SC ON: Switch on scientific detector
* SC OFF: Switch off scientific detector
* IPC ON: Switch on image pupil detector
* IPC OFF: Switch off image pupil detector
* IT ON: Switch on injection table devices
* IT OFF: Switch on injection table devices
* ST ON: Switch on spectrograph table devices
* ST OFF: Switch on spectrograph table devices

The layout of the GUI is presented in Figure 11. In the future, we could imagine adding new switches for the injection table, one per module (SFO, ADC, PDC, …) for example.



Figure 11: spica\_power\_server GUI