python_andor Documentation

Release 1.0

Jonathan St-Antoine

May 22, 2019

CONTENTS

class python_andor.**command**

acq(*nbExp*, *expTime*, ***kwargs*)

Description: This function triggeres an acquisition sequence

nbExp: number of exposure

expTime: exposure time in seconds

Options: –nowoof–: [True/False] Do not start the watch dog (does not block the main thread)

Return: return void once the nb. of exposure is acquired

acq_mode(*mode*)

Description: This function sets the acquisition mode. Choices are : Target, video,flat,dark

Return: return -1 if fails, void if successfull

fw(*filtre*)

Description: This function set the filter.

Note: If this function fails it will return -1 and write NA in the header

initialize()

Description: Initialization sequences. This function will open a tclsh (version 8.5) console, then upload source client.tcl. Afterward a series of command will be sent to the window computer. Mostly, tcl source files will be uploaded.

Return: void

script(***kwargs*)

Description: This function will execute a script. The user will be prompt the enter the target name, exposure time and object name. The header, acquisition mode and the path will be automatically updated.

Options: –video–: Not used interactivally. Will trigger the video function. Use the video function –no_header: Used to not querry telinfo and telmeteo. If the script function fails the first time because of the telmeteo or telinfo, you can manually use the set_header function and then use script(no_header=True).

Note:

1. Use the stop_acq() function to stop the acquisition of th script.
2. the video flux will be automatically launched after exposuretime+2 secondes

send(*cmd*)

Description: This function directly sends command to the window tclsh console. Do not use this function while another function is running

set_header()

Desceription: This function will set the header of the futur images. BonOMM must be open otherwise the function will fail.

Return: This function returns 0 if successfull or -1 if unsuccessfull

set_path()

Description: This function will set the window working path. E.g., /190521/Target/test-omm/.

Note: This function is useless if the class is used interactively

Return: This function will return 0 if succesffull or -1 if it failed.

`stop_acq()`

Description: This function will stop a image acquisition

`video(**kwargs)`

Description: This function will start the video flux. The user will be ask to enter the exposure time

Note:

1. use the stop_acq() function to stop the video feed

`video_flux()`

Description: This function starts the video flux

Note: Normally this function is not used interactivelly

`watch_dog(expT)`

Description: This function starts the watch dog. Essentially, it launch an infinite loop that follows every image creation in the working path directory.

Note: This function can be used interactivaly, but will block the main thread. Either use this function with the acq function or open another console the execute the stop_acq function when you want the acquisition to stop.

Return: void

CHAPTER

ONE

HOW TO USE PYTHON_ANDOR.PY

1.1 Getting Started

Getting the window comupter ready

First, connect to the window desktop that is installed on the telescope with anydesk

Fig. 1: Anydesk icon.

Then enter the number: 432 587 514. The password is Cinnamon&Nutmeg. Then, launch Audela.

Fig. 2: Audela icon.

In the console (see image), start the server script

```
source server.tcl
```

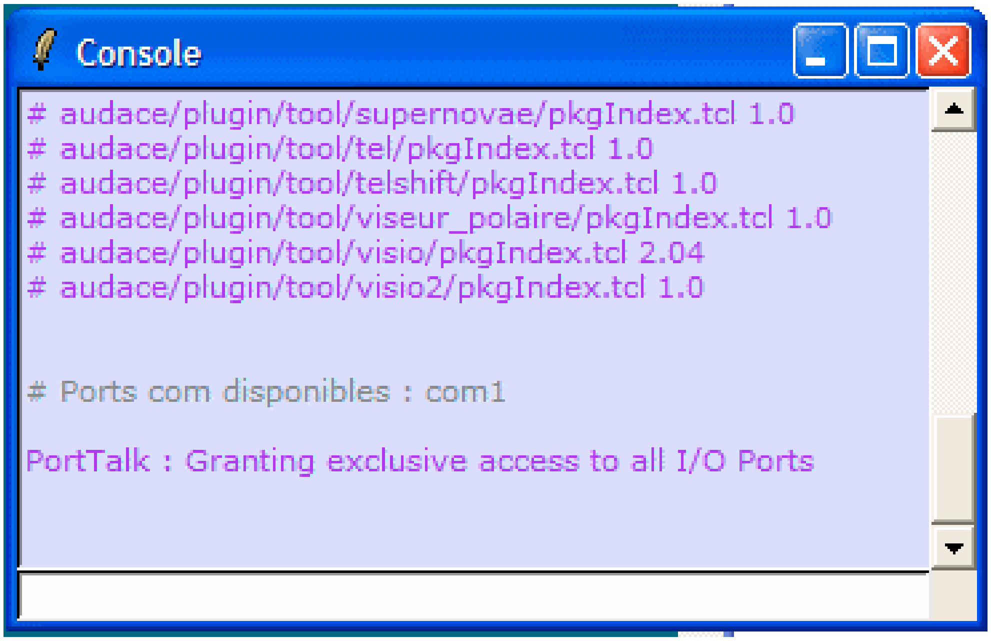
** On the Linux machine (probably Lyra)**

First open a terminal (ctrl+alt+t in Ubuntu) and open a python console (i.e., >>> python). Then enter:

```
from python_andor import command as com

com = com()
```

The camera should now be initialized and ready to be used.



```
# audace/plugin/tool/supernovae/pkgIndex.tcl 1.0
# audace/plugin/tool/tel/pkgIndex.tcl 1.0
# audace/plugin/tool/telshift/pkgIndex.tcl 1.0
# audace/plugin/tool/viseur_polaire/pkgIndex.tcl 1.0
# audace/plugin/tool/visio/pkgIndex.tcl 2.04
# audace/plugin/tool/visio2/pkgIndex.tcl 1.0


# Ports com disponibles : com1

PortTalk : Granting exclusive access to all I/O Ports
```

Fig. 3: Audela consol.

1.2 Practical example

** To launch a live feed of the camera**

In the same python console where the command class has been initialized, enter:

```
com.video()
```

You will be prompt to enter the exposure time. Do it, and a live stream should start after (exposure time)+2 seconds.

** To launch a script **

In the same python console, enter:

```
com.script()
```

You will then be prompted to enter the type of exposure [Target,flat,video or dark], the exposure time and the name of the object. Do it, and a live stream should start after (exposure time)+2 seconds.

CHAPTER

TWO

FILTER WHEEL

To change the filter simply enter in the python console:

```
com.fw(position)
```

Where the position can be ‘open’, ‘g’, ‘r’, ‘i’, ‘z’, ‘Ha’

CHAPTER

THREE

LIMITATION

This is an interface to operate Pesto that is severly limited. Hopefully, these limitations will not perturbe the observations.

3.1 Exposure Time

The exposure time is currently limited to 1 second and more.

3.2 Acquisition Mode

Currently, only the conventional mode is supported

CHAPTER

FOUR

PERFORMANCES

The camera has been quickly caracterized.

	Saturation	Gain	readout noise
Conv	13,000 adu	2.4e/adu	
EM			

Table: performance table

CHAPTER FIVE

INDICES AND TABLES

- genindex
- modindex
- search