Characterization of the OmegaCAM CCDs with the ESO test bench

Fabrice Christen

November 12, 2003

Contents

List of Figures

List of Tables

Chapter 1

Procedure to test ccd with the testbench

1.1 The complete procedure

1. Reception of the CCD
2. Check if Datasheet + flatness measurement are with the CCD
3. Choose a nickname for the chip and include it in the database

   The database corresponds to two files saved in the OmegaCAM-webSite directory. Edit and insert the information in Summary.html and ListCCDs'Nane.html. These two files can be found in the Names'CCD subdirectory see Fig. 1.1. A subdirectory, which has the nickname of the chip, has to be created in the subdirectory EngineeringGrade or Science-Grade, the subdirectory depends of the grade of the chip.
4. Put sticker with the nickname of the chip on the CCD box.
5. Scan datasheet of the CCD and copy the files in the proper directory. See Fig. 1.1. Don't forget to modify the MarconiTestReport.html file also.
6. Clean the box and store it in the Clean Room. The acces code of this room can be obtained from Sebastian Dieries.
7. When the number of CCDs is enough for a new setup send to Sebastian an e-mail to ask him to prepare a new set-up. Don't forget to precise the position of your CCD in the D-Marc setup.
   !!!BE CAREFULL THAT'S VERY IMPORTANT NOT TO MIXE THE CCD FOR THE CONTINUATION OF THE TEST!!!
8. When the set-up is ready, install the D-Marc setup and make the acquisition data (see the description of the procedure page 10).
9. Check data to see if all the data taken are all right.

   If yes, shutdown the test bench (see the procedure "How to shutdown the test bench").

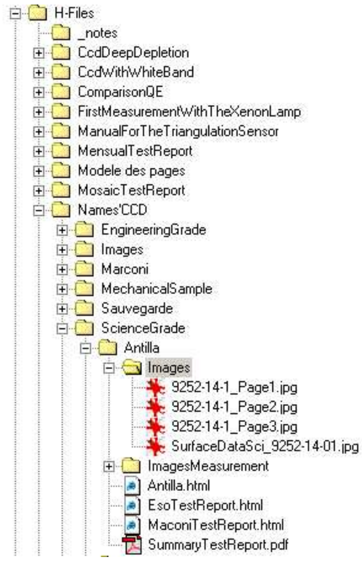

Figure 1.1: Path where to save the scan images in the OmagaCAM-website directory.

If no, repeat the wrong subsequence after anderstanding what was going wrong. Then repeat item 9.

10. One the acquisition data are done, reduce data and prepare web pages (see procedure ”How to reduce data”).

11. Transfer a copy of the modify and new pages on the OmegaCAM account via ftp.

    With a ftp software, create a new connection with the following information:

    Host name: opus6a

    user name: omegacam

    Password: Ask Sebastian Dieries to have the password.

12. Backup the data on CD and DVD.

13. CCD meeting with the responsibles of the chips to rule on the CCDs tested.

    If CCD accepted, keep it in the cleanroom until its final installation.

    If CCD rejected, prepare the CCD to send it back to E2V.

1.2 How to use the test bench

In this section the procedure to install the D-Marc setup and acquire data is described.

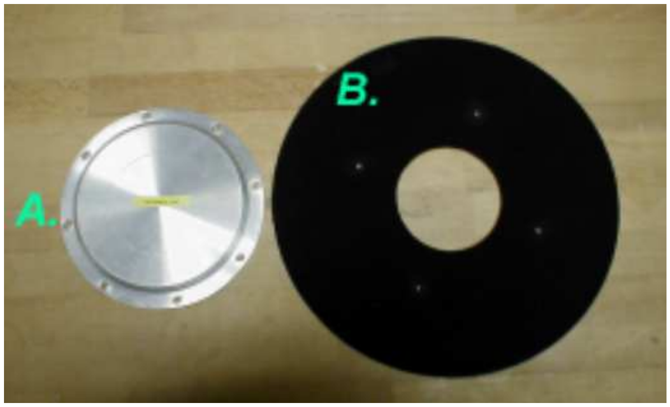

Figure 1.2: Metallic cover for the dark acquisition (A.) and the black baffle for the light acquisition (B.).

1. Install the D-Marc 1 or 2 head, see the "How to install the D-Marc 1 and 2 head on the bench" procedure page 12.
2. Connect the nitrogen bottle, see the "How to connect the nitrogen container before starting a new set-up" procedure, page 16.
3. Start the CFC controller, see the "How to start the CFC controller" procedure page 20.
4. Start FIERA, see the "How to turn on FIERA" procedure, page 23.
5. Prepare the PC for the data acquisition, see the "How to prepare the PC for a new set-up" procedure, page 23.
6. Modify the volttable.def file, see the "How to modify the volttable.def file" procedure, page 25.
7. Start the processes 'RTDServer', 'fcdNoVltSrv' and the 'fcdSlcuCon' panel under the SPARC station, see the "How to use test bench manager batch file" procedure, page 26, to start the processes.
8. With fcdSlcuCon start up the system and put it online, see the "How to use fcdSlcuCon" procedure, page 28.
9. On the PC tou have to prepare the directories on the hard disk where data will be saved 23.

Acquisition of dark and bias exposure.

10. Install the metallic cover in front of the CCDs (See Fig. 1.2).
11. Start the PRiSM software, see the "How to use PRiSM for the dark and bias acquisition" procedure page 29.

After the dark and bias acquisition, the light acquisition procedure can start.

Figure 1.3: The monochromator shutter controller. Power button of the shutter (A.) and the open/close button of the shutter (B.)

10. Install the black baffle in front of the CCDs (See Fig. 1.2).
11. Turn on the halogen lamp, see the ”How to use the halogen lamp” procedure page 30.
12. Turn on the two monochromators (Just press the power button).
13. Turn on the ammeter Ke486 (Just press the power button).
14. Open the monochromator shutter (See Fig. 1.3).
15. Start the process 'fcdtserver'. See the ”How to use test bench manager batch file” procedure, page 26.
16. Check if the SESO shutter is on the position remote, shutter close (see figure 1.26 page 35).
17. Start the PRiSM software, see the ”How to use PRiSM for light acquisition” procedure page 29.

During the process, check regularly the nitrogen level. If the bottle is empty see the ”How to exchange a nitrogen bottle during a run” procedure.

1.2.1 Procedure ”How to install the D-Marc 1 and 2 head on the bench”.

This procedure is identical for the two D-Marc head.

Tools

1. 5 mm PB key.
2. 4 screws 5 mm.
3. Screw driver.

Figure 1.4: First screw to use to attach the camera on its support.

4. 3 mm PB key.
5. 1 screw 3 mm.
6. Joint for the pump tube.
7. Fixation ring for the pump tube.
8. Nitrogen tube.

Procedure

1. Fix the head on the bench support. Always start with the screw on top (see Fig. 1.4 Point A.).
2. Connect the nitrogen tube (see Fig. 1.5 point .A).
3. Connect the nitrogen valve (see Fig. 1.5 point B.) and srew it.
4. Connect pump tube (See Fig. 1.6 point A.) with jointt and fixation ring.
5. Connect pressure sensor (see Fig. 1.6 point B.) and screw it.
6. Connect CFC controller cable (See Fig. 1.6 point C.).
7. Connect Pulpo cable (see Fig. 1.5 point .C).
8. Remove the three ground connectors on D-Marc head (see Fig. 1.7).
9. Connect clock cable (see Fig. 1.8 point .B).
10. Connect bias cable (see Fig. 1.9 point .A).
11. Connect video cable (see Fig. 1.8 point .A).

Figure 1.5: Nitrogen tube (.A), nitrogen valve (B.) and pulpo cable (.C) connected to the D-Marc head.

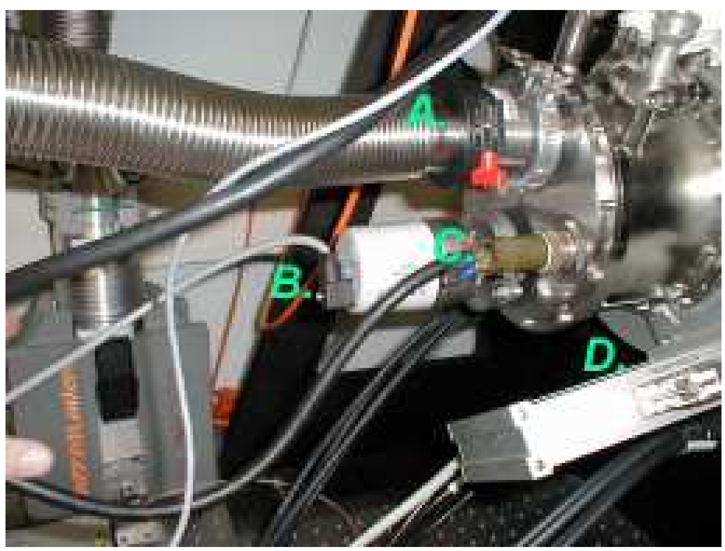

Figure 1.6: Pump tube (A.) connected. CFC controller cables (B. and C.) connected. (B.) pressure sensor cable, (C.) temperature sensor cable. The preamplifier box (D.) fixed on D-Marc head.

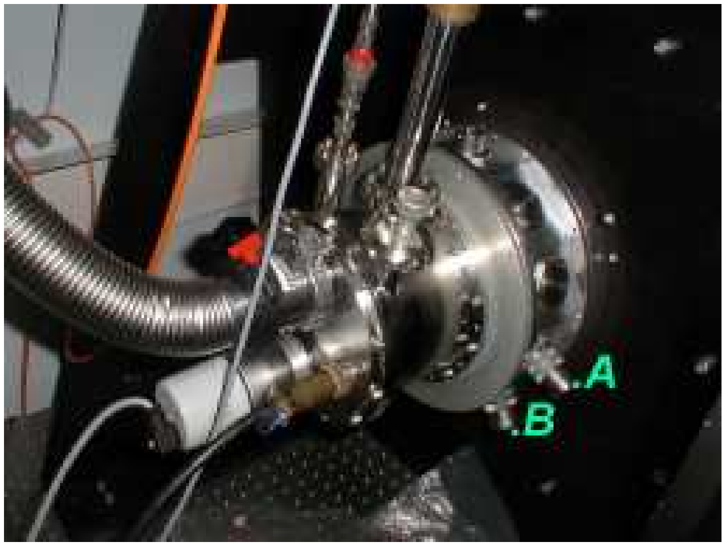

Figure 1.7: Ground connector, the third one is hidden

Figure 1.8: Clock and video connection on D-Marc head.

Figure 1.9: Bias connection on D-Marc head.

12. Fix the pre-amp box on D-Marc head with 1 screw 3 mm and the PB 3 mm key (See Fig. 1.6 point D.).
13. Protect against condensating air humidity the pre-amp box with a plastic bag.

Comments !!! BE CAREFULL !!!

- Don't swap the nitrogen tube on the D-Marc head. This will result in a leak of nitrogen.
- If you don't protect the pre-amp, some water can enter inside and cause a short circuit. That can kill the CCDs!

1.2.2 Procedure "How to connect the nitrogen container before starting a new set-up".

Tools

1. Nitrogen bottle, type: CT 120 R (see Fig. 1.10).
2. Joint for the nitrogen extraction tube.
3. Fixation ring for the nitrogen extraction tube.
4. 13 mm key.

Figure 1.10: The nitrogen bottle

5. Nitrogen extraction tube.
6. Nitrogen tube.
7. Protective gloves.
8. Air gun.

Procedure

1. Put on the protective gloves.
2. Put the jointt for the nitrogen extraction tube on the nitrogen container aperture hole.
3. On the nitrogen extraction tube open the valve #1 (see Fig. 1.11 point .A).
4. Insert nitrogen extraction tube into the LN_2 container.
5. Install the fixation ring and screw it with the 13 mm key.
6. Connect the nitrogen tube.
7. Close the valve.

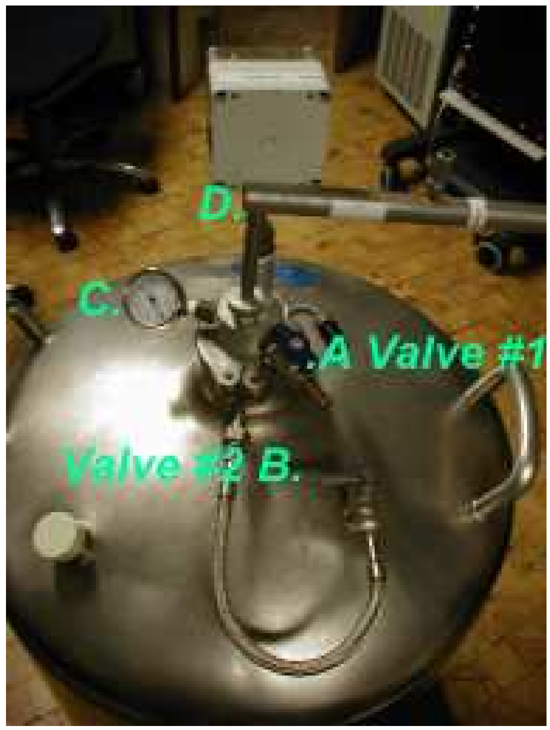

Figure 1.11: The two valves (.A and B.), valve #1 open and valve #2 close, the pressure jauge (C.) and the extraction tube (D.) on the nitrogen bottle.

Figure 1.12: Good position of the nitrogen tube, take the valve on the tube as mark.

Comments !!! BE CAREFULL !!!

- Don't introduce the nitrogen extraction tube too quickly into the nitrogen bottle, otherwise the nitrogen may boil over and cause an injury.
- When you install the nitrogen extraction tube, make sure that the valve is open.
- Don't swap the nitrogen tube. this will cause a leak of nitrogen (see Fig. 1.12).
- The valve #2 (See Fig. 1.11 point B.) can be used to increase quickly the pressure inside the bottle BUT take care that the pressure doesn't increase too much (not more than 0.5 bar on the pressure jauge, see Fig. 1.11 point C.) and NEVER leave the valve open without supervision (the equipment can be damaged).
- If the bottom of the nitrogen bottle is running with moisture, ask the responsible to regenerate it.

1.2.3 Procedure "How to exchange a nitrogen bottle during a run".

Procedure

1. Open valve #1 (See Fig. 1.11 point .A).
2. Wait until there is no pressure inside the bottle anymore (See Fig. 1.11 point C.).
3. Unscrew the nitrogen tube (See Fig. 1.12).

Figure 1.13: The CFC controller (A.) and the SESO power supply (B.).

4. Unscrew the fixation ring.
5. Put on the protective gloves
6. Remove the nitrogen extraction tube (See Fig. 1.11 point D.).
7. Take the joint off the old bottle and put it on the new one.
8. Follow items 2 to 6 of the ”How to connect the bottle before starting a new set-up” procedure.

Comments !!! BE CAREFULL !!!

- Always open the valve #1 (See Fig. 1.11 point .A) even if the pressure sensor (See Fig. 1.11 point C.) shows 0 bar. You can have a residual pressure inside the bottle. If you unscrew the fixation ring with pressure inside you can be injured by the rest of cryogenic gas.
- Use gloves, the icy tube can burn your hands.

1.2.4 Procedure ”How to start the CFC controller”.

The CFC controller permits to control the pump and the temperature of the cold plate in the cryostat (See Fig. 1.13 point A.).

Procedure

1. Put the start button on stand-by (See Fig. 1.14 point A.).
2. Connect the 220V connector.

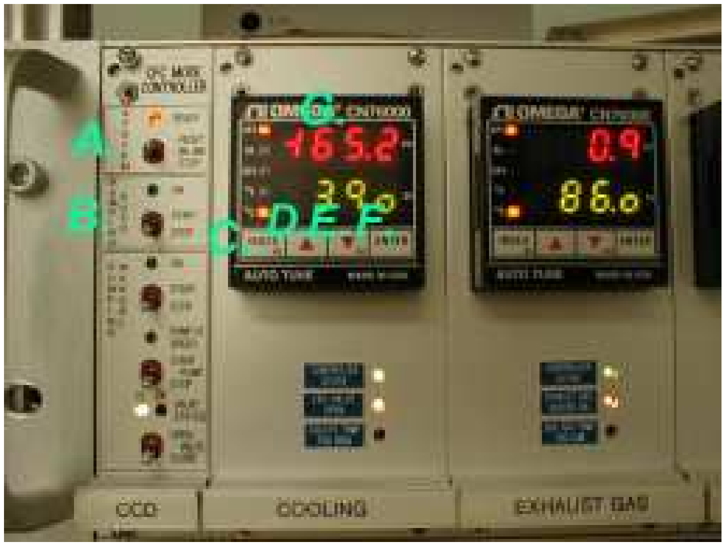

Figure 1.14: The first part of the CFC controller panel. the Start button (A.), the pumping auto button (B.), the index button of the cooling control panel (C.), un and down button (D. and E.), Enter button (F.) and the display (G.).

3. Set the set point of the cooling to -165 deg. Celsius.

    Press the 'Index' button (See Fig. 1.14 point C.).

    With the up and down button, adjust the temperature's set point at -165 deg. Celcius (See Fig. 1.14 point D. and E.).

    Press 'Enter' (See Fig. 1.14 point F.).

    Press 'Index'.

4. Open the manual valve on the pump (See Fig. 1.15 point B.).

5. Open the valve on the D-Marc head.

6. On the CFC controller pump part check is the 'Start' button is on the right (See Fig. 1.16 point B.).

7. Check also on the CFC controler if the 'Stop' button on the right (See Fig. 1.16 point C.).

8. Turn on the pump -0 to 1- (See Fig. 1.16 point A.) .

9. Put the system on-line (See Fig. 1.14 point A.).

10. Set 'Pumping Auto' to On (See Fig. 1.14 point B.).

11. When the automatic valve of the pump opens (you will hear a strong noise from the pump), close the manual (yellow) valve on the pump.

12. Pump until the CCDs are at the operating temperature (around 5 hours).

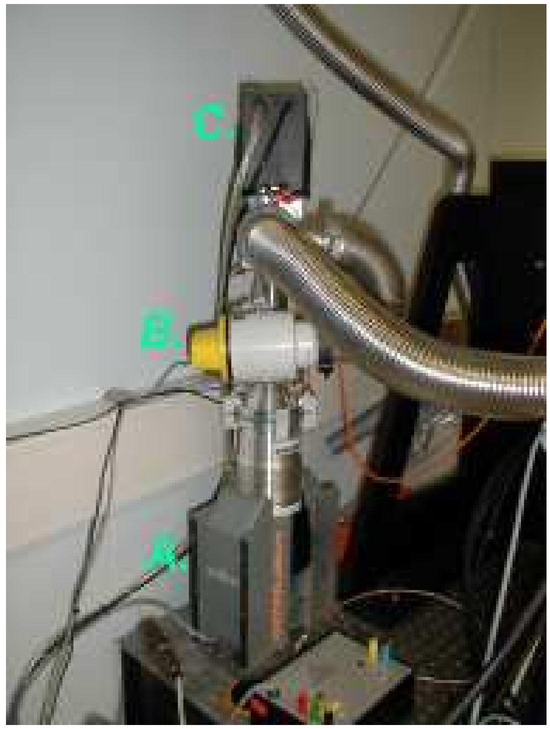

Figure 1.15: The pump and its valves. The pump (A.), the manual valve (B.) and the automatic valve (C.)

Figure 1.16:

13. Set the Hi and Low set point of the pressure to 5.10^{-4} and 5.10^{-6} mbar.

    Press the button D. (see Fig. 1.16) to select hi or low set point and hold it while you press the button E. or F. to adjust the value above.

14. Set 'Pumping Auto' to Off (See Fig. 1.14 point B.).

Comments

- If you feel confident with the system status, after the item 12 you can close the pressure valve on the D-Marc head, set the set point of the pressure, turn off 'Pumping Auto' and turn off the pump.
- If there is a leak, NEVER stop pumping. There can be ice contamination on the chips. Leave the system with 'Pumping Auto' On.
- For more details about the CFC controller, see the document #VLT-MAN-ESO-17130-2345 written by S. Moureau.

1.2.5 Procedure "How to turn on FIERA".

Procedure To start FIERA the SPARC station must be on. Pulpo will read its config file saved on it.

1. Turn on the power supply of the SESO shutter (See Fig. 1.13 point B.).
2. Turn on the cooling device of FIERA (See Fig. 1.17 point .A).
3. Turn on FIERA (See Fig. 1.18 point B. behind bottom left is the swith).

Comments .

- If you don't turn on the cooling system, the temperature inside fiera will increase until a protection limit. FIERA will shut down automatically. To start it up again, you have to wait until the temperature inside FIERA is below the limit. FIEAR will not restart automatically. When this condition concerning the temperature is full fill, shut down FIERA and turn it on again.
- Check occasionally if there is enough water inside the cooling system (See Fig. 1.17 point B.).

1.2.6 Procedure "How to prepare the PC for a new set-up"

Procedure

1. Turn on the PC
2. Copy the 'Model' directory to the Directory 'root':\ProjectName\
3. Rename the directory 'Model' you've just copied to 'yymmdd - A CCD-Name#1 - B CCDName#2'
4. Copy the 'Model' directory to the Directory 'root':\ProjectName\

Figure 1.17: The Fiera cooler

Figure 1.18: Fiera part A. and B.

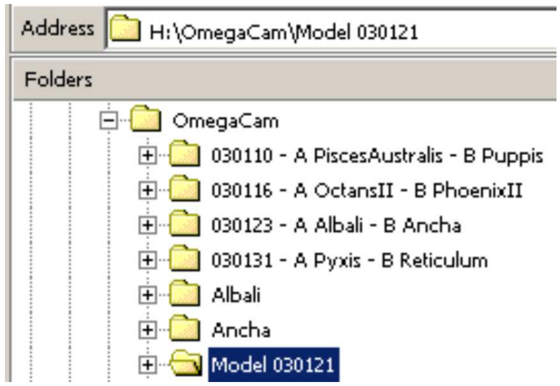

Figure 1.19: Example to use the Model directory.

5. Rename the directory 'Model' you've just copied to 'CCDName#1'
6. Copy the 'Model' directory to the Directory 'roo:\ProjectName\
7. Rename the Directory 'Model' you've just copied to 'CCDName#2'

See the example Fig. 1.19 for the CCD Albali and Ancha. The three directories concerning these two chips have been created with the directory \Model 030121.

Comments

- When you enter the information, use exactly this syntax. Otherwise the scripts (DarkAndBiasAcquisition.pgm and LightAcquisition.pgm) will crash.

1.2.7 Procedure "How to modify the voltable.def file"

Procedure For each chip:

1. Open the 'TemplateEsoTestReport.xls' file in the 'root':\OmegaCAM\ CCDName#i directory (i= 1, 2).
2. In the datasheet of the CCD from the manufacturer, take the chanel potential values and copy them to the 'TemplateEsoTestReport.xls' file, page 'data', section 'Channel potential'.
3. The values computed by excel in this section have to be copied to the volttable.def file in the SPARC station.
4. On the SPARC station type :
   cd $INS_ROOT/SYSTEM/COMMON/CONFIGFILES/$CCDNAME
5. Edit file 'volttable.def' with EMACS and copy the values from TemplateEsoTestReport.xls to this file. Don't swap the section in the 'Volttable.def' file. Biasbrd 0 corresponds to CCD#1 and BiasBrd 1 to CCD#2.

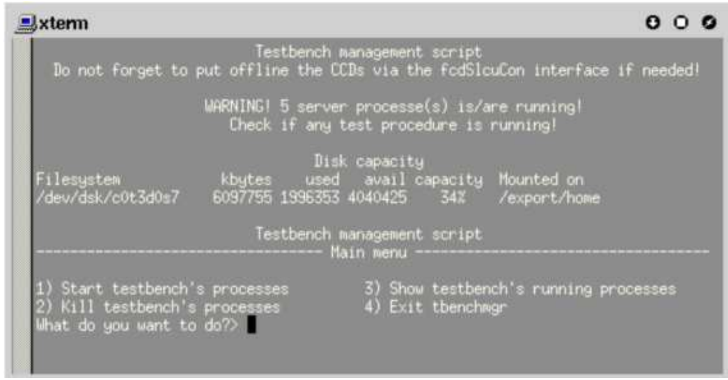

Figure 1.20: The tbenchmgr panel to control the processes.

In the TempleteEsoTestReport.xls file	In the Volttable.def file
VRD	#RDR, #RDL
VOD	#ODR, #ODL
JDR, JDL	#JDR, #JDL

Table 1.1: Variable correspondance between the TempleteEsoTestReport.xls file and the Volttable.def file

6. Save the 'volttable.def'

Comments !!!BE CAREFULL!!!

- Wrong voltages can produce bad results or, worst, damage the chip.

1.2.8 Procedure "How to use test bench manager batch file".

prerequisite step

For the dark acquisition:

- Check before that FIERA and the power supply of the shutter are on.

For the light acquisition:

- Check before that FIERA, the power supply of the shutter, the two monochromators, the ammeter Ke486 and the halogen lamp are on.

Procedure

1. On the Sparc station, start an Xterm session.
2. Type: tbenchmgr ↩ (See Fig. 1.20).

For the dark acquisition:

3. Select Kill test bench's processes.

4. Select "All server processes" to kill all server processes.
5. Select "All interface processes" to kill all interface processes.
6. Return to "Main menu".
7. Select line 1, "Start test bench processes".
8. Select fcdNoVltSrv.
9. Choose the verbosity level.
10. Return to menu "Start testbench process".
11. Start rtdServer.
12. Start fcdSlcuCon.
13. Return to the "Main menu".

For the light acquisition:

3. Select Kill test bench processes.
4. Select "All server processes" to kill all server processes.
5. Select "All interface processes" to kill all interface processes.
6. Return to "Main menu".
7. Select line 1, "Start test bench processes".
8. Select fcdNoVltSrv.
9. Choose your verbosity level.
10. Return to menu "Start testbench's process".
11. Start rtdServer.
12. Start fcdtserver.
13. Choose your verbose level.
14. Start fcdSlcuCon.
15. Return to "Main menu".

Comments

- When you start fcdtserver, it will take a long time before seeing the prompt of this process (1 or 2 mn).
- For more information see the document : The testbench documentation (revised in June 2002) written by Stephane Darbon.

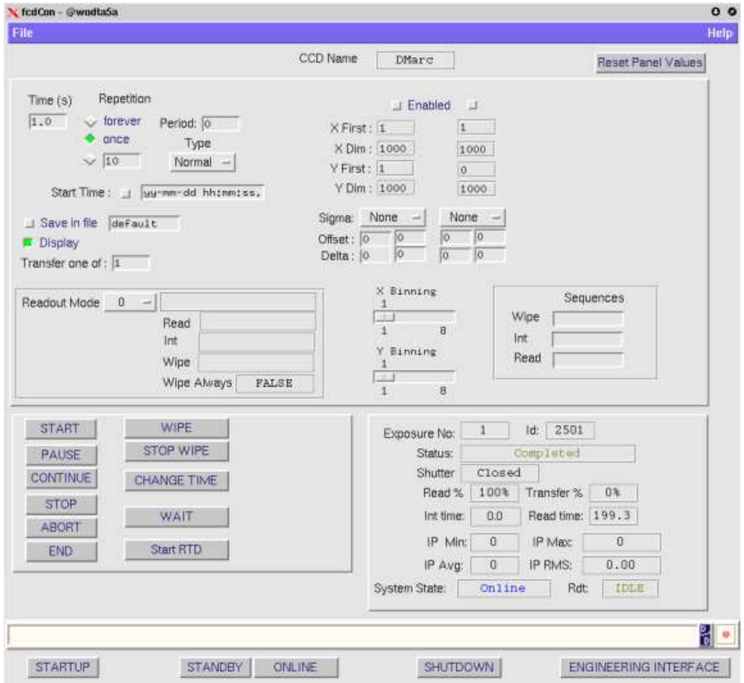

Figure 1.21: The panel fcdScluCon.

1.2.9 Procedure "How to use fcdSlcuCon"

The interface fcdSlcuCon permits you to start the system and control the acquisition of the images.

Procedure With the panel fcdSlcuCon (See Fig. 1.21):

1. Click on the button "STARTUP". Several processes will start to control the system.
2. Click on the button "ONLINE". This command will switch the relays on to connect the CCD and put them online.
3. Wait that the system is online. That can take more than 1 minute.
4. Start the panel RTD by clicking on "StartRTD"
5. In the menu "Real-time" of RTD click on attach camera. Otherwise, the CCD images will not get displayed.

Problems and troubleshoot

- Problem: Bias value around 60 000 ADU.
- Solution: Kill all the processes, Shutdown the sparc, shutdown FIERA, wait five minutes and re-start at the step 4 of the procedure page 1.2.

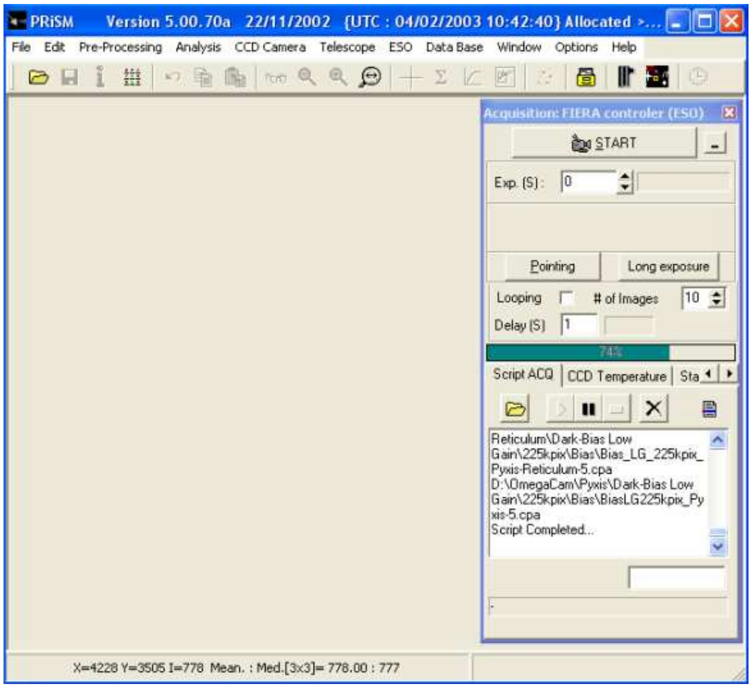

Figure 1.22: The PRiSM software.

1.2.10 Procedure "How to use PRiSM for the dark acquisition"

prerequisite

- PRiSM must be installed and properly setup on the PC, see document The testbench documentation (revised in June 2002).
- Make sure that CCDs are online.

Procedure

On the PC:

1. Start PRiSM (See Fig. 1.22).
2. Open the acquisition panel (Menu CCD Camera – > Acquisition).
3. In the register card 'Script ACQ' of the acquisition panel, open the file: .\yymmdd - A CCDName#1 - B CCDName#2 \ ScriptPRiSM \ Dark-Acquisition.pgm.

1.2.11 Procedure "How to prepare PRiSM for the light acquisition"

prerequisite

- Make sur that the CCDs are online.

- PRiSM must be installed and properly set-up on the PC, see document The testbench documentation (revised in June 2002).

Procedure

1. Start PRiSM (See Fig. 1.22).

2. Open the acquisition panel (Menu, CCD Camera − > Acquisition).

3. Open the monochromator panel (Menu, ESO − > CCD Test-bench − > Monochromator (ms257)).

4. Open the ammeter Ke486 panel (Menu, ESO − > CCD Test-bench − > Ammeter (ke486)).

5. In the register card 'Script ACQ' of the acquisition panel (See Fig. 1.22), open the file:
.\ yymmdd - A CCDName#1 - B CCDName#2 \ ScriptPRiSM \ LightAcquisition.pgm.

Problems and troubleshoot

- P: The acquisition panel, monochromator panel or ammeter panel doesn't want to start.

- S: Check with the testbenchmgr panel (See the section 1.2.8 page 26) if all the processes are running.

1.2.12 Procedure "How to use the halogen lamp"

The halogen lamp. (See Fig. 1.23). All wavelengths between 300 to 1100nm can be obtained without spectral lines. However, the UV output is low.

Procedure to start the halogen lamp
With the control panel (See Fig. 1.24)

1. Press 'POWER ON' (See Fig. 1.24 Point .F).

2. Press 'LAMP OFF/SETUP' for three three seconds (See Fig. 1.24 Point B.).

3. Normally the 'SET MODE' LED is on. If not, press 'METER/ITEM SELECT' until the LED is on (See Fig. 1.24 Point C.).

4. Press 'SET/ENTER' and turn the button 'OUTPUT ADJUST' to select the 'POWER MODE' (See Fig. 1.24 Point E.).

5. Press 'METER/ITEM SELECT' until the 'AMPS' LED is on (See Fig. 1.24 Point C.).

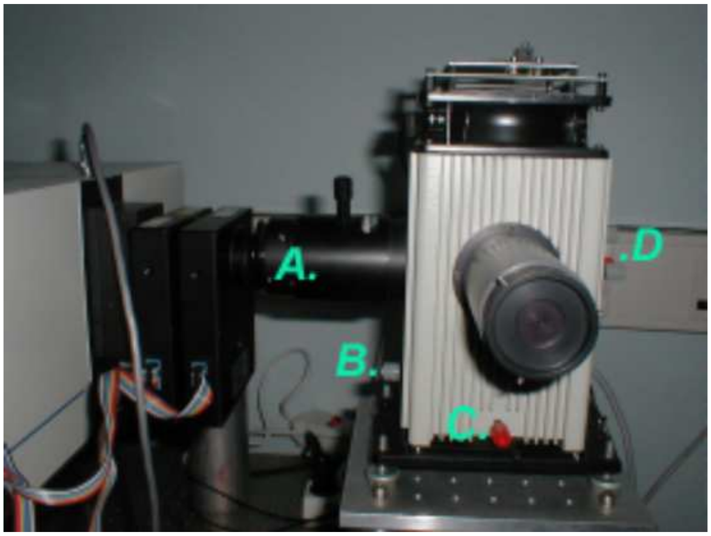

Figure 1.23: The halogen lamp.

Figure 1.24: Control panel of the halogen lamp.

6. Press 'SET/ENTER' and turn the button 'OUTPUT ADJUST' to 8A or more. As the power mode is selected, the lamp will be regulated on the base of watts and not amps. That's the reason why a higher limit is selected for the amps (See Fig. 1.24 Point E. and D.).

7. Press 'METER/ITEM SELECT' until the 'WATTS' LED is on (See Fig. 1.24 Point C.).

8. Press 'SET/ENTER' and turn the button 'OUTPUT ADJUST' to 160W, which is the maximum power that the controller will provide (See Fig. 1.24 Point E. and D.).

9. Press 'LAMP OFF/SELECT' (See Fig. 1.24 Point B.).

10. Press 'SET/ENTER' and turn the button 'OUTPUT ADJUST' to 150W, which is the power specification of the lamp (See Fig. 1.24 Point E. and D.).

11. With 'METER/ITEM SELECT' select 'WATTS' (See Fig. 1.24 Point C.).

12. Press 'LAMP ON' to start the lamp (See Fig. 1.24 Point A.).

13. Wait one hour to have the light stabilized.

Comments

- To check if there is enough light, select the wavelength 630nm with the monochromator control panel under PRiSM, open the monochromator shutter and check on the ammeter ke486 if you have more than 35nA. If not, with focusing ring (See Fig. 1.23 Point A.), and the adjustment screws (See Fig. 1.23 Point B., C, .D and one screw oposite to C.) on the halogen lamp set the beam light to increase the intensity.

Procedure to turn off the lamp

1. Press 'LAMP OFF/SETUP' to turn off the lamp. The controller will stay on stand-by.

2. Press 'POWER OFF' if you really want to shut down the controller.

Problems and troubleshoot:

- Problem: Sometime difficult to set the values or the options with the button 'OUTPUT ADJUST'.

- Solution: Turn the button slowly.

Complementary information:

- Read the Instruction Manual "50-200 WATT He, Hg(Xe), and Hg ARC LAMP POWER SUPPLY MODEL 68806" available in room 54/1 for complementary information.

1.3 How to shutdown the test bench

In this phase the most important thing is that the CCDs MUST never be the coolest point of the camera.

1. Turn off the halogen lamp if it has not been done before (See Fig. 1.24 page 31).
2. Put the CCDs on standby by using the fcdSlcuCon panel. Click on standby (See Fig. 1.21 page 28).
3. Set the temperature set-point of the CCDs at 20C in the PRiSM acquisition panel, register card 'Temperature CCD' (See Fig. 1.22 page 29).
4. Set the cold plate temperature at 20C with the CFC Controler (See Fig. 1.14 page 21 point C., D., E. and F. For more help see section "How to start the CFC Controler" page 20).
5. Open the Nitrogen valve #1 (see Fig. 1.11 point .A page 18) BE CAREFULL!!! Don't stay in front of this output valve when you open it, the outgoing nitrogen under pressure may cause an injury.
6. Wait until the system is at the ambiance temperature by checking the temperature in the acquisition panel, register card 'Temperature CCD' of PRiSM.
7. During this time you can close the monochromator shutter, turn off the monochromator shutter power supply, shutdown the monochromators and the ammeter.
8. Close PRiSM.
9. Put the CCDs offline by using the fcdSlcuCon panel. Click on shutdown (See Fig. 1.21 page 28).
10. Kill all the processes with the tbenchmgr (See Fig. 1.20 page 26).
11. Close the pressure valve on the D-Marc head.
12. Put the CFC Controler on stand-by (See Fig. 1.14 page 21).
13. Turn off the power supply SESO shutter (See Fig. 1.13 point B. page 20).
14. Turn off FIERA (See Fig. 1.13 point B. page 20).
15. Turn off the FIERA cooler (See Fig. 1.17 point .A page 24).
16. Disconnect the 220V plug of the CFC Controler.
17. Disconnect the nitrogen tube from the nitrogen extractor tube (see Fig. 1.12 page 19).
18. Disconnect the nitrogen tube from the D-Marc head (see Fig. 1.5 point .A page 14).
19. Remove the nitrogen valve (see Fig. 1.5 point B. page 14).

20. Disconnect the pressure sensor (See Fig. 1.6 point B.page 14).
21. Disconnect the CFC controller temperature cable from the D-Marc head (See Fig. 1.6 point C. page 14).
22. Disconnect Pulpo cable from the D-Marc head (see Fig. 1.5 point .C page 14).
23. Disconnect pump tube (See Fig. 1.6 point A. page 14)
24. GROUND YOU!!!
25. Disconnect the video cable from the D-Marc head (see Fig. 1.8 point .B page 15).
26. Remove the pre-amp box from the D-Marc head (See Fig. 1.6 point D. page 14).
27. Disconnect the clock and the bias cable from the D-Marc head (see Fig. 1.8 point .B page 15 for the clock connector and Fig. 1.9 point .A page 16 for the bias connector).
28. then screw the ground connectors on the bias, clock and video plug on the D-Marc head (see Fig. 1.7 page 15 point A, B and the last one is hidden beside the D-Marc head).
29. Remove the black baffle (If on the D-Marc head).
30. The D-Marc head is ready to be removed (when you remove the D-Marc head, you have to unscrew the 4 fixation screws. See Fig. 1.4 page 13 unscrew the srew A. last).

1.4 How to calibrate the test bench

Each time that a modification is done on the test bench on the light path, it is important to calibrate it. The most often changes are:

- The halogen lamp change.
- Installation of a D-Marc head with a new window.

prerequisite

- The absolute photodiode (See Fig. 1.25).
- 5 mm PB key.
- 4 screws 5 mm.

1. Install the absolute photo diode instead of the D-Marc head and connect it to the ammeter ke6514 and turn this ammeter on.
2. Prepare the test bench as if a light acquisition has to be done.
3. Open the SESO shutter manually (Select Local mode with the switch B. and open shutter with the switch D. See Fig. 1.26).

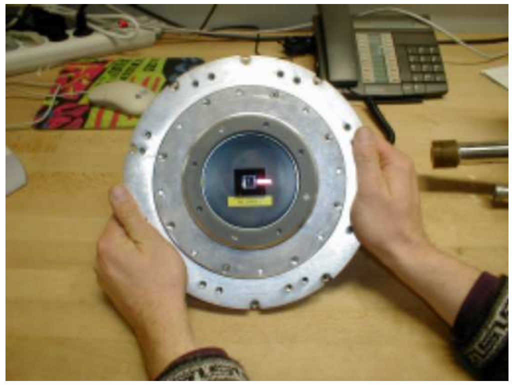

Figure 1.25: The calibrate photodiode.

Figure 1.26: The SESO shutter. A: Assist or logic mode, B: Local or remote mode, C: LED control and D: Open or close shutter.

4. In the register card 'Script ACQ' of the PRiSM's acquisition panel (See Fig. 1.22), open the file:
.\ Model \ ScriptPRiSM \ Calibrate.pgm.

5. Save the results.

6. Create a new 'Calibrate.txt' file by extracting the usefull information in the result file saved before. Use an old 'Calibrate.txt' file to see inside the information you have to extract in the result file.

7. In the directory 'Model', save this file in the same directory of the old 'Calibrate.txt'.

8. the test bench is ready to be used.

Chapter 2

Scripts explanations

2.1 Introduction

Each CCD needs to be characterized to highlight their caracteristics and their defects. The acquisition of these images is separated in two parts. In the first one, all the images which don't need light are taken. This part will be called "Dark and Bias Acquisition". A script has been written to realize the images' acquisition (DarkAndBiasAcquisition.pgm). The second part corresponds to the acquisition of the images which needs light. The name of this second part is "Light Acquisition" and the name of the script is LightAcquisition.pgm. The PRiSM software is used to make all the acquisitions and to automatize the work, the PRiSM script language is used (See Chapter 3). In this chapter the DarkAndBiasAcquisition.pgm and LightAcquisition.pgm scripts are described. This chapter needs to be read as a complement of information to understand or modify these scripts.

2.2 Dark and bias acquisition and reduction

Dark acquisision images are made to determine

- Dark current
- Glowing
- Charge injection
- Hot pixels
- Very bright pixels
- Remanance

To determine these parameters the following images are taken:

- 5 darks 1h 225kpix/s Low Gain Right-Right port
- 5 darks 15min 225kpix/s Low Gain Right-Right port

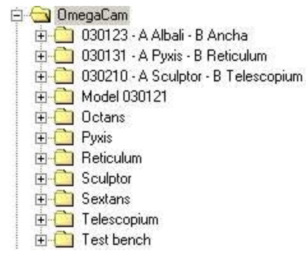

Figure 2.1: Before starting a new data acquisition, Directories need to be created. You have here for example the three directories concerning the CCD Pyxis and Reticulum. These directories 'Pyxis', 'Reticulum' and '030131 - A Pyxis - B Reticulum' are a simple copy of 'Model 030121' directory.

- 5 bias 225kpix/s Low Gain Right-Right port
- 5 darks 1h 225kpix/s High Gain Right-Right port
- 5 darks 15min 225kpix/s High Gain Right-Right port
- 5 bias 225kpix/s High Gain Right-Right port
- 5 darks 1h 50kpix/s High Gain Right-Right port
- 5 darks 15min 50kpix/s High Gain Right-Right port
- 5 bias 50kpix/s High Gain Right-Right port

To make the acquisition of these images the script DarkAndBiasAcquisition.pgm is used. This script possesses two main parts, the acquisition part and the data reduction part.

2.2.1 Dark acquisition

D-Marc 1 and D-Marc 2 setup are two dewars composed of two CCDs. Each taken image is a mosaic of two CCDs. They are saved in the directory created for this purpose (See Fig. 2.1). Based on the list above, a script (DarkAndBiasAcquisition.pgm) has been written to acquire the data. In the following, the first part of this script is described with the help of the diagrams Figure 2.2 and Figure 2.3.

Fig. 2.2 Section I. The diagram starts with the creation of the directories used by the script (See Fig. 2.1).

Fig. 2.2 Section II. To start this script, under PRiSM open the "file" menu, chick on "script", choose the script DarkAcquisition.pgm and click "ok".

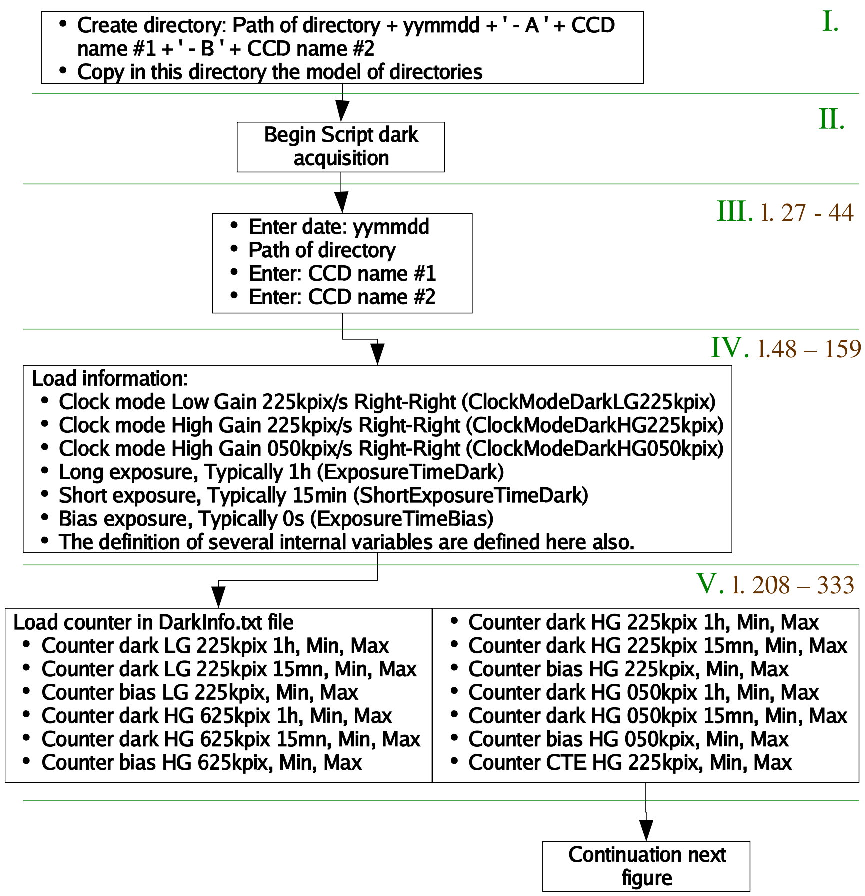

Figure 2.2: First part of the script's diagram 'Dark Acquisition'.

Fig. 2.2 Section III. Four data types need to be entered before the script really starts. This information corresponds to the name of the CCDs (and also the directories' name) where all the images will be saved. Once this information is given no more intervention is needed. You have to give the date, the same date given to create the directory (the date is stored in 'Date$' in the script), the path of the directory (stored in 'Path$') and the two names of the CCDs (are stored in CcdA$ and CcdB$).

Fig. 2.2 Section IV. In the box 'Load information', variables for the different clock modes and exposure times are defined.

- Clock mode 225kpix/s Low Gain Right-Right port, which corresponds to the mode 8 in the fcdSlcuCon panel (ClockModeDarkLG225kpix=8).
- Clock mode 225kpix/s High Gain Right-Right port, which corresponds to the mode 1 (ClockModeDarkHG225kpix=1).
- Clock mode 50kpix/s High Gain Right-Right port, which corresponds to the mode 3 (ClockModeDarkHG050kpix=3).
- The exposure time for the 1 hour darks (in the script the time is defined in milliseconds, ExposureTimeDark=3600000).
- The exposure time for the 15 minutes darks (ShortExposureTimeDark= 900000)
- The exposure time for the bias (ExposureTimeBias=0).

In this section, internal variables like paths of directories or parameters are also specified (See in the script the lines 48 - 117).

Fig. 2.2 Section V. and Fig. 2.3 Section VI. This script uses two other files, DarkInfo.txt and DarkInfoBackup.txt. The content of those two files is:

```
1 1 5 Acquisition dark LG 225kpix 1h
1 1 5 Acquisition dark LG 225kpix 15mn
1 1 5 Acquisition bias LG 225kpix
6 1 5 Acquisition dark HG 625kpix 1h
6 1 5 Acquisition dark HG 625kpix 15mn
6 1 5 Acquisition bias HG 625kpix
1 1 5 Acquisition dark HG 225kpix 1h
1 1 5 Acquisition dark HG 225kpix 15mn
1 1 5 Acquisition bias HG 225kpix
1 1 5 Acquisition dark HG 050kpix 1h
1 1 5 Acquisition dark HG 050kpix 15mn
1 1 5 Acquisition bias HG 050kpix
6 1 5 Acquisition CTE HG 225kpix Fe55
```

These files are identical and are used to save the status in case of the system crashes. If you have to restart the script, these values will be reloaded and the script will restart exactly at the point defined there. The text after the third column is only for comments. Currently the clock mode 625kpix/s High Gain

Right-Right is available in the test bench but it is not validated. No measurements are done with it, that's why one has 6 instead of 1. Once this mode will be ready for scientific purpose, replace 6 by 1 to acquire the data. When the Fe55 setup will be ready for scientific purpose, one replaces 6 by 1 in the last line.

The counter, the starting point and the end point are loaded. See here for example the variables used to load the information of the first line:

- StartNumberDarkLG225kpix1h: contains the first constant.
- BeginNumberDarkLG225kpix1h: contains the second constant.
- NumberDarkLG225kpix1h: contains the third constant.
- InfoDarkLG225kpix1h$: contains the rest of the line which are comments (Acquisition dark LG 225kpix 1h).

In the script DarkAcquisition.txt, lines 248 - 324 contain all the variables used to store the whole table.

The meaning these variables is the following. For each line of DarkInfo.txt file, the script acquires the data until the counter (in this example, the variable StartNumberDarkLG225kpix1h) is higher than the end point (NumberDarkLG225kpix1h). After each image saved on the disk, the counter is incremented and the DarkInfo.txt file updated.

Fig. 2.3 Section VII. Before starting the acquisition, the CCD mounted in the dewar on the test bench, has to be at the operating temperature (-120 Deg). The script sets the temperature at -120 Deg. During the cooling phase, FIERA is initialised to always have the shutter closed, the fastest mode (the mode 7) is set and biases are taken. From those biases (named 'Img'), the temperature is extracted from the header by the function GetTCCD and saved in the variable 'Temperature'. While the temperature is higher than -120 Deg., the script continues taking biases. When the temperature is reached, the program goes to the next section.

This section (read ccd during cooling) is also very important for the remanance. If the CCDs are cooled without a continious reading, strong remanance can be observed on the images. To reduce this effect the CCDs must be read continiously during the cooling phase.

Fig. 2.3 Section VIII. This section is the heart of the program.

For clock modes and exposure times defined at the beginning of the script (section IV), images are taken and saved in the directories created at this purpose. You can see on the Fig. 2.4 an example. It is the structure of the directories and their composition at the end of the acquisition data for the CCDs Pyxis and Reticulum. The similar structure can be found in the directory 'Dark-Bias Low Gain' (HG letters in the names of the images are replaced by LG).

The acquisition part is now done. All the images are saved in the directories. The second part of this script, the data reduction, can start.

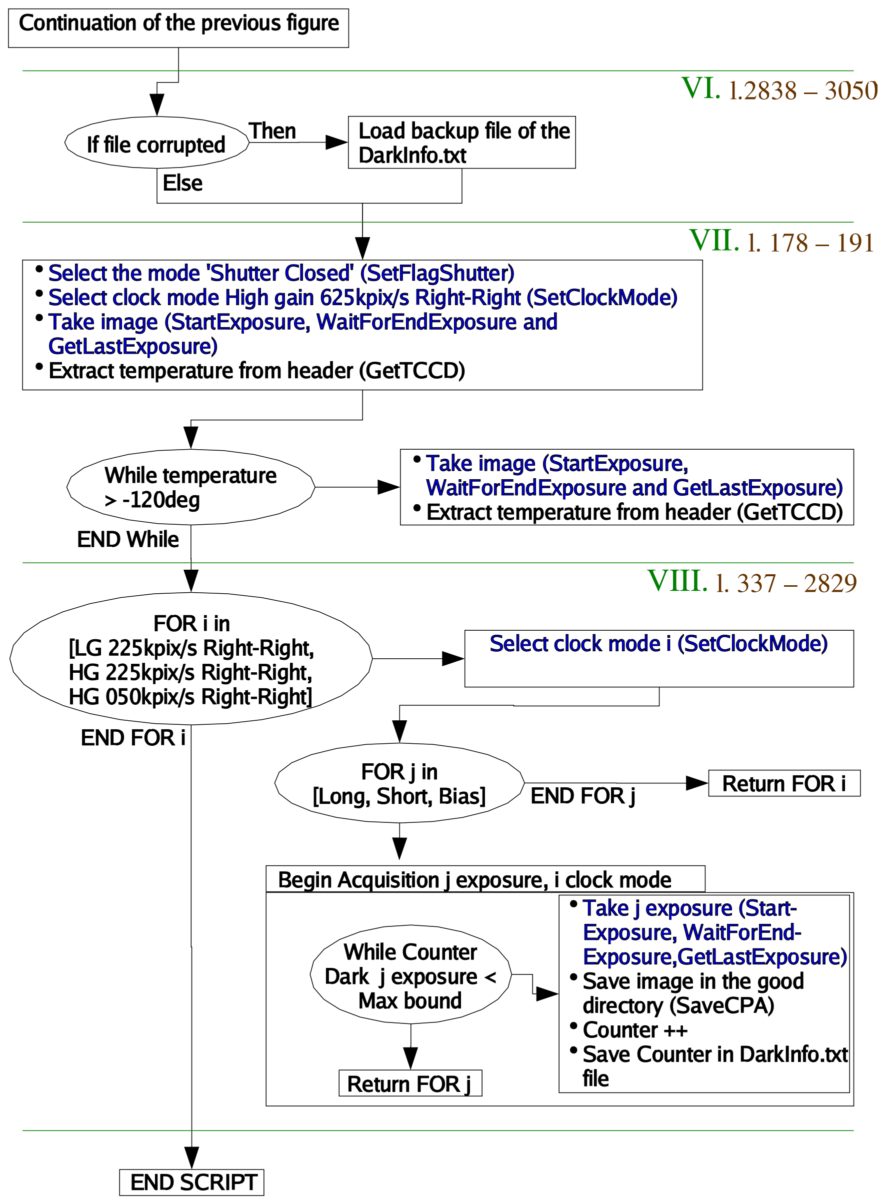

Figure 2.3: Second part of the script's diagram Dark Acquisition.

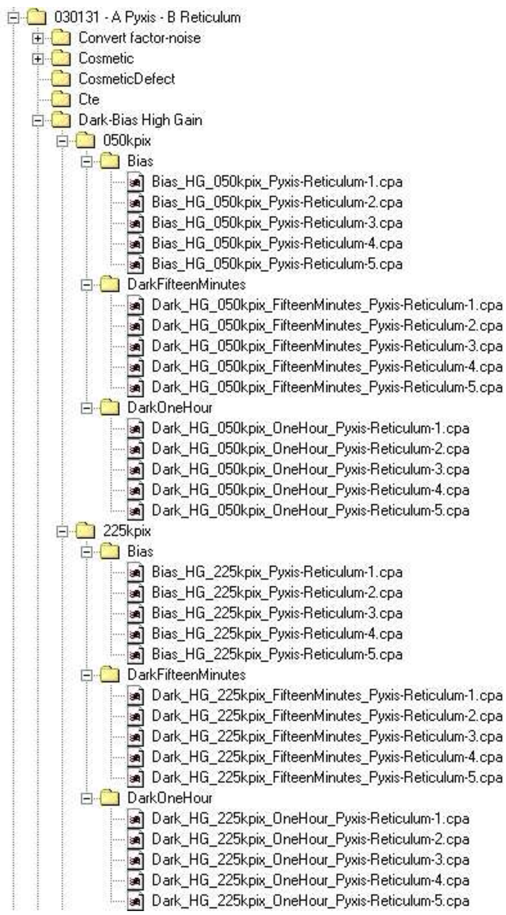

Figure 2.4: Structure and composition of the directory Dark-Bias High Gain after the acquisition of the data.

2.2.2 Data reduction of the dark and bias images

In this section, some explanations are presented for the diagram Fig. 2.5 and Fig. 2.6. This section corresponds to the second part of the script DarkAndBiasAcquisition.pgm.

Fig. 2.5 Section I. In the script, for each mode used to acquire the data, the position of sentitive areas, needs to be specified.

The coordinates of the CCD #1 with the pre and overscan are: X11_WO (=1), Y11_WO (=1), X12_WO (=2148), Y12_WO (=4200).

The coordinates of the CCD #2 with the pre and overscan are: X21_WO (=2149), Y21_WO (=1), X22_WO (=4296), Y22_WO (=4200)

The coordinates of the sensitive area in the cut images are defined (X1 =49 or 50, Y1=1, X2=2096 or 2097, Y2=4102). For X1 and X2 two values are given. It is due to the fact that the sensitive area is not at the same place in function of the mode used.

Also the path where to load the mosaic image and to save the cut images are specified here (Image$, SaveImageA$ and SaveImageB$).

Fig. 2.5 Section II. In this section, the mosaic images made during the acquisition procedure are cut and the results are saved in the proper directories with the parameters defined in the section I. In the example Fig. 2.1 and Fig. 2.4, the images saved in the directories

- \ 030131 - A Pyxis - B Reticulum \ Dark-Bias High Gain \

are cut in two parts and saved in

- \ Pyxis \ Dark-Bias High Gain \

and

- \ Reticulum \ Dark-Bias High Gain \

respectively. The same thing is done for the files in the subdirectory Dark-Bias Low Gain (In Fig. 2.7 the result of the cut-out can be seen for the CCD Sculptor).

Fig. 2.5 Section III. Then, for each CCD, a median computation is done with the 5 images in the subdirectory 'Bias', 'DarkFifteenMinutes' and 'DarkOneHour' (ImA[i], ImB[i] are the tables of images used to compute the master image for the CCD A and B in the script). Based on the median image, an image is saved with the pre and overscan. The CPA format is used and the image is saved in the directories created for this purpose. (The coordinates of the image are defined in the section I and the paths' variables where to save the median image with overscan are SaveMedianWO-A$, SaveMedianWO-B$). Another image with the sensitive area is saved using CPA format in the correspondant directory (see section I for the coordinates and the paths' variables where to save the median image without overscan are SaveMedian-A$, SaveMedian-B$). This image is also saved, using the JPEG format, in the directory HTML-Files. After saving this image, a 15x15 binning is computed and saved using the FITS format in the correspondant directory. This image is also saved under JPEG format in the directory HTML-Files (See Fig. 2.8). An example of the sub directory 'Dark-Bias High Gain' of the CCD Sculptor is shown Fig. 2.7.

2.3 Light acquisition

The light acquisition part is done to determine

- Conversion factor and noise
- Quantum Efficiency
- cosmetics
- Linearity
- Pocket pumping (to identify traps)
- Very large trap
- bad columns

To determine these parameters the following images are taken:

- 2 bias 50kpix/s High Gain Right-Right port 512 lines
- 2 flats 50kpix/s High Gain Right-Right port 512 lines
- 2 bias 225kpix/s High Gain Right-Right port 512 lines
- 2 flats 225kpix/s High Gain Right-Right port 512 lines
- 20 flats 50kpix/s High Gain Right-Right port
- 20 flats 225kpix/s High Gain Right-Right port
- 2 flats 225kpix/s Low Gain Right-Right port 512 lines
- 2 TDI (Transfert During Integration) 225kpix/s High Gain Right-Right port 512 lines
- 2 flats Pocket pumping 225kpix/s High Gain Right-Right port
- 5 bias 225kpix/s High Gain Right-Right port 512 lines 512 columns.
- 92 flats 225kpix/s High Gain Right-Right port 512 lines 512 columns.

To make the acquisition of these images the script LightAcquisition.pgm is used (see diagram Fig. 2.9). This script possesses five main parts:

- Conversion factor and noise
- Cosmetics
- Linearity
- Pocket pumping
- Quantum efficiency

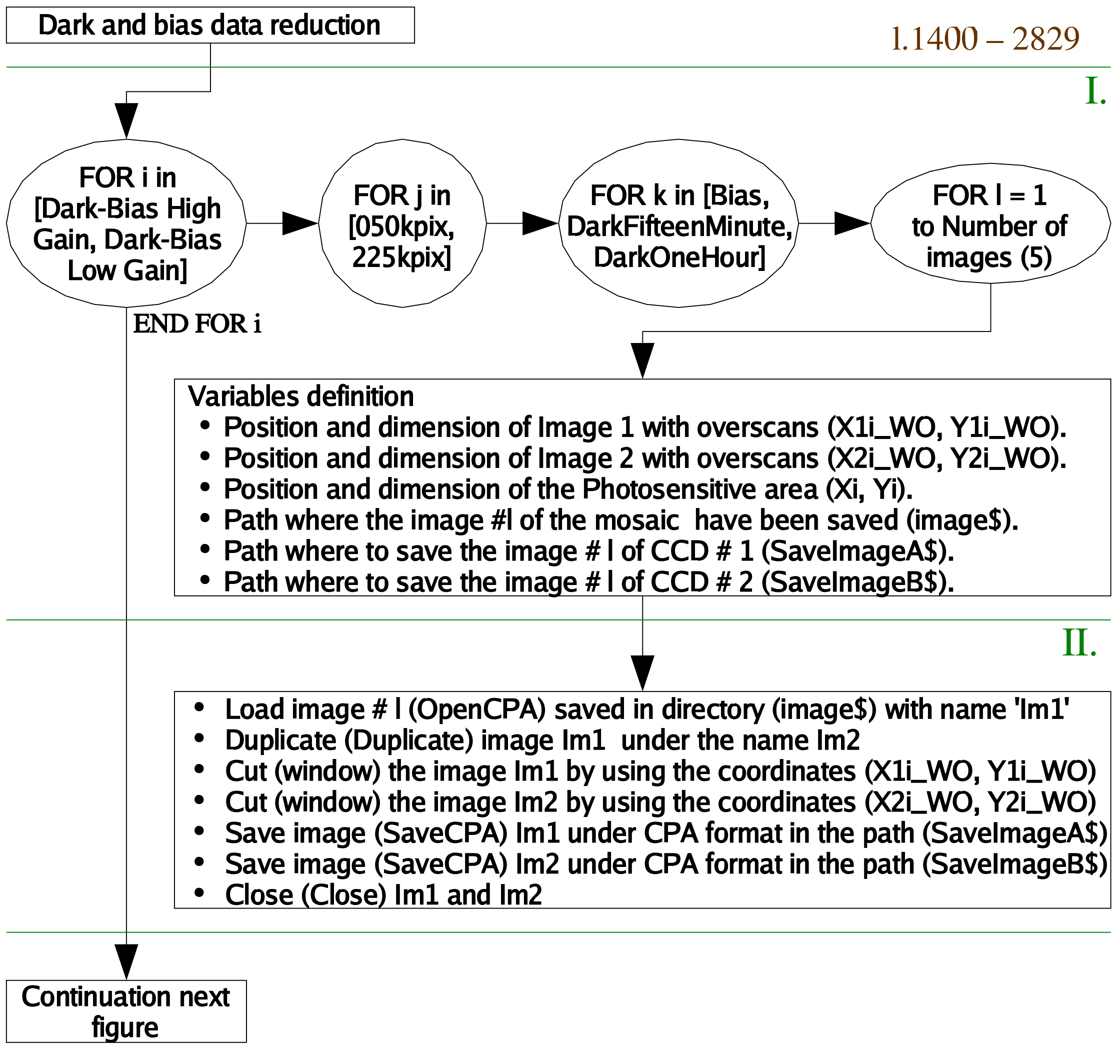

Figure 2.5: First part of the script's diagram : dark and bias reduction.

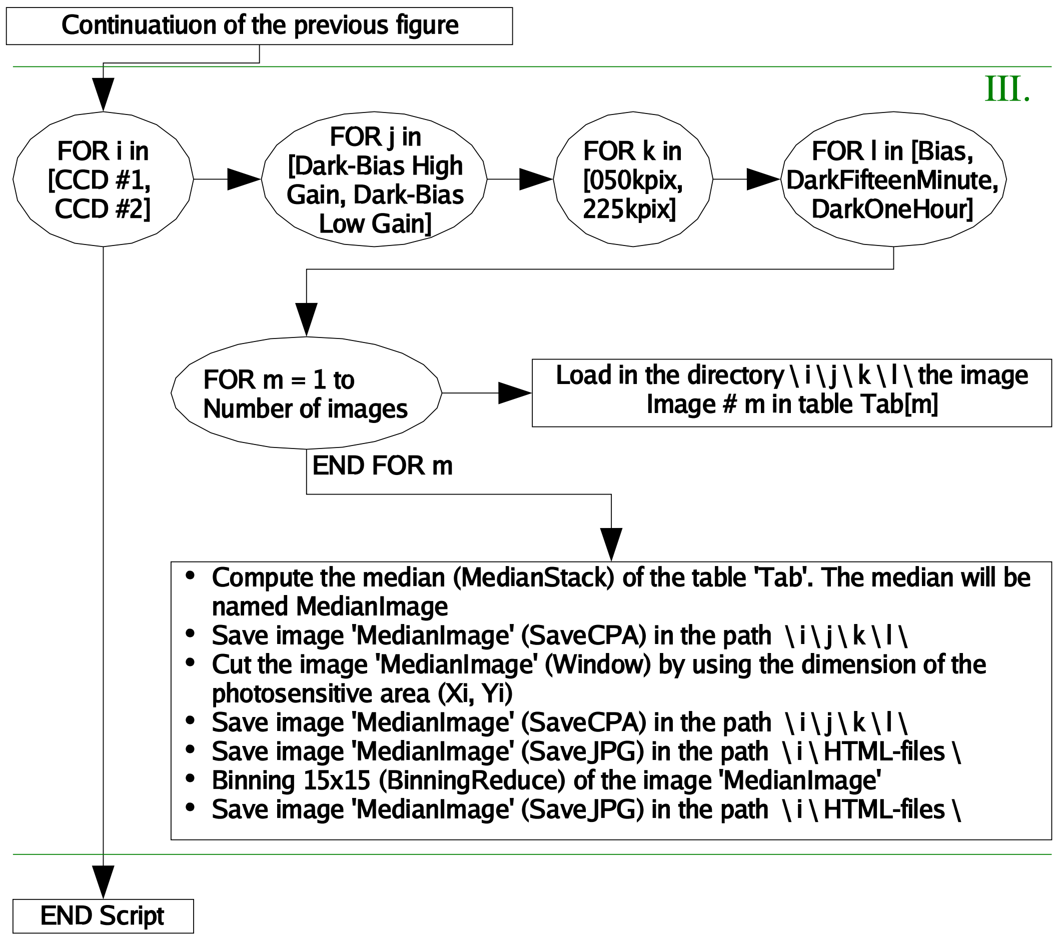

Figure 2.6: Second part of the script's diagram : dark and bias reduction.

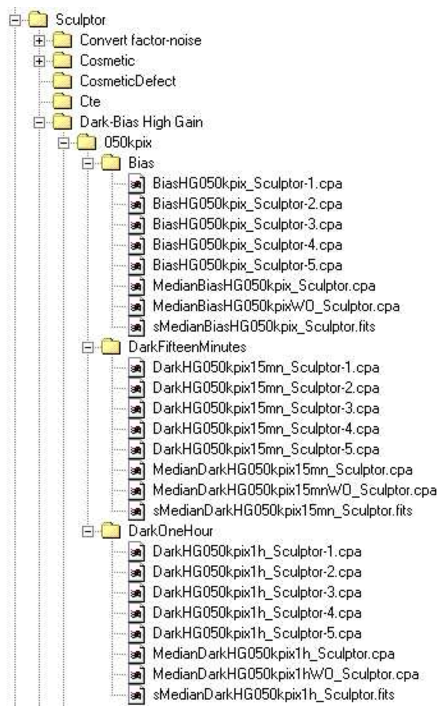

Figure 2.7: Structure of the directories after saving the images.

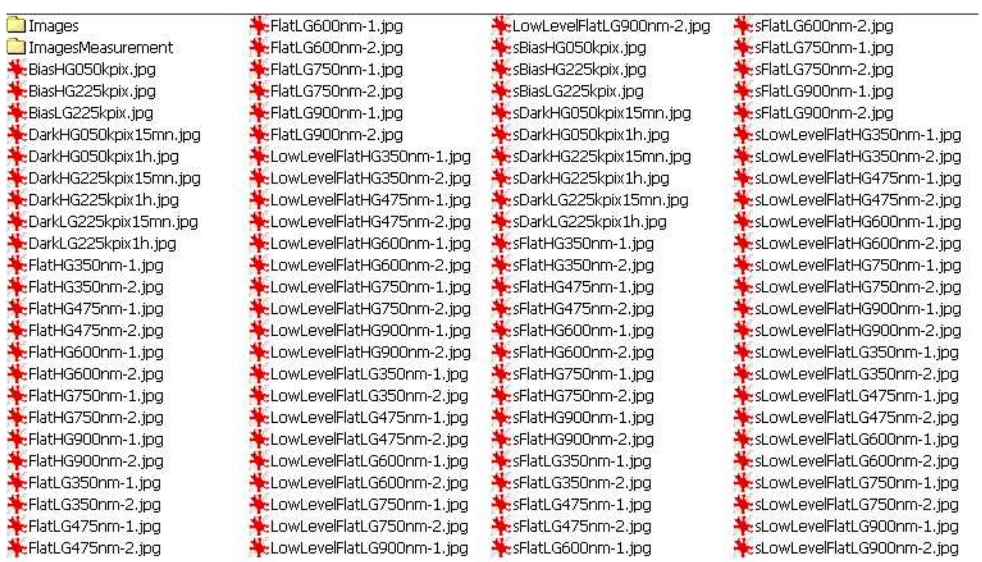

Figure 2.8: Files in the HTML-files directory.

Fig. 2.9 Section I. As for the previous script, it is necessary to check if the directories are ready before starting it. If they are not, they have to be created.

Fig. 2.9 Section II. Four data types need to be entered before the script really starts. This information permits to save the data in the good directories. Once done the script does not need intervention anymore. You have to give the date, the same date given to create the directory (Date$), the path of the directory (variable Path$ in the script) and the two names of the CCDs (CcdA$ and CcdB$).

Some internal variables are also defined here. Variables for directories, clock modes, parameters for functions, filters values, bandwidths, wavelengths, exposure times and temperature can be found between lines 44 and 142 of the LightAcquisition.pgm file.

Fig. 2.9 Section III. Before starting the measurements, the script loads information in the file LightInfo.txt. If this one is corrupted it loads the backup file LightInfoBackup.txt. These files are identicals and are used to save the status in case of the system crashs. If 'statut' equal zero, the measurement concerning this sub section has to be done. If the 'statut' equal 1 the subsection is not executed. See below the content of the file LightInfo.txt and LightInfoBackup.txt.

```
0 Conversion factor and noise 0= To do; 1= Done.
0 Cosmetic 0= To do; 1= Done.
0 Linearity method TDI 0= To do; 1= Done.
0 Pocket pumping 0= To do; 1= Done.
0 Quantum efficiency 0= To do; 1= Done.
```

The first number of each line are loaded and stored in the variables DoneConvertionFactorAndNoise, DoneCosmetic, DoneLinearityTDI, DonePocketPumping and DoneQE respectively. The rest of the lines are only comments.

Fig. 2.9 Section IV. This part is the kernel of the script. Each sub-function, ConvertionFactorAndNoise:, Cosmetic:, LinearityTDI:, PocketPumping:, QE:, are called if the variables defined in the section III are equal to 0.

2.3.1 Conversion Factor and noise

The diagram Fig. 2.10, described the complete sequence to acquire data and to calculate the conversion factor and the read out noise.

Fig. 2.10 Section I. The subscript starts with the initialisation of the testbench.

The shutter is set on position close, the monochromator filter at 100%, the monochromator bandwidth at 7nm and the monochromator wavelength at 630nm.

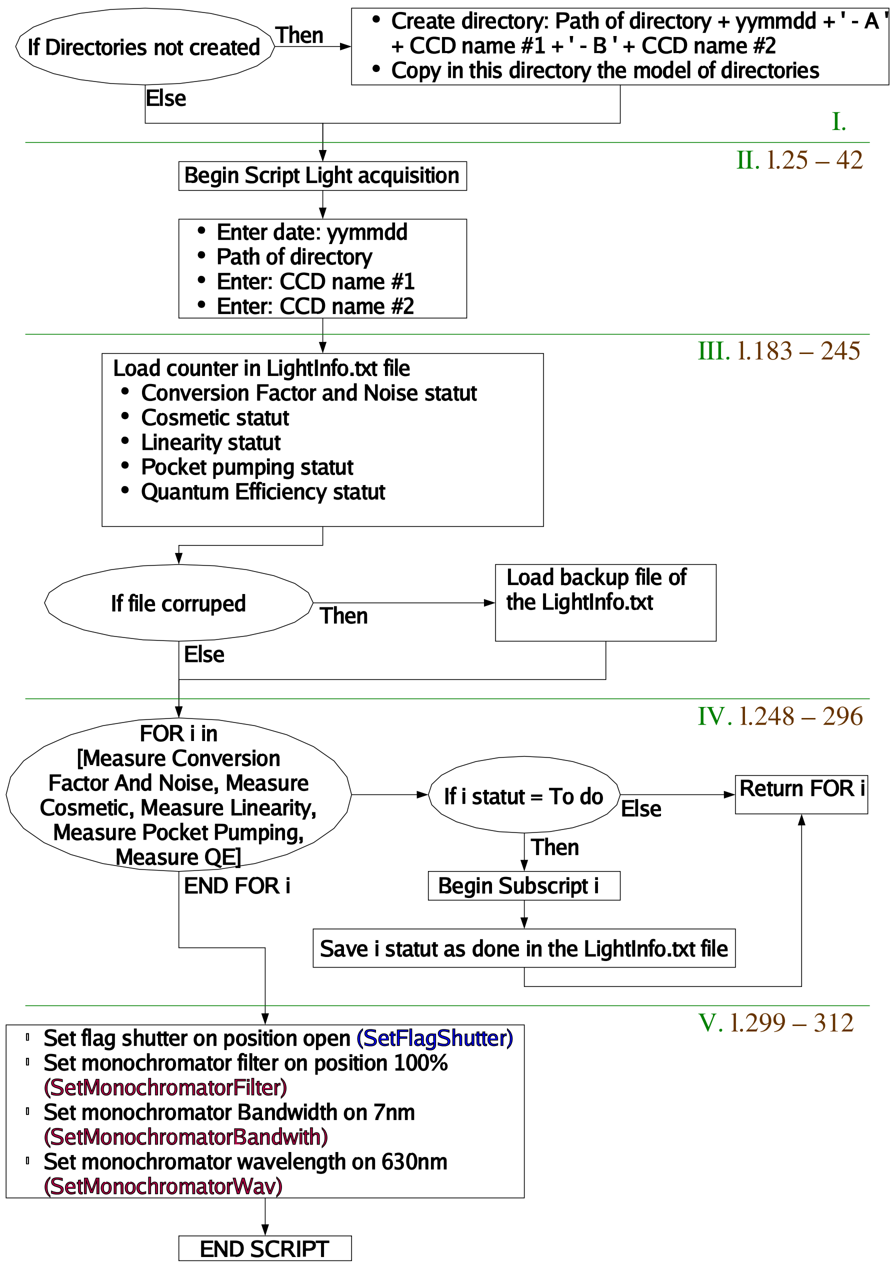

Figure 2.9: Diagram of the script light acquisition.

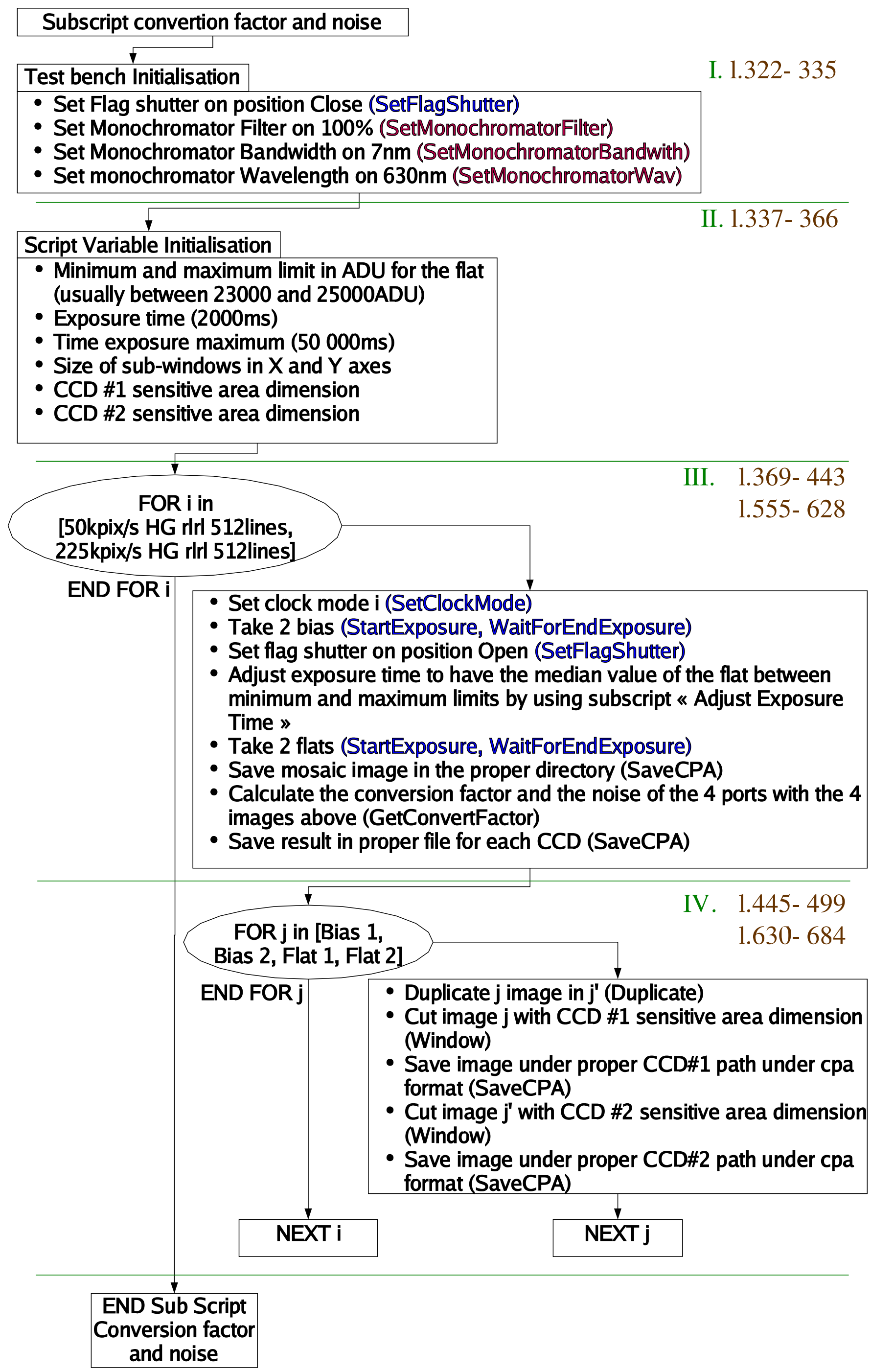

Figure 2.10: Diagram of the subscript, Conversion Factor.

Fig. 2.10 Section II. In this section, the variables used by this sub-function are defined.

To calculate the conversion factor, boundaries for the intensity light need to be set (BorneSup=46000, BorneInf=42000 units in ADU).

The exposure time and the maximum exposure time authorized are initialized (Exposure=2000, TexpMax=50000).

The coordinates of the window where to calculate the conversion factor and the noise and the number of sub-divided windows inside the window are also defined (NbWindowX = 10, NbWindowY = 10, X1 CcdALeftPort = 100, Y1 CcdALeftPort = 10, X2 CcdALeftPort = 1060, Y2 CcdALeftPort = 500, X1 CcdARightPort = 1090, Y1 CcdARightPort = 10, X2 CcdARightPort = 2080, Y2 CcdARightPort = 500, X1 CcdBLeftPort = 2210, Y1 CcdBLeftPort = 10, X2 CcdBLeftPort = 3210, Y2 CcdBLeftPort = 500, X1 CcdBRightPort = 3240, Y1 CcdBRightPort = 10, X2 CcdBRightPort = 4160, Y2 CcdBRightPort = 500).

Fig. 2.10 Section III. Then starts the body of the script. With these lines, the read out noise and the converssion factor of the four channels are calculated for two modes. The 50kpix/s high gain mode and the 225kpix/s high gain mode. Two flats and two bias are taken and saved in the directory 'yymmdd - A CCD#1 - B CCD#2 \ Linearity'. These four images plus the coordinates defined in the section II are used as input of the function GetConvertFactor to calculate the values (the variables ConvertFactorLeftPortCcdA, RMSNoiseLeftPortCcdA, ConvertFactorRightPortCcdA, RMSNoiseRightPortCcdA, ConvertFactorLeftPortCcdB, RMSNoiseLeftPortCcdB, ConvertFactorRightPortCcdB, RMSNoiseRightPortCcdB are used as output of the function GetConvertFactor). These results are saved in the ResultCFandN.txt file in the directories 'CCD#1 \ ConvertFactor-Noise' and 'CCD#2 \ ConvertFactor-Noise'.

Fig. 2.10 Section IV. Then, the mosaic image is cut in two parts and saved in the directories 'CCD#1 \ ConvertFactorNoise' and 'CCD#2 \ ConvertFactorNoise'.

2.3.2 Cosmetic

The diagram Fig. 2.11 explains the subscript Cosmetic. It has been written to acquire 40 differents flats in 2 groups. In the first group, images are taken with the clock mode 50kpix/s, high gain, right right port. In the second, images are taken with the clock mode 225kpix/s, Low gain, right right port. In each group there are two groups. The first one is called high level, which corresponds to flats with a mean value around 48 000 ADU and the second one is called low level, which corresponds to flats with a mean value around 3 400 ADU. In each subgroup 10 images are taken. 2 at 350nm, 2 at 475nm 2 at 600nm, 2 at 750nm and 2 at 900nm. The bandwidth of the monochromator is set to 5nm for each image.

Fig. 2.11 Section I. Initialisation of the test bench.
The monochomator is set to a bandwidth of 5nm and the filter at 100%. The shutter is set to autorize the opening during the exposure time.

Fig. 2.11 Section II. Initialisation of the variables.

To converge, the sub-script needs boundaries for the exposure time (TexpMin=7500, TexpMax=240000) and the light intensity (BorneInf=23000, BorneSup=25000).

A table of integer is created to select the wavelengths (WaveLength[1]= 350, WaveLength[2]= 475, WaveLength[3]= 600, WaveLength[4]= 750, WaveLength[5]= 900 units in nm).

Two tables of integer are created to choose clock modes (ClockMode[1]=3, ClockMode[2]=8, ClockModeFast[1]=4, ClockModeFast[2]=9).

Fig. 2.11 Section III. The monochromator wavelength is set with the values contented in the table WaveLength (see section II). The clock mode is set with the values contented in the table ClockModeFast. Then the exposure time and the filter are ajusted with the subscript 'Adjust exposure time'. Once done the clock mode is set with the parameters which are in the table ClockMode.

Fig. 2.11 Section IV. The flats are taken and saved in the directory 'yymmdd - A CCD#1 - B CCD#2 \ Cosmetic \ 050kpix or 'yymmdd - A CCD#1 - B CCD#2 \ Cosmetic \ 225kpix with specified name (see figure 2.12 for example). This image is cut in two parts and saved in the directory of the CCD#1 \ Cosmetic \ 050kpix or CCD#1 \ Cosmetic \ 225kpix, depends the mode used. The same is done for the CCD #2.

The dimension of the images with the overscans:
the CCD #1: X1=1, Y1=1, X2=2148, Y2=4200
the CCD #2: X1=2149, Y1=1, X2=4296, Y2=4200.

Dimension of the images without the overscans:
the CCD #1: X1=50, Y1=1, X2=2097, Y2=4102 for the clock mode 3
the CCD #1: X1=49, Y1=1, X2=2096, Y2=4102 for the clock mode 8.
the CCD #2: X1=2198, Y1=1, X2=4245, Y2=4102 for the clock mode 3
the CCD #2: X1=2197, Y1=1, X2=4244, Y2=4102 for the clock mode 8.

Fig. 2.11 Section V. The exposure time is reduced (the variable Exposure is multiplied by 0.07) and the flats low level are taken and saved. The constants used in the section IV are taken to select the size of the images with and without overscans.

2.3.3 Linearity

The diagram Fig. 2.13 explains the subscript Linearity.

Fig. 2.13 Section I. Initialisation of the variable used by this sub-function. The sub-script needs boundaries for the intensity and a starting point for the exposure time. The starting point of the exposure time is set to 10000ms (Exposure=10000). The boundaries for the the intensity are 46000 ADU and 50000 ADU (BorneInf=23000, BorneSup=25000).

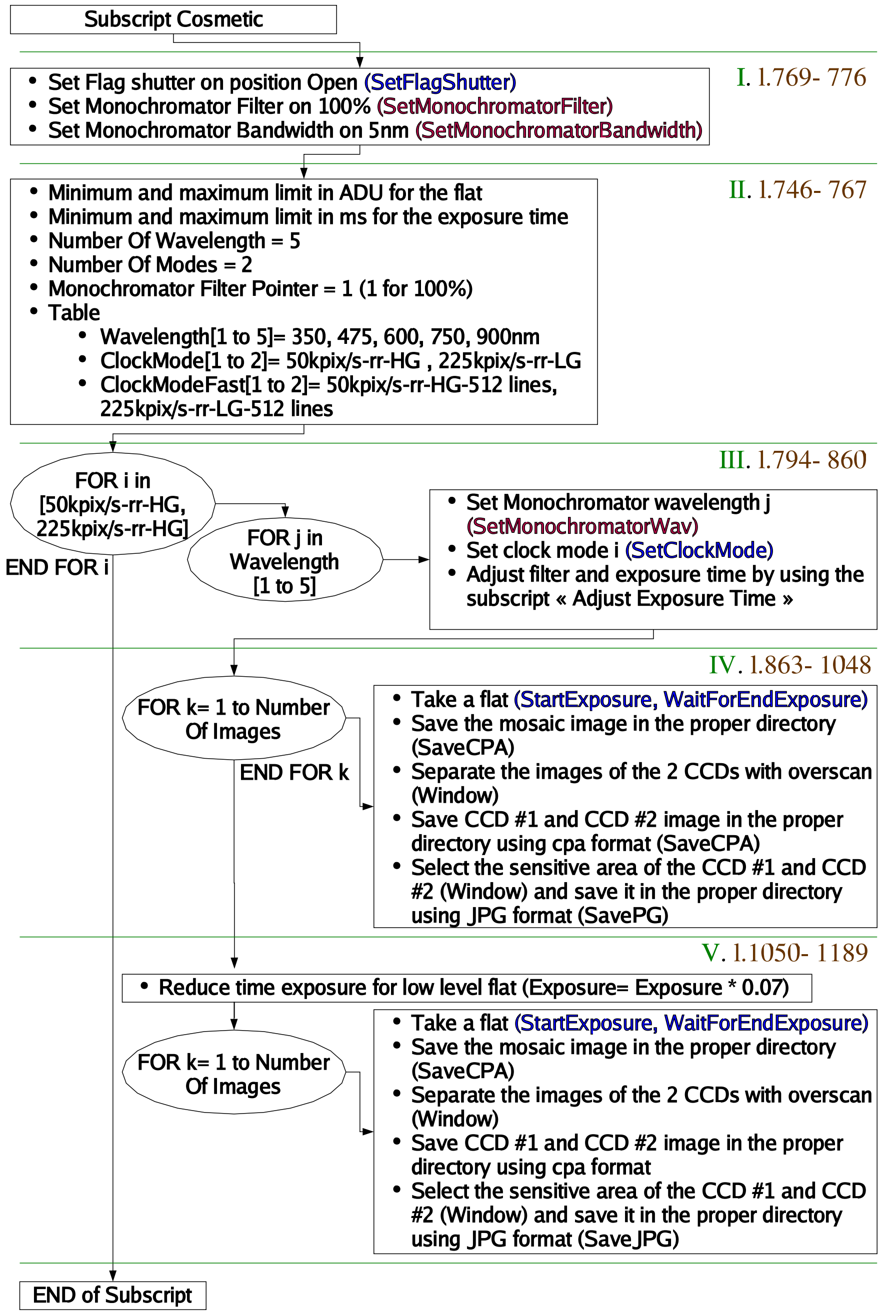

Figure 2.11: Diagram of the subscript, Cosmetic.

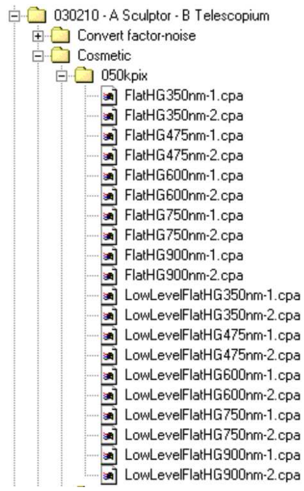

Figure 2.12: Path to save the cosmetic images.

Fig. 2.13 Section II. Initialisation of the test bench.

The monochromator of the test bench is set for the linearity measurement (Wavelength set to 632nm, bandwidth to 8nm, filter to 100%). The shutter is set to "close during the integration" and the clock mode used is the clock mode 9 (225 Kpx/s, Right-Right port, Low Gain, 512 lines).

Fig. 2.13 Section III. Two images are taken using the TDI method (Transfer During Integration). These images are saved in the directory 'yymmdd - A CCD#1 - B CCD#2 \ Linearity. These images are cut in two parts and saved in the directory of the CCD#1 \ Linearity and CCD#2 \ Linearity. The name of the images are LinearityTDI-1.cpa and LinearityTDI-2.cpa.

Fig. 2.13 Section IV. After taking the TDI images, two flats are taken with the same mode used to acquire the two first images. Before taking these two flats the exposure time is adjusted by using the subscript 'Adjust exposure time'. The mosaic images are saved in the same directory than the mosaic TDI images. These images are cut in two parts (for each CCD) and saved in CCD#1 \ Linearity and CCD#2 \ Linearity directory respectively.

There is no Linearity function implemented in the script language. To calculate the linearity the function Linearity from the PRiSM board has to be used.

2.3.4 Pocket Pumping

The diagram Fig. 2.14 explains the subscript Pocket pumping.

Fig. 2.14 Section I. Initialisation of the test bench.

In this section the monochromator is set with the following parameters: The filter is set to 100%. The wavelength to 630nm. The bandwidth to 7nm. The

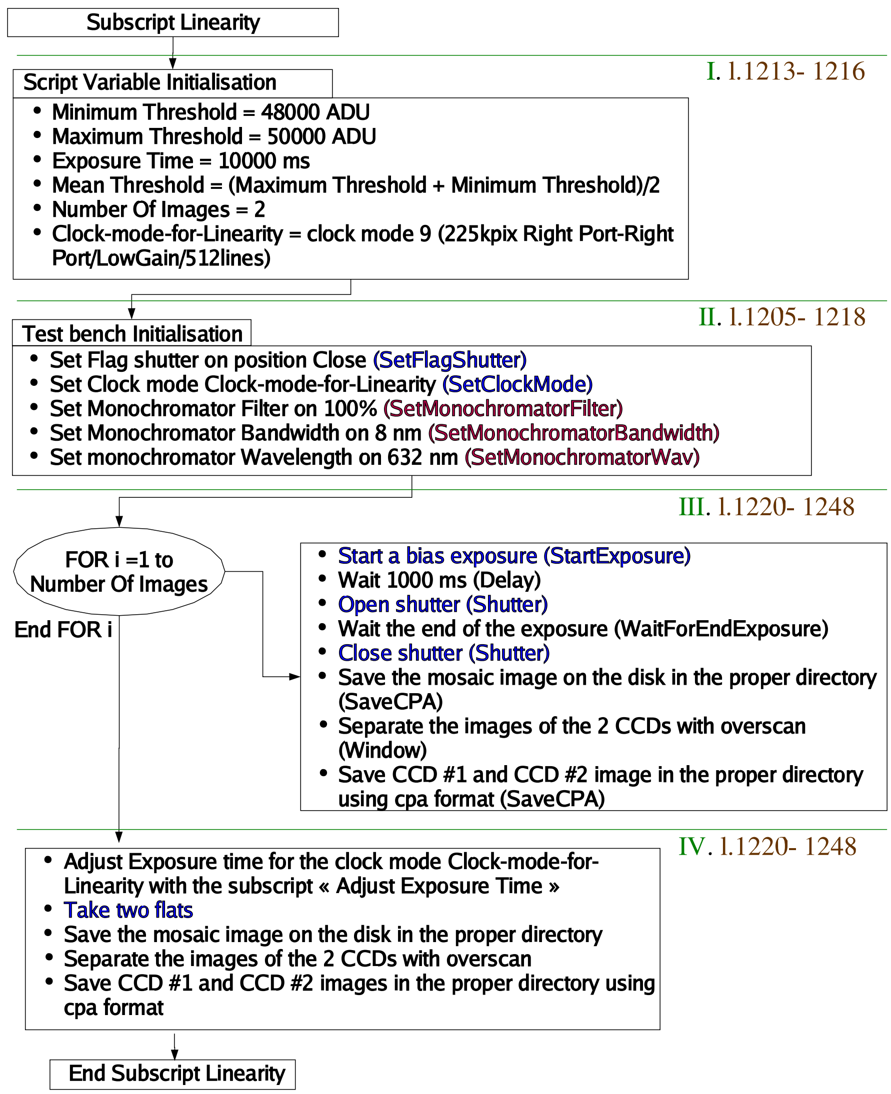

Figure 2.13: Diagram of the subscript, Linearity.

clock mode to 2 (225 Kpx/rr/HG/512). The shutter set to the position "Open during integration".

Fig. 2.14 Section II. Initialisation of the variables for the sub-function PocketPumping. The starting point of the exposure time is set to 2000ms (Exposure=2000). The boundaries of the exposure time are 2000ms and 100000ms (TexpMin=2000, TexpMax=100000). The boundaries for the intensity are 46 000 ADU and 50 000 ADU (BorneInf=23000, BorneSup=25000).

Fig. 2.14 Section III. The exposure time (variable Exposure) and the filter (variable MonochromatorFilter) are adjusted to have light between 46 000 and 50 000 ADU with the subscript 'Adjust Exposure Time'.

Fig. 2.14 Section IV. Then two images are taken by using the clock mode 11. This mode permits to shift the complete image ten lines down and ten lines up 500 times. The pixels which capture few electrons are not detected with the normal reading. They become visible with this special treatment. These mosaic images are saved in the directory 'yymmdd - A CCD#1 - B CCD#2 \ PocketPumping'. These images are cut in two parts (one for each CCD) and saved in CCD#1 \ PocketPumping and CCD#2 \ PocketPumping directory respectively.

2.3.5 Quantum Efficiency

The diagram Fig. 2.15 and Fig. 2.16 explain the subscript Quantum Efficiency.

Fig. 2.15 Section I. Initialization of the test bench.
In this section the monochromator is set with the following parameters:

- The filter is set to 100%
- The bandwidth is set to 7nm
- The wavelength is set to 630nm
- The shutter is set to the "Open during integration" mode
- The clock mode is set to the mode 2

Fig. 2.15 Section II. Initialization of the variables used in this sub-function. During this initialization two files are loaded, CalibrationFile.txt and WindowTransmission.txt. In the CalibrationFiles.txt file, the wavelength, the values of the flux in the sphere (diode) and on the CCDs, the error of the CCD flux and the error of the diode flux will be finded for each wavelength. These numbers will be used to calculated the new flux of light on the chips for the new exposures. They are loaded in the following real tables: wavelength[i], FluxCCD[i], FluxDiode[i], ErrFluxCCD[i], ErrFluxDiode[i]. Each table has the size of the number of lines found in the CalibrationFile.txt file. The WindowTransmission.txt file contents the transmission factor at each wavelength (all the values are equal to 1 because now the window is included in the calibration procedure,

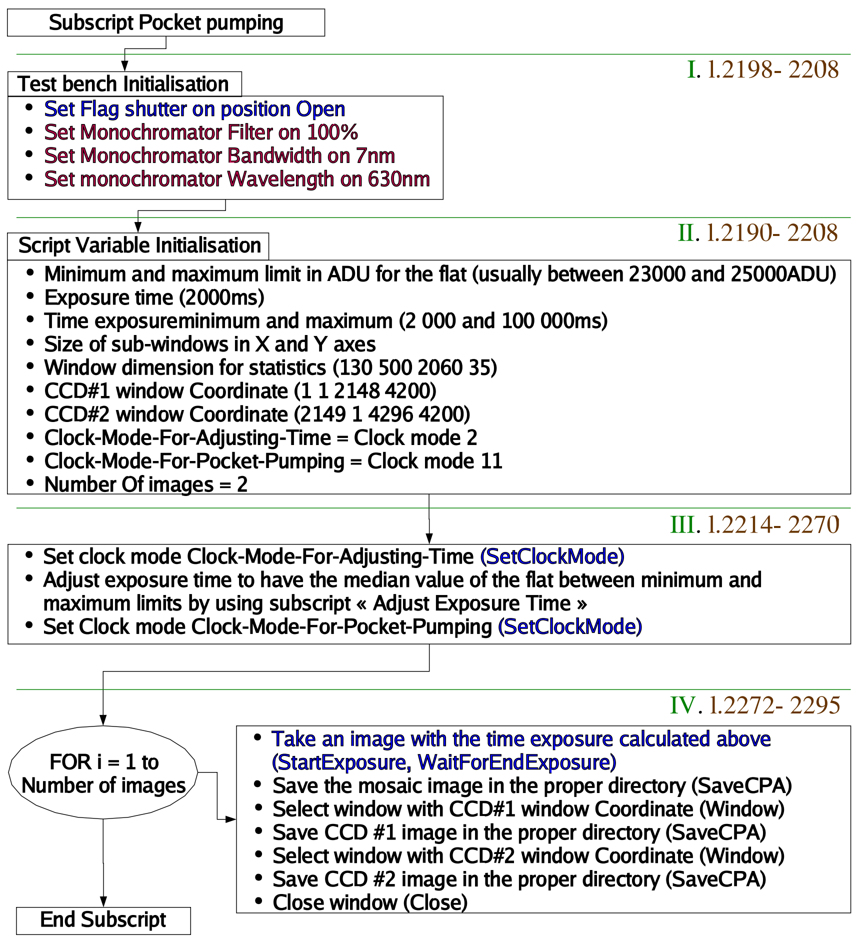

Figure 2.14: Diagram of the subscript, Pocket pumping.

the values are loaded in the real table Transmission[i]). These tables are used by the GetQE function to calculate the QE and the PRNU.

The starting point of the exposure time is set to 1050ms (Exposure=1050). The boundaries of the exposure time are 3000ms and 80000ms (TexpMin=3000, TexpMax=80000). The boundaries for the intensity are 46 000 ADU and 50 000 ADU (BorneInf=23000, BorneSup=25000).

The position of the window to calculate the conversion factor and quantum efficiency are defined. For the CCD#1: X11=1584, Y11=1, X12=2096, Y12=512. For the CCD#2: X21=2197, Y21=1, X22=2709, Y22=512.

The number of sub-windows are 10 in X and 10 in Y (NbWindowX=10, NbWindowY=10). In each sub-window the conversion factor and the quantum efficiency will be calculated. Basic statistical studies will be done to calculate the mean conversion factor and mean QE.

The function GetQE needs also the size of the pixels in microns (PixelSizeX=15, PixelSizeY=15), some old parameters not used anymore are set to 1 (DistanceMeasurement=1, DistanceCalibration=1).

Fig. 2.15 Section III. After the initialisation of the variables and the test bench, two bias and two flats are taken to calculate the conversion factor and the read out noise for the two CCDs. These mosaic images are saved in the directory 'yymmdd - A CCD#1 - B CCD#2 \ QE'. The values calculated for CCD#1 are saved in the file 'CCD#1 \ QE \ ResultQE.txt' and the values for the CCD#2 in the file 'CCD#2 \ QE \ ResultQE.txt'. The mosaic image is cut in two parts and saved in the same directory than for the resultQE.txt files. After, the temperature is taken and saved in the resultQE.txt files. Five bias are taken and a median bias is computed. In this image, two windows are selected and saved in the directory 'CCD#1 \ QE and 'CCD#2 \ QE. A master bias is also cumputed with 5 bias (MedianBias).

Fig. 2.16 Section IV. For each wavelength the exposure time and the filter are adjusted with the function 'Adjust exposure time'. Then, the acquisition of the image is realized and the intensity of light is measured. The Flux, the Wavelength and Bandwidth are saved in the header and the image saved on the disk in the directory 'yymmdd - A CCD#1 - B CCD#2 \ QE'. The mosaic image is cut (one image for each ccd) in two parts and saved on the disk in the directory 'CCD#1 \ QE \' and 'CCD#2 \ QE \'.

Fig. 2.16 Section V. In this section the calculation of the quantum efficiency is done. The path of images, the master bias, the coordinates of the window where to calculate the QE, the size of the pixels, the conversion factor, old parameters, the bandwidth and the tables loaded from the CalibrationFile.txt and WindowTransmission.txt files are given to the function GetQE to calculate the QE and the PRNU. Two files are created. The first contents the QE and PRNU table. This file is used to create the Test Report. The second one contents the table and detailed information to compute the QE and the PRNU. This file is used to check the results in case of problems.

Fig. 2.16 Section VI. A second acquisition is performed from 1100nm to 300nm to check possible hysteresis problems.

Fig. 2.16 Section VII. The calculation of the quantum efficiency is done. Two files are created like in the section V.

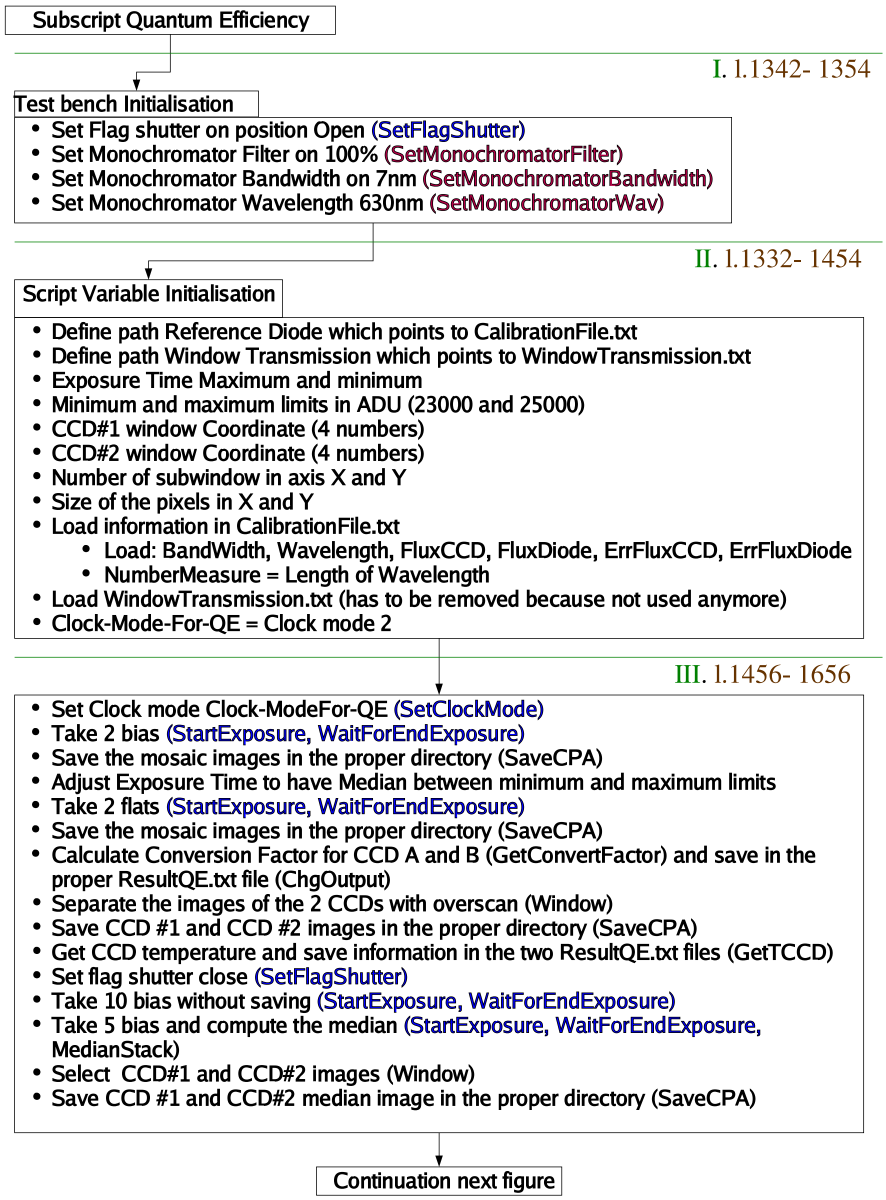

Figure 2.15: First part of the subscript, Quantum Efficiency.

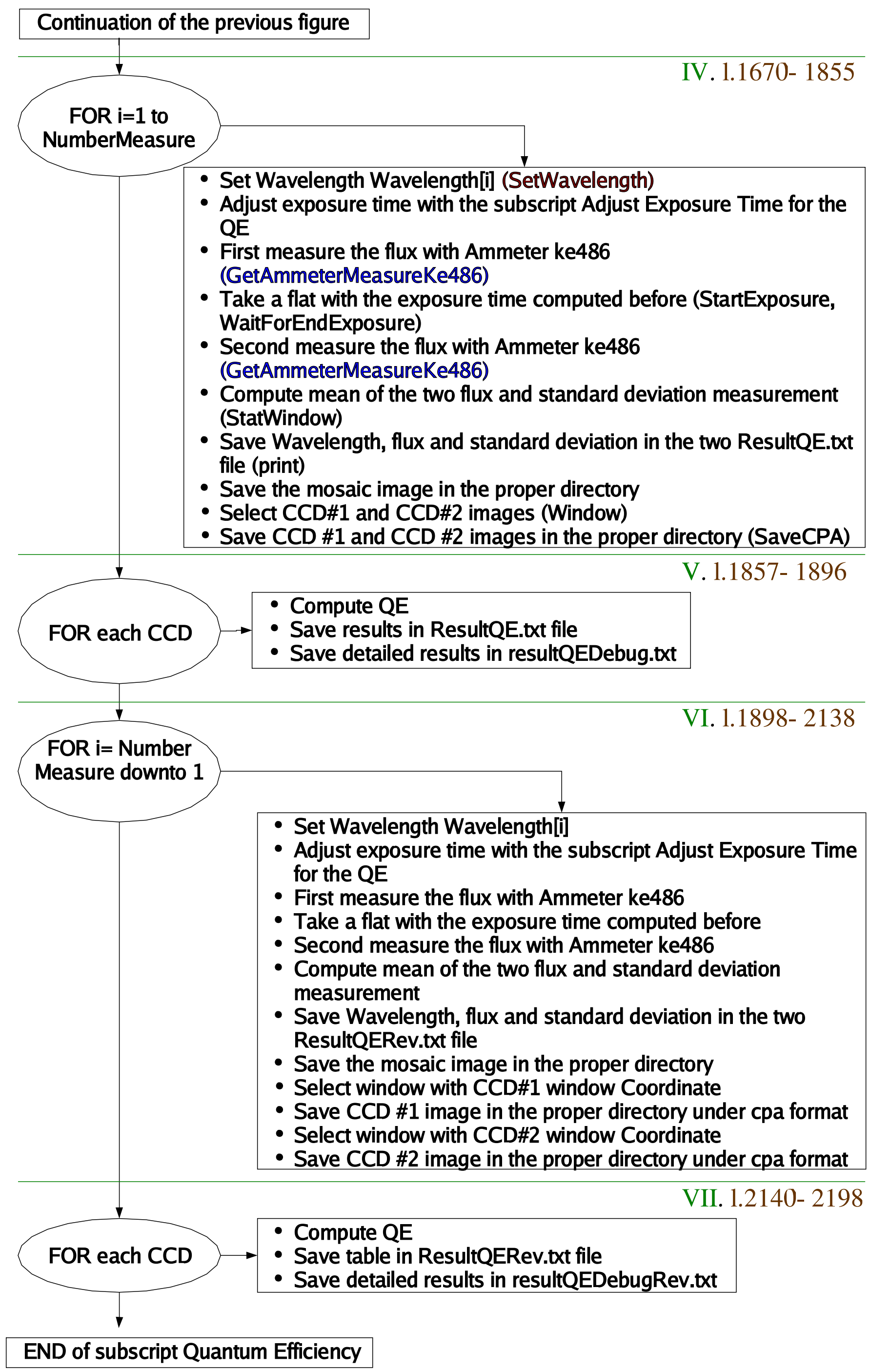

Figure 2.16: Second part of the subscript, Quantum Efficiency.

Chapter 3

Script Language

3.1 Introduction

PRiSM is an astronomical software written by Cyril Cavadore and Boris Gaillard. Data acquisition and high level scientific data reduction can be performed. In this chapter, the functions used in the scripts will be described.

3.2 PRiSM language for the scripts.

All the functions of PRiSM are not described in this annexe. You will find only the function used in the script DarkAndBiasAcquisition.pgm and LightAcquisition.pgm. To have the complete list and description of all the functions, please contact Cyril Cavadore, author of this software.

The script language is a structural language which resembles to BASIC. Some words are reserved for the structure of the program (if, then,... see the list below) and the other for the predefined functions (OpenCPA, SaveCPA, GetConversionFactor,...). The predifined functions are used to make specific tasks (open an image, calculate Read-Out-Noise,...).

3.2.1 Composition of a script

A script is composed of two parts.

- The main program
- The sub programs

The sub programs are usully after the main program.

3.2.2 Comments

To insert comments in a script, two commands can be used: REM or //

Example:

```
REM this line is a comment
// this one too
```

3.2.3 Mathematical operation

Simple operation can be done in the script: +, -, *, /

Comparisons are also possible: >, <, =, ≤ (<=), ≥ (>=)

3.2.4 Variables

To declare a string variable: a$="example"

To declare an integer and real variable: b=123, c=1.23

PRiSM makes no difference between integer and float number.

3.2.5 Table

Under PRiSM it is possible to create table of number, of string and images.

example:

```
table of number:
tab[1]=1
tab[2]=2
table of string:
tab[1]$="a"
tab[2]$="b"
table of images:
Open tab[1] "c:\images\Images1.cpa"
Open tab[2] "c:\images\Images2.cpa"
```

Tips: Never put space before and after "=". You will have a bug!

3.2.6 The PRiSM vocabular.

PRiSM pocesses reserved words for the structure of the language.

- if then else endif
- for next
- goto
- gosub return
- print

And also specific function for the data acquisition and the data reduction. These functions are described below.

3.2.7 print

Description: This function permits you to print text or the contents of a variable (string or integer).

Syntaxe: `print var$, print var`

Input: var$ or var: variable.

Output: No Output

Example: `print "Hello world!"`

3.2.8 for... next

Description: To make a loop the function for... next can be used.

Syntaxe:

```
for counter=i j
next
```

Input: counter: Variable which will vary.
i: Number which is your starting point.
j: Number which is your ending point.

Output: No Output

Example:

```
for i=1 5
print i
next
```

3.2.9 if then else endif

Description: Keywords to make a condition test. The conditions are: =, >, <, ≤ (<=), ≥ (>=)

Syntaxe:

```
if condition then
instructions
else
instructions
endif
```

Input: No Input

Output: No Output

Example:

```
a=1
if a=2 then
print "a= 2"
else
print "a not equal to 2"
endif
```

3.2.10 goto

Description: During the execution of a script, you can jump from one part to the program to another with this function.

Syntaxe:

```
goto sticker
```

Input: sticker: Name of the sticker where you want to jump in the script.

Output: No Output

Example:

```
counter=1
step1:
print counter
counter=counter+1
```

```
if counter<10 then
goto step1:
else
print "End of the program"
```

3.2.11 gosub... return

Description: This function permits you to create a sub program. During the execution of a script, the compiler jumps to the sub program specified when it reaches `gosub`. Once the sub program is executed, the compiler comes back to continue the script.

Syntaxe:

```
gosub Sticker:
:
Sticker:
instructions
return
```

Input: sticker: Name of the sticker where you want to jump.

Output: No Output

Example:

```
___________Main program____________
counter=1
if counter<10 then
gosub SubFunction1:
else
print "End of the program"


___________Sub functions___________
SubFunction1:
print counter
counter=counter+1
return
```

3.3 Functions in the dark and light script acquisition.

3.3.1 Open_Filetxt_For_Read

Description: This function opens a file in the mode read.

Syntaxe: `Open_Filetxt_For_Read F1 PathFile`

Input: PathFile: Path where the file is recorded.

Output: F1: Name of the file in the script environment.

Example: `Open_Filetxt_For_Read F1 "c:\txt\txtfile.txt"`

3.3.2 Open_Filetxt_For_Rewrite

Description: This function opens a file to write in.

Syntaxe: `Open_Filetxt_For_Rewrite F1 PathFile`

Input: PathFile: Path where the file is saved.

Output: F1: Name of the file in the script environment.

Example: `Open_Filetxt_For_Rewrite F1 "c:\txt\txtfile.txt"`

Comment: To write in the opened file you need the function WriteFileTxt

3.3.3 WriteFileTxt

Description: This function permits you to write a line in the file you have opened with the function `Open_Filetxt_For_Rewrite`.

Syntaxe: `WriteFileTxt F1 Line$`

Input: F1: Name of the file defined by the function `Open_Filetxt_For_Rewrite`
Line$: String variable which contents the text to save.

Output: The file modified on the disk.

Example: `WriteFileTxt F1 "Hello World!"`

3.3.4 EndOfFile

Description: This function checks if you are at the end of the file opened with Open_Filetxt_For_Read.

Syntaxe: `EndOfFile F1 Eof`

Input: F1: File opened with the function `Open_FileTxt_For_Read`

Output: Eof: Integer.
if Eof = 0 you are not at the end of the file.
if Eof = 1 you are.

Example:

```
Eof=0
Sticker:
ReadFileTxt F1 Line$
Print Line$
EndOfFile F1 Eof
if Eof=0 then Goto Sticker:
```

3.3.5 ReadFileTxt

Description: This function permits you to read a line of a file opened with the function Open_Filetxt_For_Read.

Syntaxe: ReadFileTxt F1 Line$

Input: F1: File opened with the function `Open_FileTxt_For_Read`

Output: Line$: Line of the file F1

Example: See example of `EndOfFile`

3.3.6 CloseFileTxt

Description: This function closes a file opened with the function Open_Filetxt_For_Read or Open_Filetxt_For_Rewrite.

Syntaxe: `CloseFileTxt PathFile$`

Input: PathFile$: Path where the file is saved on the disk.

Output: No Output.

Example: `CloseFileTxt "c:\txt\txtfile.txt"`

3.3.7 NbMot

Description: This function counts the number of words in a string variable.

Syntaxe: `NbMot a$ nbword`

Input: a$: String variable.

Output: nbword: Number of words in the variable a$

Example:

```
a$="Hello world"
NBMOT a$ nbword
print a$ (2 will be given)
```

3.3.8 Str

Description: This function converts a number in a string.

Syntaxe: `Str Number tmp$`

Input: Number: Integer or float.

Output: tmp$: String variable with the number converted.

Example:

```
Number=1
Str Number tmp$
print tmp$ "1" (will be given)
```

3.3.9 StartExposure, WaitForEndExposure

Description: These functions permit to make the acquisition of an image. StartExposure sends the command to start an integration with the time specified (in milliseconds) after this function. WaitExposure stops the execution of the script until the image is saved in the SPARC hard drive.

Syntaxe: `StartExposure ExposureTime`
`WaitForEndExposure`

Input: ExposureTime: Integer which specifies the time of an integration (always in millisecond).

Output: No Output

Example:

```
ExposureTime=1000
StartExposure ExposureTime
WaitForEndExposure
GetLastImageExposure Image
```

3.3.10 GetLastImageExposure

Description: This function transferts the image saved after the call of StartExposure and WaitExposure into the PRiSM evironment to work with.

Syntaxe: `GetLastImageExposure Image`

Input: No Input.

Output: Image: Name of the image in the script environment.

Example: See example of the section "StartExposure, WaitForEndExposure".

3.3.11 SaveCPA

Description: This function permits to save an image under the CPA format. It is a comppressed format used under PRiSM.

Syntaxe: `SaveCPA Image PathImage$`

Input: Image: Name of the image you want to save.
PathImage$: Path of the image where you want to save the image.

Output: No Output

Example: `SaveCPA Image "C:\Images\Image01.cpa"`

3.3.12 Close

Description: This function closes an image open with Open.

Syntaxe: `Close Image`

Input: Image: The name of the image in the script environment.

Output: No Output

Example: See the section "Open".

3.3.13 SetFlagShutter

Description: This function controls the SESO shutter.

Syntaxe: SetFlagShutter Number

Input: Number:
1, the shutter stay close during the integration time.
0 authorized the opening of the shutter.

Output: No Output

Example:
```
Close=1
Open=0
SetFlagShutter Close
```

3.3.14 SetTransformRescaleFITsDynamic

Description: This function transforms a 16 bits image into an 15 bits signed image. Your image sampled with 65536ADU will have under PRiSM 32768 ADU (each pixel are divided by 2) if you use the the parameter 0.5. With the parameter 1, PRiSM cuts the image at 32767 ADU. All the pixels higher than this value will have 32767 ADU.

Syntaxe: `SetTransformRescaleFITsDynamic Number`

Input: Number:
0.5 to adjust the pixel values of the image.
1 to cut the pixel values of the image at 32767ADU

Output: No Output.

Example:
```
AdjustFits=0.5
CutFits=1
SetTransformRescaleFITsDynamic CutFits
```

3.3.15 SetLoadFitsToFloating

Description: This function permits to open an image as integer or real.

Syntaxe: `SetLoadFitsToFloating Number`

Input: Number:
0 for integer
1 for floating point

Output: No Output

Example:
```
Integer=0
Real=1
SetLoadFitsToFloating Integer
```

3.3.16 SetClockMode

Description: This function permits you to choose the clock mode you need to acquire an image.
Definition of the clock mode:
Mode 1 : 225 Kpx/rr/HG
Mode 2 : 225 Kpx/rr/HG/512
Mode 3 : 50 Kpx/rr/HG
Mode 4 : 50 Kpx/rr/HG/512
Mode 5 : 50 Kpx/rlrl/HG
Mode 6 : 50 Kpx/rlrl/HG/512
Mode 7 : 625 Kpx/rr/HG
Mode 8 : 225 Kpx/rr/LG
Mode 9 : 225 Kpx/rr/LG/512
Mode 10 : 225 Kpx/rr/LG
Mode 11 : 225 Kpx/rr/HG/PocketPumping
Mode 12 : Integrate Test High
Mode 13 : Integrate Test Low
Mode 14 : 225 Kpx/rlrl/HG
Mode 15 : 225 Kpx/rlrl/HG/512
Check regularly if there is no new mode.
These mode are defined in the fcdSlcuCon software. See section 1.2.9 about the fcdSlcuCon software.

Syntaxe: `SetClockMode Number`

Input: Number: 1-15 which corresponds to the mode to use (see above).

Output: No Output

Example:
```
ClockModeDarkLG225kpix=8
SetClockMode ClockModeDarkLG225kpix
```

3.3.17 Open

Description: This function is used to open an image on the disk. You have to specify the path of the image and the name of this one to use it under the PRiSM environment.

Syntaxe: `Open Im1 Image$`

Input: Image$: Path of the image saved on the disk.

Output: Im: Name of the image in the script environment.

Example:
```
Image$="C:\Im\Image.cpa"
Open Im1 Image$
```

3.3.18 Duplicate

Description: This function duplicates an image and give to this new image the name you specify.

Syntaxe: Duplicate Im1 Im2

Input: Im1: Image prealably opened with the function Open.

Output: Im2: The name of your duplicated image.

Example:
```
Open Im1 "C:\Image\Image01.cpa"
Duplicate Im1 Im2
```

3.3.19 Window

Description: This function crops an image in the rectangle you specify. The coordinate of two opposite corner are given.

Syntaxe: `Window Im1 X1 Y1 X2 Y2`

Input: Im1: The image to crop.
X1, X2: The X and Y position of the first corner.
X2, Y2: The X and Y position of the second corner.

Output: Im1: The cropped image.

Example:
```
X1=50
Y1=50
X2=100
Y2=100
Open Im1 "C:\Image\Image01.cpa"
Window Im1 X1 Y1 X2 Y2
```

3.3.20 MedianStack

Description: This function computes the median of an images' group . This function needs a table of images, two pointers to specify the beginning and the end in the table and the name of the median image.

Syntaxe: `MedianStack Image StartPoint EndPoint Median-Image`

Input: Image: Table of images.
StartPoint: The first image in the table taken in account for the median stack.
EndPoint: Last image in the table taken in account for the median stack.

Output: Median-Image: Name of the median image in the script environment.

Example:
```
Open Im[1] "C:\Image\Im01.cpa"
Open Im[2] "C:\Image\Im02.cpa"
Open Im[3] "C:\Image\Im03.cpa"
One=1
NumberOfImages=3
MedianStack Im One NumberOfImages Median-Im
```

3.3.21 SaveJPG

Description: This function saves the image on your disk under the JPG format. You have to specify the image and the path where you want to save it.

Syntaxe: `SaveJPG Image Path$`

Input: Image: The image to save.
Path$: The path where to save the image on the disk.

Output: No Output

Example: `SaveJPG Image "C:\Images\Image01.jpg"`

3.3.22 BinningReduce

Description: This function reduces an image. Two parameters are given to specify the size of the rectangle which will be used to calculate the new median pixels.

Syntaxe: `BinningReduce Image Number01 Number02`

Input: Image: Image to reduce.
Number01, Number02: Dimension of the rectangle which will be replaced by the median value of the new pixel.

Output: Image: The image reduced.

Example: `BinningReduce Im 15 15`

3.3.23 SaveFIT

Description: This function saves the image on your disk under the FITS format. You have to specify the name of the image and the path where you want to save it.

Syntaxe: `SaveFIT Image Path$`

Input: Image: The image to save.
Path$: Path where to save the image on the disk.

Output: No Output.

Example: `SaveFIT Image "C:\Images\Image01.fit"`

3.3.24 GetTCCD

Description: This function permits to read in the header of an image the temperature.

Syntaxe: GetTCCD Image Temperature

Input: Image: Name of the image.

Output: Temperature: The temperature saved in the header by Fiera.

Example:
```
Open Im1 "C:\Image\Image01.cpa"
GetCCD Im1 Temp
```

3.3.25 SetMonochromatorFilter

Description: This function permits to change the filter of the monochromator. There are 5 filters:
1 : 100% of the light passes through the monochromator
2 : 50% of the light passes through the monochromator
3 : 25% of the light passes through the monochromator
4 : 10% of the light passes through the monochromator
5 : 1% of the light passes through the monochromator

Syntaxe: `SetMonochromatorFilter Number`

Input: Number: 1-5 which correspond to the filter, see above.

Output: No Output.

Example:
```
Filter100%=1
Filter050%=2
Filter025%=3
Filter010%=4
Filter001%=5
SetMonochromatorFilter Filter100%
```

3.3.26 SetMonochromatorBandWidth

Description: This function controls the bandwith of the output monochromator. You can choose between 1 to 10nm.

Syntaxe: `SetMonochromatorBandWidth Number`

Input: Number: 1-10 which corresponds to the size of the bandwidth centered on the wavelength chosen with the function SetMonochromatorWav.

Output: No Output

Example:
```
BandWith7nm=7
SetMonochromatorBandWidth BandWith7nm
```

3.3.27 SetMonochromatorWav

Description: This function sets the wavelength at the output of the monochromator. You can select a number between 300 and 1100nm.

Syntaxe: `SetMonochromatorWav Number`

Input: Number: 300-1100 which correspond to the output wavelength.

Output: No output

Example:
```
Wavelength630nm=630
SetMonochromatorWav Wavelength630nm
```

3.3.28 StatWindow

Description: This function permits to realize a statistical study in a selected window.

Syntaxe: StatWindow Image X1 Y1 X2 Y2 Mean SquareMean StandardDeviation Maximum Minimum Flux Median

Input: Image: The opened image into you want to make a statiscal study.
X1, Y1, X2, Y2: Coordinate of the two opposite rectangle's corners where the study has to be done.

Output: Mean: Mean of the pixels.
SquareMean: Square mean inside the defined box.
StandardDeviation: Standard deviation inside the defined box.
Maximum: Maximum pixel inside the defined area.
Minimum: Minimum pixel inside the defined area.
Flux: Flux inside the defined box.
Median: Median of the pixel values inside the defined box.

Example:
```
Open Im1 "C:\Image\Image01.cpa"
StatWindow Image X1 Y1 X2 Y2 Mean SquareMean StandardDeviation
Maximum Minimum Flux Median
print Flux (will give you the flux)
```

3.3.29 Round

Description: This function rounds a number to have an integer value.

Syntaxe: `Round Number Result`

Input: Number: Number to round

Output: Result: Rounded number

Example:
```
Exposure=1.23
Round Exposure RoundedExposure
Print RoundedExposure (will print 1 on the screen)
```

3.3.30 ClockModeDescribe

Description: Extract in the header the clock mode used during the exposure.

Syntaxe: ClockModeDescribe

Input:

Output:

Example: ClockModeDescribe ClockModeForConvertionFactorandNoise050kpixHG

3.3.31 GetConvertfactor

Description: This function computes the conversion factor (CF) and the read out noise (RON) inside a defined rectangle in the image.

Syntaxe: `GetConvertfactor Flat1 Flat2 Bias1 Bias2 X1 Y1 X2 Y2 NbWindowX NbWindowY ConvertFactor RMSNoise`

Input: Flat1, Flat2: Two flats.
Bias1 and Bias2: Two bias
X1, Y1, X2, Y2: The coordinates of the opposite rectangle's corners where you want to calculate the CF and the RON.
NbWindowX, NbWindowY: Number of subdivision in X and Y to compute the CF and the RON.

Output: ConvertFactor: The conversion factor.
RMSNoise: The read out noise.

Example:

```
Open Im1 "C:\Image\Image01.cpa"
Open Im2 "C:\Image\Image02.cpa"
Open Im3 "C:\Image\Image03.cpa"
Open Im4 "C:\Image\Image04.cpa"
X1=10
Y1=10
X2=1000
Y2=1000
NbWindowX=10
NbWindowY=10
GetConvertfactor Flat1 Flat2 Bias1 Bias2 X1 Y1 X2 Y2 NbWindowX
NbWindowY ConvertFactor RMSNoise
print ConversionFactor RMSNoise
```

3.3.32 Autovisu

Description: The cuts of the image are adjusted.

Syntaxe: `Autovisu Image`

Input: Image: The image onto the adjustments have to be done.

Output: The image with the cuts adjusted.

Example:

```
Open Im "C:\images\Im01.txt"
Autovisu Im
```

3.3.33 Sub

Description: This function realizes the substraction of two images.

Syntaxe: `Sub Image01 Image02`

Input: Image01, Image02: Two images. The first image will be substracted by the second one.

Output: Image01: The result of the substraction will be stored in Image01.

Example:
```
Open Im1 "C:\Image\Image01.cpa"
Open Bias "C:\Image\Bias01.cpa"
Sub Im1 Bias
```

3.3.34 Delay

Description: This function inserts a delay (in milliseconds) in the execution of the script.

Syntaxe: Delay Number

Input: Number: The delay in millisecond (1000).

Output: No Output

Example:
```
TimeToWait=1000
Delay TimeToWait
```

3.3.35 ChgOutput

Description: This function changes the output of the print function to a file.

Syntaxe: `ChgOutput Path$`

Input: Path$: Path where you want to write in a file on the disk.

Output: No Output

Example:
```
ChgOutput "C:\Txt\result.txt"
print "Hello world!" ("Hello world!" will be written in
the file result.txt)
```

3.3.36 GetAmmeterMeasureKe486

Description: This function sends a request to the ammeter ke486 to measure the intensity of the photodiode inside the sphere. This function returns a mean and the standard deviation (based on 10 measurements).

Syntaxe: `GetAmmeterMeasureKe486 Measure StdDevMeasure`

Input: No Input

Output: Measure: The measure of the intensity inside the sphere.
StdDevMeasure: The standard deviation of the measure.

Example:
```
GetAmmeterMeasureKe486 Measure StdDevMeasure
print Measure StdDevMeasure
```

3.3.37 setFlux

Description: This function adds the intensity measured by the ammeter ke486 into the header of an image.

Syntaxe: setFlux Image Flux

Input: Image: The image into the flux value has to be written.
Flux: The intensity measured.

Output: No Output

Example:
```
GetAmmeterMeasureKe486 Measure StdDevMeasure
setFlux Image Measure
```

3.3.38 setWavelength

Description: This function writes in the header the wavelength used during the exposure.

Syntaxe: setWavelength Image Number

Input: Image: Image into the parameter "wavelength" has to be written in the header.
Number: 300-1100 which corresponds to the wavelength used to acquire the image.

Output: No Output

Example:
```
Wavelength=660
setWavelength Img Wavelength (will write this information
in the header)
```

3.3.39 setBandwidth

Description: This function writes in the header the bandwith of the monochromator used during the exposure.

Syntaxe: `setBandwidth Image Number`

Input: Image: The image into the bandwith has to be written. Number: The bandwith used during the exposure.

Output: No Output

Example:
```
Bandwith7nm=7
setBandwidth Img Bandwith7nm
```

3.3.40 GetQE

Description: This function computes the quantum efficiency and the photon response non uniformity of the ccds based on the images given. This function uses also parameters which come from two files. The first file contains the information produced by the calibration procedure. The second one contains the window transmission coefficient at different wavelength.

Syntaxe: GetQE TabIm NbMeasure MedianBiasA 1 1 512 512 PixelSizeX PixelSizeY ConvertFactor DistanceMeasurement DistanceCalibration Bandwith WavLen FCCD FDiode DebugMode EFCCD EFDiode Tr Wavel QE ErrQE PRNU Enthropy

Input: TabImA: Table with the images.
NbMeasure: The number of images
MedianBias: The master bias image
X1, Y1, X2, Y2: The coordinates of the opposite corner of the rectangle where the QE has to be calculated.
PixelSizeX, PixelSizeY: The size of the pixel in X and Y in micron.
ConvertFactor: The conversion factor.
DistanceMeasurement, DistanceCalibration: Have to be equal to 1. These two parameters are not used anymore in the calculation.
Bandwith: The bandwith setted in the monochromators during the acquisition of the images.
DebugMode: The mode you want to use (standard and debug)
The following tables come from the calibration files.
WavLen: Wavelength
FCCD:Flux on the CCD
FDiode: Flux on the photodiode
EFCCD: Error of measurements on the flux on the CCD
EFDiode:Error of the measurements on the flux on the CCD
Tr: Transmission factor of the glass at each wavelength

Output: Wavel: Table with the wavelengths.
QE: Table with the Quantum Efficiency.
ErrQE: Table with the error of the QE measurement
PRNU: Table with the PRNU.
Enthropy: Table with the Enthropy.
This function will also produce two graphic representations. One for the QE the other for the PRNU.

Example:

```
Open Im[1] "C:\Image\Im01.cpa"
Open Im[2] "C:\Image\Im02.cpa"
Open Im[3] "C:\Image\Im03.cpa"
NbMeasure=3
Open MedianBias "C:\Image\MasterBias.cpa"
X1=10
Y1=10
```

```
X2=1000
Y2=1000
PixelSizeX=15
PixelSizeY=15
ConvertFactor=0.55
DistanceMeasurement=1
DistanceCalibration=1
Bandwith=5
GetQE TabIm NbMeasure MedianBiasA 1 1 512 512 PixelSizeX
PixelSizeY ConvertFactorCcdA DistanceMeasurement DistanceCalibration
Bandwith WavLen FCCD FDiode DebugMode EFCCD EFDiode Tr
Wavel QE ErrQE PRNU Enthropy
```