XNIR

User's Guide

Version 1.1

Carlo Baffa
Vladimir Gavryusev
Elisabetta Giani

November 24, 1997

Contents

List of Figures

Release notes

Version 1.1

Some bugs found during the exploitation are fixed. Some new functionality added:

- The new task, which can be invoked from the main menu by the button **Source - Sky** is added. The main purpouse of this task is to provide the check of the source pointing, and, probably, to adjust the necessary acquisition timings. This task makes an acquisition of the pointed source, moves to the other place in the sky, then subtract this two frames and shows the result in SAOimage. The exact sequence of the acquisitions is defined by the mosaic file **checkfr.txt**. The sequence and the pointing positions could be changed there, if necessary. Anyway only two last frames will be subtracted.

- Two buttons **Sky Subtraction** and **Subtract 2 last Frames** are added to the Display Frame sub-menu. First one permits to choose two frames to be subtracted one from another. The result can be viewed then by use of the SAOimage. Second one does the subtraction of two last acquised frames and result visualisation in one click.

- The additional label with the information about the *Spot Size* is added to the **Image** widget.

There are some additional items stored in the file *xnir.ini*.

Version 1.00

This is the first production version of the Xnir, released at 10.09.96. It supports all basic measurement tasks as the **nirtest** does. The user interface is now X11 widget based interface (created upon ATHENA widget set). The software is created as a network distributed software, while basically is executed on the PC dedicated for the measurements.

The manual contains the basic information necessary for the use of the software for the measurements, its configuration in the user dependent part and some technical information concerning the software and file system organization, which can help the observer in his/her work at the telescope.

The comments on the features of the software and the manual and any suggestions are welcome. Please contact us at


```
Vladimir Gavryusev:       vladimir@arcetri.inaf.it
Carlo Baffa:              baffa@arcetri.inaf.it
```

Introduction

This manual describes how to use the software Xnir (**X**11 **N**ear **I**nfra**R**ed), which was created for the real time control of the measurements using the Arcetri two-dimensional infrared instruments ARNICA [1] and LONGSP [2].

Planning and executing the infrared measurements is *different* from executing the optical ones. The reader is strongly suggested to consult [3] for a full discussion of the physical effects and of the subtraction techniques.

This software is written using C-language and is splitted into two parts — user interface and hardware dependent processes. User interface (*xnir*) is organized as a standard widget–based X11 client program and can be executed under DESQview/X [4] or any Unix X11 environment. Under observer's control, this program starts/stops hardware control processes locally or remotely, establishes connections and execute a dialog with them. The interprocess communication uses the standard stream sockets on the top of TCP/IP protocol. The Xnir is an intrinsically distributed software and can be executed across the INTERNET as well as on the stand alone PC, dedicated to the measurements.

The user has direct access only to the windows provided him by the user interface program, hence we describe here how to use them. There are also several files which have direct impact on the measurement session and we describe them later on.

The Common Elements of the User Interface

The generic window, appearing during the work, can contain several common elements, such as:

- *Button* : Invokes the associated action (starts the jobs, pops-up the sub-menus, etc.). The button border is highlighted when the pointer is placed on it, to indicate that the button is ready for the selection.

To invoke the corresponding action, it is enough to **click** on it the left mouse button. Selected button changes its color for the short time.

- *Label Sub-window* : This window usually contains short text, explaining the meaning of the other accompanied window. This type of the window is used also for the display of the regular messages following the execution of the current task.

- *Text Sub-window* : The main purpose of this type of the window is to permit the user to set the desirable value for some job parameters. There are displayed one or more lines of the text. The window can be or not editable by the user. In the first case a cursor appears when the pointer is inside the window. Editable mode lets the user place the cursor anywhere in the text and modify the text at that position. The text written by the user is automatically stored by the program when the user press the *Return* key or moves the pointer outside the text window. **If the cursor doesn't appear when the pointer is inside the window, it means that this window is not-editable and its behavior is like a label window**. But the parameter value shown in such a window still can be changed by the choice of the suitable item from the *List* window, which in this case can be popped-up using the nearly placed button **List**.

- *Scrollbar* : It is composed of a slider (with vertical or horizontal orientation) with a thumb inside. The scrollbar can be *informative* and in this case the thumb inside the scrollbar can not be moved by the user. Ussually it is used to reflect roughly the performed percent of the task under the execution. In the other situations the *controlled* scrollbars provide the possibility to scan the contents of the accompanied window in the different directions. To move the thumb down (right) the user has to **click** the left mouse button in it. To move the thumb in the opposite directions the user has to use the right mouse button. The middle mouse button can be used to move the thumb in both directions while the button is pressed on it and not released (**dragged**).

- *List Sub-window* : This window displays several lines of the text (usually list of the files, list of the options, etc.) and one of the items can be chosen by the **click** of the left mouse button on it. If the list of the items do not fit into the window size, the controlled scrollbars are provided.

- *Compound List Widget*: This widget is popped-up by use of the button **List**. It contains two buttons, **DISMISS** and **DEFAULT**, and the list sub-window. The list sub-window is used to choose the desirable item from the list. The **DEFAULT** button can be used to return back the default item. The button **DISMISS** destroys this widget. When this type of the widget is popped-up, the pointer is permitted to move only inside of it. The user has to chose the item and/or dismiss this widget to continue the work.

- *Compound Message Widget* : This widget is popped-up by the program when the observer should be informed about something what can influence the session. There is, evidently, the text window with the information and one button, **DISMISS**, which should be used to destroy this widget. The Fig. 1 shows an example of such a message widget.

Figure 1: Widget with an Example of the Message

All the windows, which can appear during the session, are the **standard X11 windows** and can be treated using the **standard Xserver actions**, while it is strongly recommended to close them in the normal situation by use of the provided buttons instead of the killing the window by the Xserver action. Some internal logic can be destroyed in this way sometimes. But in the abnormal situation the Xserver actions can sometimes help.

The File Structure of the Xnir Software

The file structure of the Xnir software is placed on the PC dedicated for the control of the measuring devices (ARNICA, LONGSP), but some of its top elements should exist on the computer executing the user interface program (*xnir*) too. We describe briefly below the main elements of the file structure on the dedicated PC.

The top directory of the **Xnir** software (on the disk D:) is *ir_soft*. It contains the directories:

- *bin* with all executable files (exe and batch). `D:\IR_SOFT\BIN`
- *data* where the fits files, obtained during the measurements, are stored. `D:\IR_SOFT\DATA`
- *resource* with the resource files (files with the list of the available filters, mosaics, etc.). `D:\IR_SOFT\RESOURCE`

There could be also some other directories with the sources or some additional files, but they are related only with software development and maintenance.

Chapter 1

The Work, Step by Step

1.1 How to start Xnir

The only program with which the users has contact is **xnir.exe** (or **xnir** under Unix). The **xnir.exe** can be executed only in the protected mode, hence it needs DESQview/X running on the PC and the access to `go32.exe`. In normal situation *xnir* should be launched by means of a DESQView/X menu. This menu can be popped-up by the click of the right mouse button on the empty part of the screen, or, tapping the *ALT* key. Then the sub-menu *OPEN WINDOW* should be selected (by mouse, or by the direction keys and *RETURN/ENTER* key or by *O* key). The item corresponding *xnir* with desireable option should be selected in this menu.

If *xnir* is started from the **DOS** session, the directory `d:\ir_soft\bin` should be included into the current PATH variable or should be the current directory.

The *xnir* program controls both the infrared instruments: ARNICA (with the polarimeter, too) and LONGSP. It is necessary to specify one of the following options to start *xnir*:

- **-a**[*rnica*] to start ARNICA control program
- **-p**[*olar*] to start ARNICA + polarimeter control program
- **-l**[*ongsp*] to start LONGSP control program

The *xnir* program is an X11 client and can be provided by the standard X11 options. One important example is the **-display** parameter. If this parameter is provided, all the windows will appear on the referenced screen.

To start the program the user has to issue the command

xnir [-[option] -[option] ...]

in the suitable (in sense of the current directory or PATH variable) dos session. When *xnir* is issued without any option it displays the help message.

1.2 The Main Menu

When *xnir* is started, there appears the window with the main menu.

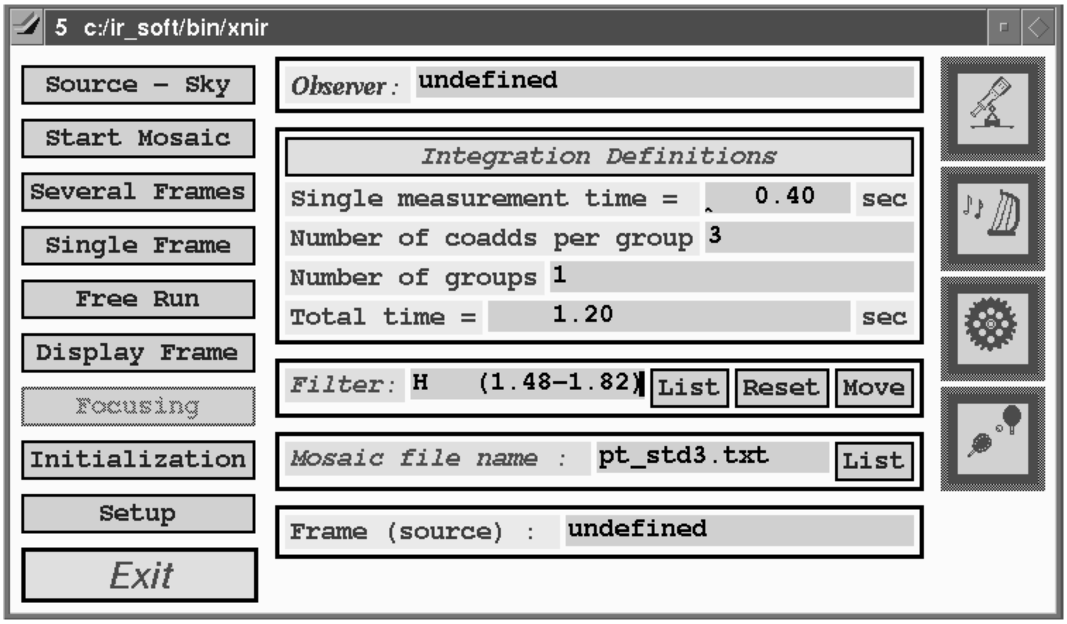

Figure 1.1: Widget with Main menu for ARNICA without the polarimeter

The main menu, provided by *xnir*, looks a little bit different depending on the set of devices controlled, but the general organization is the same. The Fig. 1.1 shows the picture in the situation when the measurement session involves the camera ARNICA without the polarimeter (**-a** option).

The main menu consists of three panels. The left panel (command panel) contains the buttons associated with the different tasks. The buttons can be **sensitive** or **insensitive**, depending on the situation, but the general look of the panel is always the same.

The middle panel (parameter panel) is dedicated for the setting of the user or observational program dependent parameters. It can contain some different sub-windows depending on the set of the devices involved into the measurement session.

The right panel (status panel) reflects the status of the instruments (devices) involved into the measurements (the telescope, the electronics, the motor controller, the Ping-Pong board and, if any, the polarimeter). There are several icons with simple pictures, placed vertically. If the corresponding device is not ready or not available, the icon has a thick red border.

In the very beginning the task *initialization* is always started automatically. This task pops-up its own window, which will be described in the section 1.7.1. Nearly all items in the main menu are in the **insensitive** state until the test of the hardware will be finished. This task initialize the instrument and a wheel to a known state.

1.3 Setting of the Parameters

The parameter panel (the middle one) purpose is to provide for the observer the easy way of the setting and changing of the measurement environment.

The *xnir* normally saves its state when the current task is finished (the current job, not the program). Hence, if for the some reason the observer needs to restart the program, he/she turns back to the previous setting, *if the program is restarted with the same option.*

The panel is splitted into several sub-windows. The top sub-window has the label **Observer:** and the window of the *text* type, where the observer name **should be inserted or changed in the beginning of the session**.

The next sub-window is a little bit more complex. There is a label explaining the contents of whole sub-panel (**Integration Definitions**), and there are also several items with their own labels and the body *text* windows. There are four items:

- **Single measurement time** : it is the elementary integration time (in seconds) of the single measure (or *coadd*);
- **Number of coadds per group** : it is the number of elementary integrations which are averaged and stored as a single fits file;
- **Number of groups** : shows how many times the “group measurement” (described by previous item) is repeated for each telescope position;
- **Total time** : it is the total integration time (in sec) calculated in the

following manner:

$$total\_time = number\_of\_group * number\_of\_coadds * single\_time$$

Only the first three parameters can be defined by the observer, while the total integration time is automatically calculated by the program itself. If the new value of the parameter does not fit into the predefined limits, the nearest limit is set for that parameter.

Below resides the **Filter** sub-panel with the label, not-editable text window, where the current filter description is displayed, and three buttons: **List**, **Reset** and **Move**.

A click on the **List** button pops-up the compound *List* widget (see the page VII), which permits to choose from the provided list of the available filters. The item chosen appears in the body text window of the filter panel, but **no action is executed at ones**. To **really** move the filter to the new position it is necessary to click at the button **Move** or to start a measurement. The button **Reset** moves the filter forcely to the origin, because any filter setting is *relative*. The filter movement from an *unknown* position has no sense and can be a reason of the incorrect observations.

Important !

- The observer should not worry about the real filter movement. The *xnir* resets the filter **automatically** at the beginning of the session and then **moves it to the position defined in the filter panel** before the start of any real measurement (if it was changed, of course). The observer responsibility is **only to set the correct filter**.

The buttons **Reset** and **Move** are provided because sometimes there is the necessity to execute the filter movements *without* the execution of the measurements.

Mosaic. It is a nickname of the special mode of the measurements (see page 12 and section 2.4). This mode permits the execution of a series of the measurements of the source and of the nearby sky to evaluate and later subtract spatial variations of detector responsivity and background fluctuations. The sequence of the positions to be measured is written in the *mosaic file* (see section 2.4). For a complete discussion the observer **should** see [3]. In this sub-panel, the observer can select the mosaic file from the list popped-up by use of the **List** button. The current mosaic file name is written inside the text window. The list provided is the contents of the sub-directory **mosaic**

in the directory **resource**. There is a lot of different mosaic files there. If the observer needs his/her own file to execute, it should be placed into this directory. **Note.** The *nirtest* uses the same files but **they are in different directory**.

The last, common for any option, sub-panel is the **Frame (source)** panel. The name of the source to be observed is written in this window. This information is proveded to *xnir* by the program that controls the telescope, if that program provides it. On the contrary it is a user duty to insert into the text window the corresponding information.

If *xnir* is started with the options **-p** or **-l**, there appear the additional sub-panels. We will describe them in the sections specially denoted to that cases.

The items in this panel are sensitive to the user intervention not always, but only when the action can not disturb the current job under the execution. The buttons usually become shady when they are insensitive. There is no way to see if the text window is sensitive or not, only to try. But the panel is set insensitive as a *whole*, so just look at the state of the buttons inside of it.

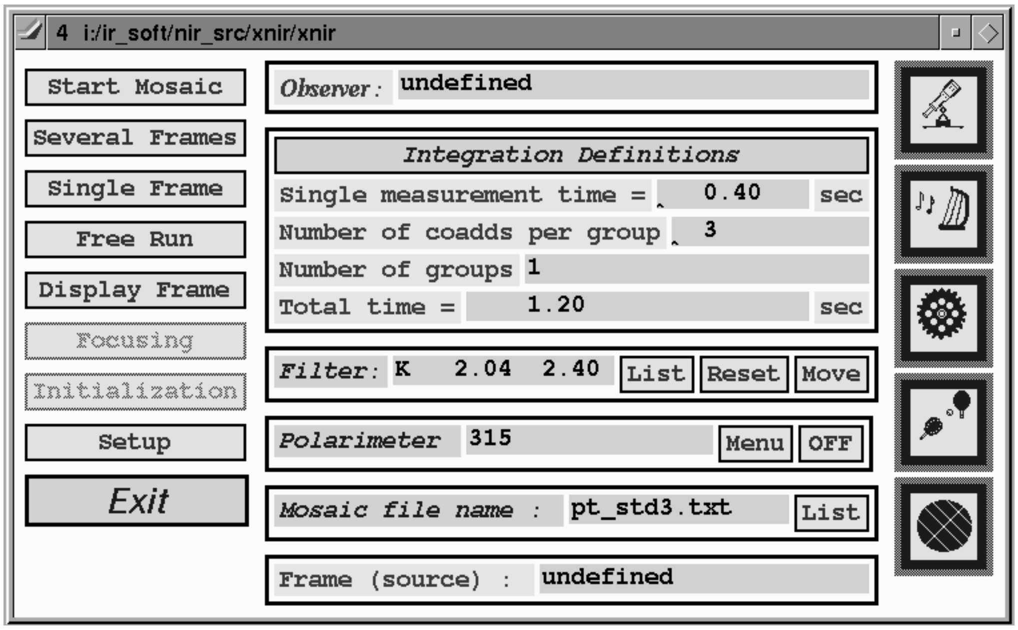

Figure 1.2: Widget with Main menu for ARNICA with the Polarimeter

1.4 Additional Parameters for the Polarimeter

When *xnir* is started with the option **-p**, inside of the parameter panel appears the sub-panel for the *Polarimeter*. In this case the main menu looks like it is shown in the Fig. 1.2. Please note also the additional icon for the polarimeter in the status panel. The additional sub-panel has, as usually, the label *Polarimeter*, the text window (not-editable), where the current angle of the polarimeter in degrees is shown and two command buttons: **Menu** and **ON/OFF**.

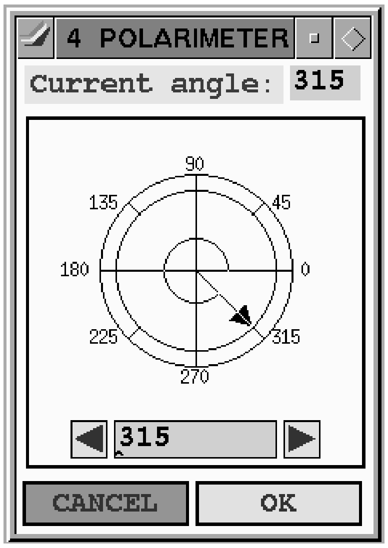

Figure 1.3: Widget for the Selection of the Polarimeter Angle

- **Menu** : When you click on this button, the sub-widget for the selection of the polarimeter angle is popped-up (Fig. 1.3). There is the current angle value shown in the top line (it is *the current position of the polarimeter* and *does not change during the manipulations in this widget*). Below, one finds the whole ring of the polarimeter positions with the current one marked by the arrow and numerical value. This picture is accompanied by the compound widget with two buttons and the text window between them where the current value of the angle is repeated again.

  The choice of the angle value can be done in several different modes:

- writing the exact value inside the text window of that compound widget;
- using the left (<) or right (>) buttons to change the angle value by one degree in the corresponding direction;
- clicking with the pointer inside the ring picture at the proper place.

The position of the arrow in the picture and the value in the text window below it should change simultaneously,not depending on the way the user uses for the choice of the new angle value.

The command buttons **CANCEL** and **OK** let the user reject or accept the new chosen angle value and close this widget. When the *Menu* widget is closed, the new chosen (if **OK** button was used) angle value appear in the text window of the *Polarimeter* sub-panel.

Important !

- The real polarimeter movement is not executed when the new angle value is chosen. You just have changed the angle which *will be used for the subsequent operations, not a real position of the polarimeter*. There is still the possibility to change your choice.

• **ON/OFF** button: When the user clicks at this button, the program *runs* the motor process to put the polarimeter **in** or **out** of the camera optical beam. It is important to note, that when the button label is *OFF*, it means that the polarimeter *is not inserted in the instrument optical beam*, not the action to remove it from there. If the user clicks at the button, the label changes to *ON* and the program *insert* the polarimeter in the optical beam and sets the chosen angle. So the button label represents the **polarimeter status**.

Important !

• The rotation of the polarimeter is done in an automatic mode by the program when the user starts any acquisition job. It is the program itself that has to compare the current polarimeter angle with the value stored inside the text window. If these two values are different, the polarimeter rotation is started. If the user changes the polarimeter angle when the command label is *OFF*, it is the program itself that, before rotating polarimeter, inserts it in the optical beam. So, the observer is not obliged to use the button **ON/OFF**, even **it is recommended do not use it**.

1.5 Additional Parameters for the LONGSP

When *xnir* is started with the option **-l**, inside of the parameter panel appear two sub-panels for the *Diaphragm* and the *Grating*. The main menu looks like it is shown in the Fig. 1.4.

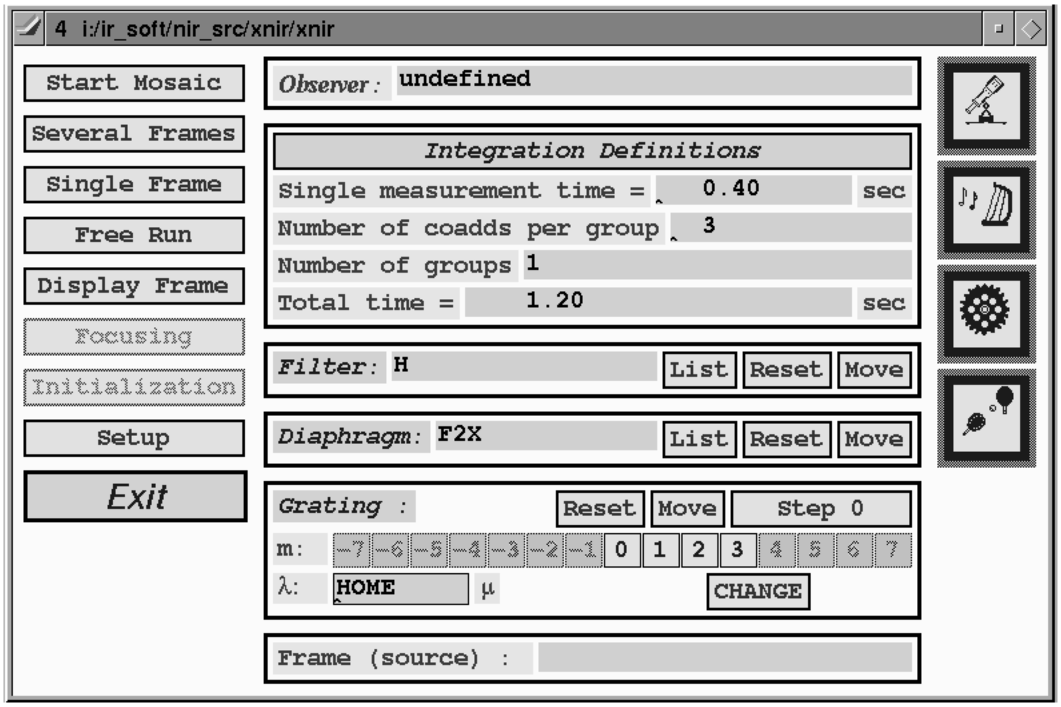

Figure 1.4: Widget with Main menu for LONGSP

- **Diaphragm** : this sub-panel has the same organization as the filter sub-panel. Everything concerning the filter choice is completely applied to the diaphragm choice.
- **Grating** : this sub-panel is more complex. The grating can be characterized in two different ways — by the single value of the *step* or by two parameters *order (m)* and *wave length (λ)*.

Starting from the top there are the next three commands: **Reset**, **Move** and **Step**. The first two run the motor process to reset or move the grating wheel. The third one has in the label additional number corresponding to the current position of the grating wheel in *steps* and, when it is clicked on, pops-up the menu for the direct setting of the position in the steps(Fig. 1.6). This menu will be described in details later.

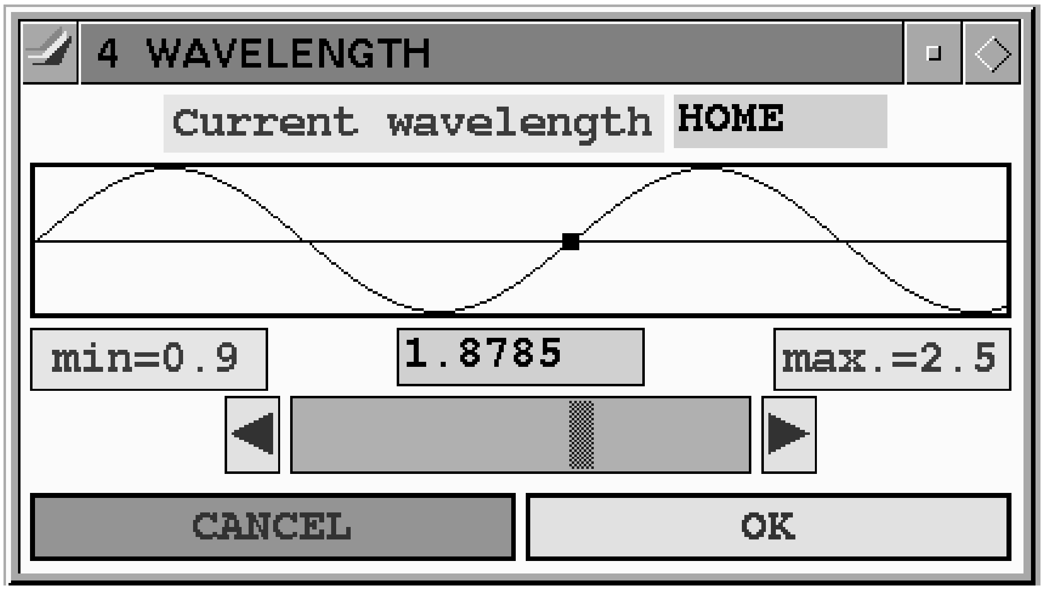

Figure 1.5: Widget for the Selection of the Wavelength for Grating

Below there are two sub-panels:

- **m (order)** : the grating order can be easily selected by the click on the button whose label corresponds to the order value. The selected button will be highlighted. While there is the long set (from -7 to +7) of the orders shown, only the orders from 0 to 3 are available for the observer normally. The others are used in test mode by the developers of this device. The buttons corresponding to the values can not be selected are shown in different colors (gray).
- λ **(wavelength)** : There are the label, text window with the value, the label with units and the button **CHANGE**. The **CHANGE** button lets the user to select the working wave-

length. The additional menu (Fig. 1.5) is popped-up. The wavelength can be chosen by:

* writing the value inside the text window;
* clicking on the left (<) or right (>) scroll buttons that decrement (increment) by 0.0001 μm the wavelength value;
* moving the thumb inside the scrollbar with the mouse;
* clicking at the proper place inside the wave picture window with the left mouse button.

The length of scrollbar corresponds to the wavelength range of the detector (0.9 μm — 2.5 μm). The command buttons **CANCEL** and **OK** let the user to reject or accept the last chosen value. The label at the top of the menu window shows the current grating wavelength value. The position of the grating wheel in the steps is calculated using the order and the wavelength value chosen by the user.

The grating position can be changed also by the direct setting of the position in the steps. The menu (Fig. 1.6) includes two pictures: the left one shows all the grating positions (including the HOME and the MIRROR) while the right one shows only a zoom of the operative area. To change the grating wheel position the user can:

- write the value inside the text window;
- use the left (<) or right (>) scrollbar buttons to change the position by **one step**;
- click the left mouse button at the chosen point inside the right picture window [1];

The menu includes also the button whose label changes from **Absolute** to **Incremental** when we click on it. When the button shows the *Absolute* label, the position value chosen by the user is interpreted as the absolute grating position (i.e. **relative to the home position**). When the button label is *Incremental* the value written inside the text window is interpreted as the number of the steps used to move the wheel from the current position. The pictures always show the absolute positions. The program calculates automatically the wavelength value from the position in the steps chosen by

[1] the left one is not active when the program works in the *user mode*

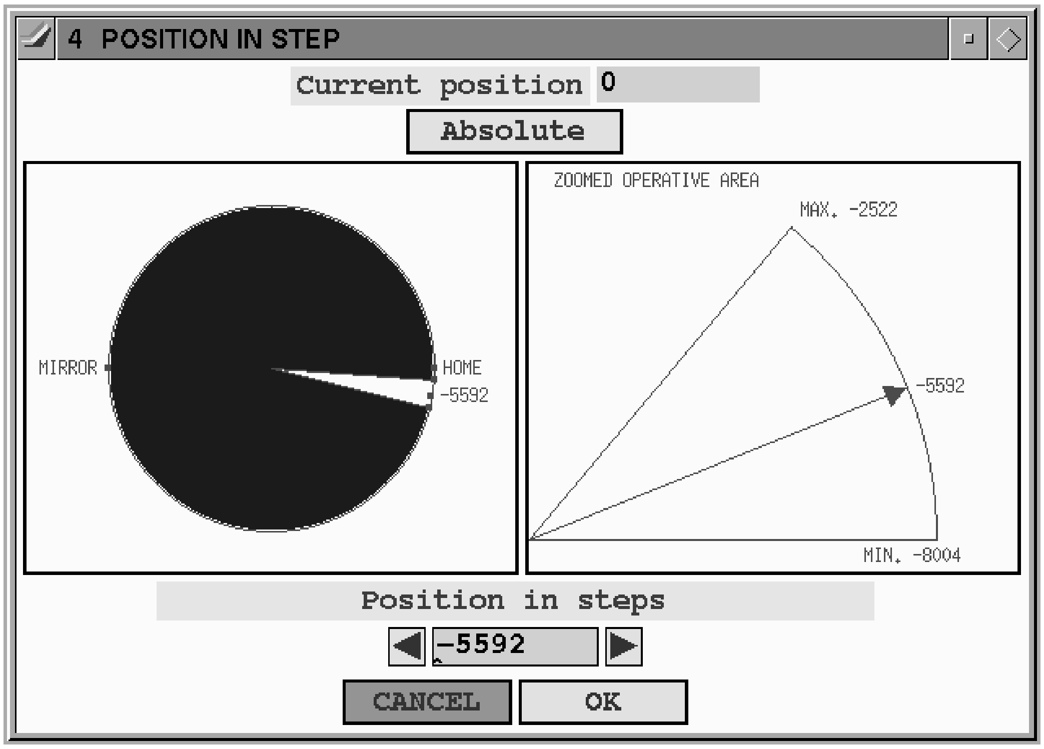

Figure 1.6: Widget for the Selection of the Grating Position in the Steps

the user. This calculation is done using a fixed value of the order. When the currently chosen order does not correspond to any reasonable solution, it is changed.

The buttons **CANCEL** and **OK** have the evident meaning, as it was described above for the other sub-menus.

Important !

- As for the filter, the polarimeter and the diaphragm, the real movement of the device is **not** executed when the observer sets the new values in this panel. The buttons **Reset** and **Move** should be used to place the device into the chosen position really. But, as for the other devices, the movement is executed in an automatic mode by the program when the user starts any acquisition job. It is the program itself that has to compare the current grating position with the chosen values. If these two values are different, the grating rotation is executed. **The observer is recommended to leave the control to the program.**

- Not all the combinations of the setting are valid, hence the corresponding control is executed and the observer is warned, when the parameter affected by his current manipulation request the change of the other parameters.

1.6 The Buttons of the Command Panel

The buttons in the left panel of the main menu are the same for the all options, ARNICA or LONGSP.

There are the buttons which invoke the execution of the measurements themself, providing four different ways to do that:

- **Source - Sky** : This procedure could be useful for the check of the correctness of the source positioning and chosen acquisition parameters. The simple *Mosaic* acquisition (see below) is executed as is described in standard mosaic file *checkfr.txt*. Normally first the source is measured, then the sky and then again the source, to ensure the correct source positioning. The last two frames are automatically subtracted and the result is shown using SAOimage.

- **Start Mosaic** : This procedure takes as many frames as the number of the groups for each telescope position specified in the mosaic file chosen (prepared) by the user. Each frame is the mean of the fundamental integrations (coadds) in the group. Each frame is saved as a single fits file. After each exposure the telescope position is changed accordingly to the mosaic file (Section 2.4).

- **Several Frames** : This procedure takes as many frames as the group number value. Each frame is the mean of the coadds in each group. The telescope position remains the same during the all time of the measurements.

- **Single Frame** : This is the elementary measurement, from which all the others tasks are composed. The only one exposure is taken with the integration time equal to the *single integration time* value.

- **Free Run** : This task executes continuously single frame exposures. The data taken are not stored on the hard-disk, but the results can be displayed on the screen. Can be used for the focusing or for the quick investigation of the area around the source.

The button **Display Frame** pops-up the menu widget, which permits to the user to display the frames taken during the observations (Fig. 1.7).

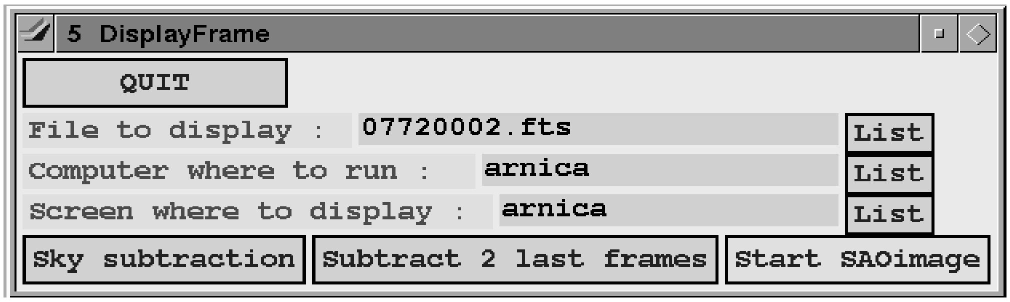

Figure 1.7: Menu, popped-up by the Display Frame button

This window contains the next three items:

- *File to display* : the user can select from a list menu activated with the **List** button the image file to display. The program automatically writes inside the not-editable text window the name of the last fits file stored on the disk.

The following options are normally used for the remote observing/consulting:

- *Computer where to run* : the user can select inside the list the name of the computer across the network, where to start the display program. If the computer, where the data are stored, and the computer, where the display program is started, are different, the program transfers the fits files with ftp from one machine to the other. In this case the user is asked to give the username and the password of the account assigned on that machine. If the user has an own account on that machine, he/she can give his/her username and password. The user can also specify the directory where the transferred data are to be stored. The internet address can be inserted into the text window directly or can be chosen from the predefined list.

- *Screen where to display* : the user can choose from the list the computer screen where the image has to be shown. The responsibility to provide the access permissions is on the observer, if the computer where to run and the computer where to display are different.

Currently, the only image viewer in use is SAOimage, which is started by the corresponding button. There are two other useful buttons. The button **Sky Subtraction** permits to choose two different frames from the stored files to be used as a "source" and a "sky" frames, and subtract them placing the result into the "source" file. This file can be visualized by **Start SAOimage**.

The button **Subtract 2 last Frames** automatically executes this complete procedure using as a "source" and a "sky" two last aqcised frames.

The button **Focusing** is not activated for the moment since this part is yet in the elaboration process..

The button **Initialization** starts the instrument initialization procedure. For a deeper description see the section 1.7.1.

The button **Setup** pops-up the *Setup* menu, which permits the user to set some variables of the session environment. See the section 2.3.

1.7 The Informative Widgets of the Processes

The **Xnir** software is divided in several independent but interconnecting processes. The observer works with the user interface process only, which starts and stops the other task oriented processes, or automatically or at the user request.

There are three such task oriented processes:

- **Initialization**
- **Acquisition**
- **Motor Control**

The user interface provides the special window for each process started, where some information about the task is displayed. The name of the process the window belongs to (Acquisition, Hardware Initialization, Motor Control) is on top bar. These windows contain the fields, where the informative messages are written from time to time, the scrollbars to show percentage done of the current sub-task, and some command buttons.

1.7.1 Initialization Process

This process is started automatically when the *xnir* program is executed for the first time. It could be started by the observer too, if necessary. The aim of this process is to initialize all the instrument devices, including the telescope colloquium.

This process looks first for the presence of the telescope connection, it executes the reset and the programming of the instrument electronics, then it programs the motor controller and, finally, it executes the test on the ping-pong board. During these operations the user is continuously informed how the procedure is going on.

The example of the informative window for this process is shown in Fig. 1.8.

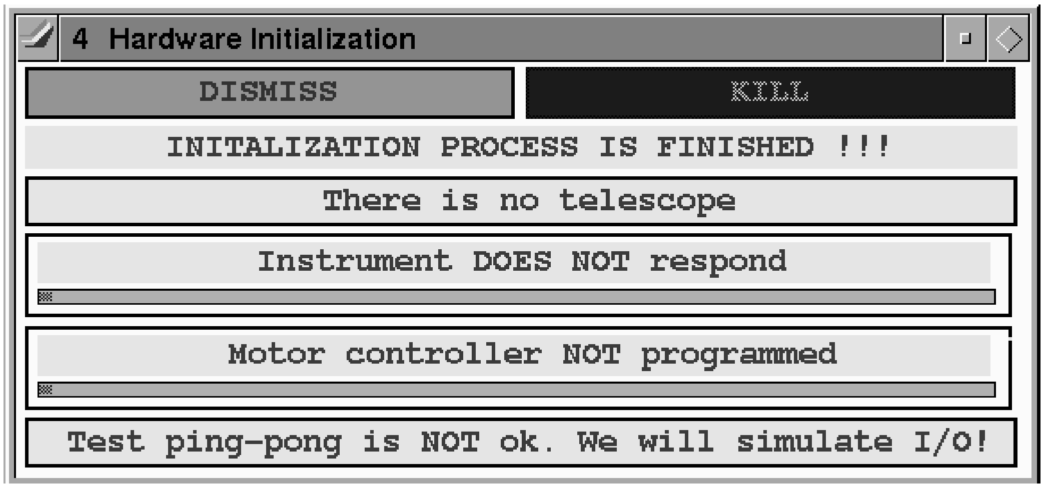

Figure 1.8: The informative Widget for the Initialization Process

When the process starts, the corresponding window appears on the screen. This window has been organized in several label windows, where the messages for the user are printed. Look the Chapter 4 to have the detailed list of the messages. When some tests are executed successfully or not, the corresponding icon in the status panel of the main menu changes its status.

The window includes also two command buttons: **DISMISS** and **KILL**. With help of the first one the observer can destroy the informative widget, when it is no more useful, to free the screen. The second one permits to terminate the corresponding process forcely, to kill it. When this button is clicked, the user is asked to confirm his choice.

1.7.2 Motor Process

This process executes each movement of the motor wheel and it is started either by the user when he clicks on the **RESET** or **MOVE** buttons[2] or by the program in an automatic mode when it is necessary because of the difference between the current and the required motor position. In this case it is started before the start of the acquisition process.

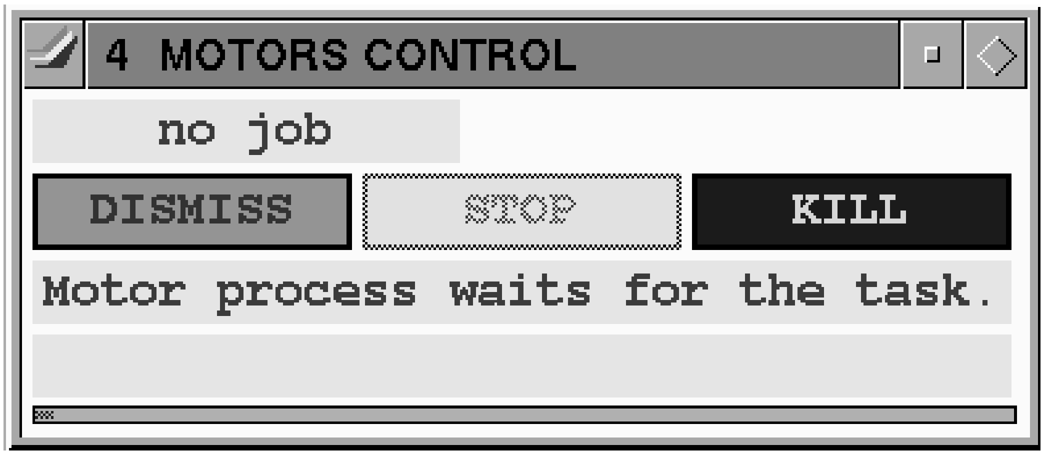

Figure 1.9: The informative Widget for the Motor Process

Since the instrument can have more than one optical element (as in the case of the polarimeter and grating), the informative task window (Fig. 1.9) displays in the top label window which motor is addressed (for example Filter, Diaphragm, etc.) and which kind of the movement has been started. There are also from the top to the bottom:

- three command buttons: **DISMISS**, **STOP**, **KILL**;
- two label windows where some informative messages appear;
- a scrollbar that lets the user follow the moving of the wheel;

The meaning of the buttons is the same for the all widgets. The new one, not described so far, is the button **STOP**. Its action issue the request to the process to stop the current job, which will be stopped in this case nicely, i.e. in the proper time, when this action will not damage the environment too much. The process itself remains alive. There is a difference between

[2]The button **ON/OFF** (polarimeter sub-panel) starts the motor process too

this button action and the action of the button $\boxed{\textbf{KILL}}$. The last one kills the process without any safety consideration.

There are two general types of the motor movements:

1. **Motor wheel reset.** Any position of the motor wheel is the position in the steps relative to the origin, the home position. The *reset* is an action to set the wheel at the *home* position. The wheel reset is an operation the user has to do from time to time during the normal work. In fact, during the positioning of the motor, there can be some steps lost or added, hence there can be some difference between the real and declared position in steps of the motor, even if the velocity of the wheel movement is chosen to minimize that. An accumulated error can in principle damage the measurement quality. The reset is executed automatically by the program when the *xnir* program is started or when the movement of the motor was previously stopped by the user with the help of the $\boxed{\textbf{STOP}}$ button. **In reality, the program executes not the simple reset, but always after the reset, moves the wheel into the predefined position (the position corresponding to the current choice of the filter, diaphragm, etc.).**

2. **Motor wheel movement.** The *move* action moves the motor wheel from the current position to that one chosen by the user. When the user has selected the position of the optical element, the motor will execute the real movement, if the user clicks on the $\boxed{\textbf{MOVE}}$ button. If the user do not execute this operation, the program itself, before the start of the acquisition process, verifies all the motor positions. If the current position of one (or several) motor is not equal to the selected one, the movement process is started automatically. The motion can be stopped by the user by use of the $\boxed{\textbf{STOP}}$ button. In this case, before moving again the motor, a wise choice will be to execute a reset of the wheel (see above).

1.7.3 Acquisition Process

This process is started when one of the first four buttons ($\boxed{\textbf{Start Mosaic}}$, $\boxed{\textbf{Several Frames}}$, $\boxed{\textbf{Single frame}}$ or $\boxed{\textbf{Free Run}}$) is pressed by the user. When the acquisition process is started, the corresponding informative window appears on the screen (Fig. 1.10).

This window is organized in a similar way as the Motor Process one. In the label window at the left top side it is written which kind of the task is

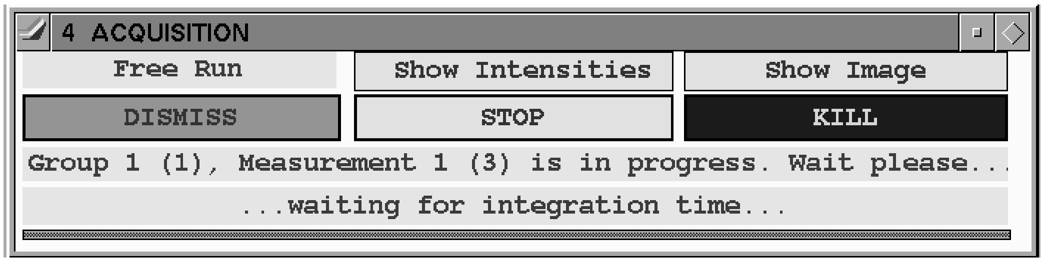

Figure 1.10: The informative Widget for the Acquisition Process

started by the user (i.e. Mosaic, Several Frames, Single Frame or Free Run) while in the other label windows below, appear the current messages for the user. An informative scrollbar shows the user the percent of the sub-task done at the moment. Above the usual three command buttons **DISMISS**, **STOP** and **KILL**, there are two useful buttons which permit the observer to control the quality of the measurements.

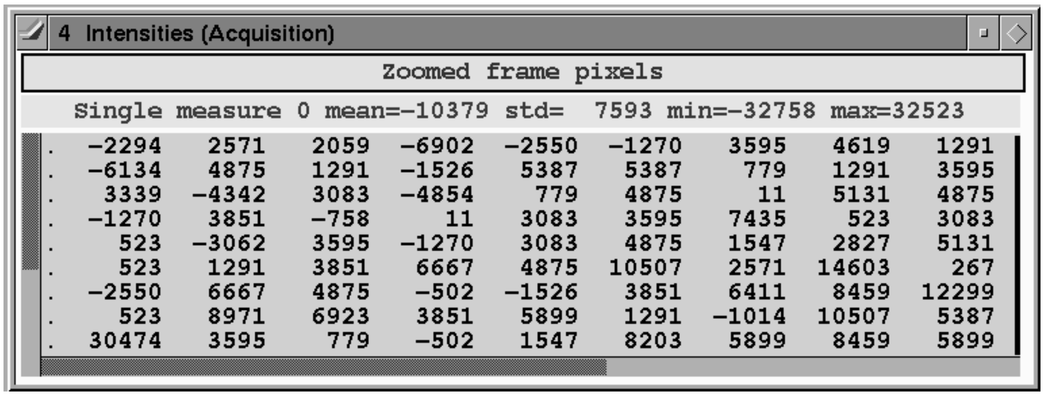

Figure 1.11: The Widget with the measured Intensities

- **Show Intensities** : when this button is pressed, a secondary window (marked with *Intensities (Acquisition)* label) is popped-up on the screen (Fig. 1.11). The button label changes from **Show Intensities** to **Dismiss Intensities** and click on it again closes this secondary window.

  In the opened window there are shown the numerical values of some pixels of the CCD array. If the label of the button at the top of this window is *Evenly distributed pixels* the intensities of the pixels taken at any eights rows and columns are shown. The observer can click on this button and the label and the choice of the intensities will be changed. The button label becomes *Zoomed frame pixels* and the intensities of the pixels inside the 16x16 square zoomed area are shown. How to choose this zoom area one should see below in the description of the action of the next button.

- **Show Image** : this command opens a secondary window (marked with the label *Image (Acquisition)*) in which the image obtained during the observation is shown. The button label changes from **Show Image** to **Dismiss Image**. To close the *Image* widget the user has to click on this button again. The *Image* widget is compound one (Fig. 1.12).

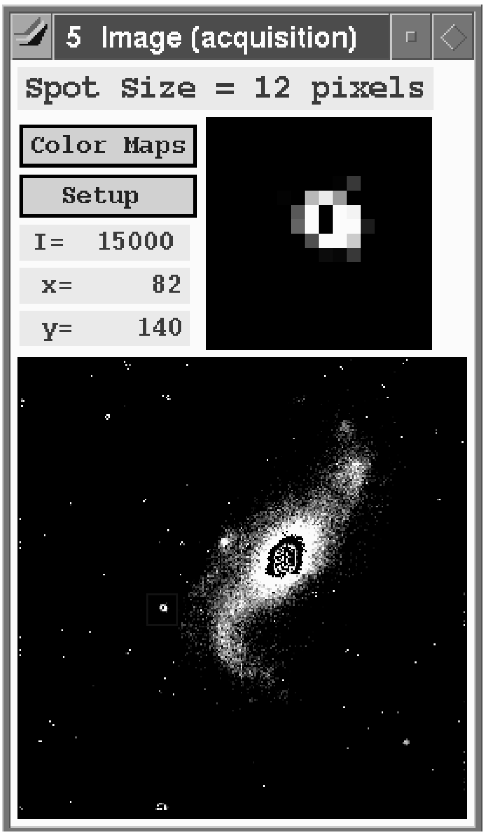

Figure 1.12: The Widget with the acquired Image

The whole image or its part, shown inside of the 256x256 square at the bottom of this widget, is obtained by the *linear map* of the predefined interval of the measured intensities into the color index range (from 0 to 127). The user can change the interval of the intensities. It permits to look at the image using different contrast and resolution. **One should remember, that any change of the predefined environment becomes in charge only from the next image and later**. The already displayed image is affected only by the change of the **Color Maps**. This button permits to choose one of the predefined color maps from the list. At the right top side of the widget the zoomed image of the 16x16 pixels square is shown. The *center of this box* can be chosen by the *click on the left mouse button* at the chosen place in the whole image below. The position of the pointer is taken as the square center and is shown in *x=* and *y=* labels at the left side of the zoomed image. The *original* (i.e. not mapped) intensity found in this pixel is reported in the label *I=*. The zoomed area is shown in the core image as a square box. The top label shows the *Spot Size*, which could be useful during the focussing process. This value is calculated as a number of the pixels above the predefined level relatively to the maximum intensity found in the *zoom* area. When center of the zoomed area is changed, whole 16x16 square is checked for the maximum intensity, then the smaller square, 7x7 pixels centered for the found maximum, is analyzed. The discriminative level is the same as for the *semi–automatic focusing* procedure and, while that procedure is not used yet, the level can be set using *Setup Focusing* sub-menu. The spot size 49 pixels means that the intensity is practically flat in the zoomed area. The observer should remember, that to have this value it is necessary to have the image first.

The image visualized inside the bottom window is transmitted through the interprocess communication or even across the network. This can be a heavy work for the program, so the user can choose not to send all the image but only a part of it. The image region can be selected using the right mouse button. The right mouse button should be **pressed at one corner of the desirable area and then dragged to the opposite corner**. There appears a changing rectangular box which shows the selected area. The selection is finished when the button is **released**. It is also possible to configure the program not to send the image at all. This choice could be done using the menu popped-up by the button **Setup** (Fig. 1.13).

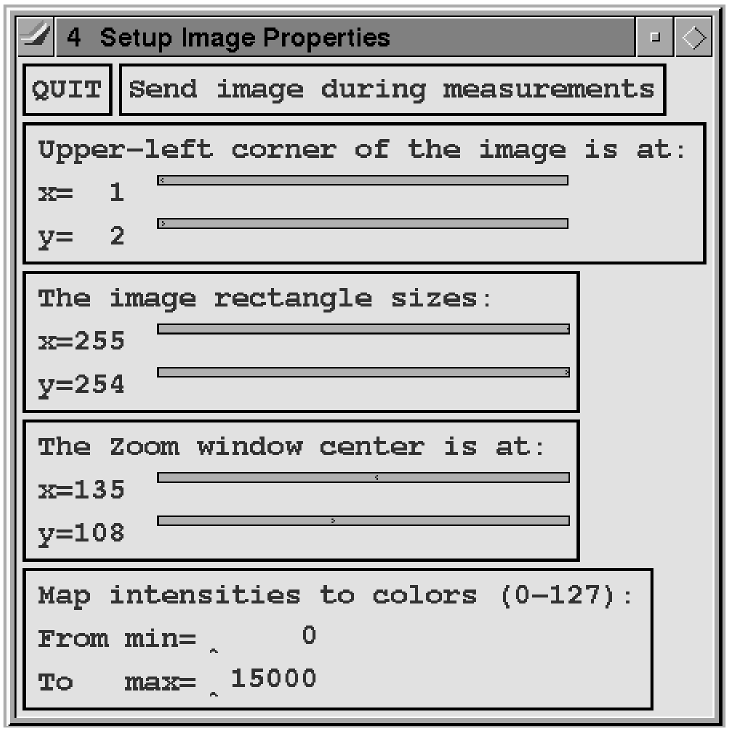

Figure 1.13: The Widget for the Setup of the Image Properties

This menu permits also to set all the image properties discussed above. To change the choice *to send/not to send* the image from the acquisition process to the user interface just click on the corresponding button. To change the area to send, one has to use the provided scrollbars. To change the interval of the mapping of the intensities, one has to edit the text window with the shown limits (*min=* and *max=*). This menu is also accessible from the general setup menu (**Setup** button in the main menu).

To increase the performance of the measurements we never transmit the image when the Image widget is not opened, even if the setting in charge is to send it.

Note.
It is possible to reduce the image size to 1 pixel by the incorrect setup. In this case it will be hardly visible. It can happen, for example, if the position of the left/upper corner is shifted toward the right/low corner and then returned back again. Obviously, during the first part of the

procedure, the possible image size is forcely reduced, but **the observer has himself** *change it back to the desirable values.*

1.8 Protocol

During each acquisition job the program automatically creates or updates a text file named `protocol.acq` and stores it in the `d:\ir_soft\data` directory. This file contains the principal information about the measurements done by the user. There are:

- the date and time of the observation;
- the name of the observer;
- the hosts from which the program and the hardware are controlled;
- the sequential measurement number and the source name;
- the name of the fits file;
- the airmass value;
- the focus value;
- the filter used;
- the number of coadds and the single integration time;
- the mosaic name (if the mosaic task was used);

Below we report an example of this file

```
--------------------------------------------------------------------------------
 Tue Sep  3 13:42:54 1996     Observer(s):   user1
--------------------------------------------------------------------------------
 Observations at:  arnica     Controlled from:  arnica
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------


   1 source     03300001  Air: 0.00  Foc:    0  Filt: K    3x0.40   sec
```

```
2 source     03300002  Air: 0.00  Foc:    0  Filt: K    3x0.40   sec pt_std3
3 source     03300003  Air: 0.00  Foc:    0  Filt: K    3x0.40   sec pt_std3
4 source     03300004  Air: 0.00  Foc:    0  Filt: K    3x0.40   sec pt_std3
5 source     03300005  Air: 0.00  Foc:    0  Filt: K    3x0.40   sec pt_std3
6 source     03300006  Air: 0.00  Foc:    0  Filt: K    3x0.40   sec pt_std3
```

This file is a plain ascii file and can be easily printed by the user with print command. Any comment can be added to this file using the usual text editors. The user should respect the file share rules while adding comments during the active *xnir* session. It is better to restrict this activity to the time **between** the execution of the acquisition tasks.

Chapter 2

The Setup of the Session Environment

As for any complicated enough software, the Xnir session depends on the setting of many environment variables.

The most important part of them, which are subject to change with each task execution, is accessible from the main menu in the parameter panel and is described in the previous chapter. There are other parameters which also have an impact on the session execution, but are of less frequent change or are reserved to maintenace. There are also the parameters which do not influance directly the measurements or the session execution but can change completely the look of the menus (the fonts, labels, etc.)

There are several ways to change the session parameters. There is the standard data base of the X11 environment. The fonts, colors, menu labels and even some internal parameters are organized as the X11 resourses and are retrived from that data base during the initialization of the session. All the parameters have the default values installed if the data base is empty. While the X11 data base can be very complicated, usually the user preferences are placed into the application resource file in the *root* user directory and they have the priority above all other levels of the resource data base. Of course, those parameters, provided as the parameters in the issued command line, have the priority above any resources defined in a static data base. In Unix environment the Xnir resource file is named **Xnir** according the rules of X11. Under DESQView/X there is the special directory where the system looks for the resource files, `C:\DVX\APP-DEFA` , and the Xnir resource file name is **xnir.res**. This file will be described shortly in the next section.

As we have remarked already, *xnir* saves its state after each sub-task finished, hence there is a file, namely **xnir.ini**, with the set of the saved parameters. This file is placed in the **software root directory**. Normally it is `d:\ir_soft`. All the parameters found in this file have the final priority in the initialization part of the session. We describe this file below too.

Some parameters can be still changed during the session. There is the **Setup** command button which provides the general access to the *Setup* of different session features. Sometimes the access to setup sub-menus is available from the feature-dependent menus, as it was already described for the *Image* widget of the *Acquisition process* (see the page 21). The observer should remember, that the new setting will not be saved at ones.

2.1 The Description of the Resource File xnir.res

The resource file can contain the description of *any* X11 resource. Below we discuss shortly the contents of an example of this file provided for *xnir* by default.

Warning !

- **The user is not recommended to touch this file without real necessity. Its maintenence implys the presence of corresponding knowledge of X11 interconnections. The uncarefull parameter setting can produce very strange effects, which can lead to complete impossibility of correct usage of this X11 client program. Moreever, there are also the parameters specific for this application and maintened by the developers only. Everybody who does not follow this recomendation does it for his/her own responsibility.**

The contents of the file `C:\DVX\APP-DEFA\XNIR.RES` :

```
*ObserverLabel.font:      *times-bold-i-*-*-140-*-*-*-*-iso8859-1
*ObserverName.font:       *times-bold-i-*-*-180-*-*-*-*-iso8859-1
*Exit.font:           *helvetica-bold-o-*-*-180-*-*-*-*-iso8859-1
*Command*font:          *courier-bold-r-*-*-140-*-*-*-*-iso8859-1
*IntegrationLabel.font: *courier-bold-o-*-*-140-*-*-*-*-iso8859-1
*GratingLabel.font:     *courier-bold-o-*-*-140-*-*-*-*-iso8859-1
*FilterLabel.font:      *courier-bold-o-*-*-140-*-*-*-*-iso8859-1
*DiaphragmLabel.font:   *courier-bold-o-*-*-140-*-*-*-*-iso8859-1
*PolarLabel.font:       *courier-bold-o-*-*-140-*-*-*-*-iso8859-1
```

```
*MosaicLabel.font:      *courier-bold-o-*-*-140-*-*-*-*-iso8859-1
*ImageFbut.Command.font:*courier-bold-r-*-*-100-*-*-*-*-iso8859-1
*ImageFbut.Label.font:  *courier-bold-r-*-*-100-*-*-*-*-iso8859-1
*Command*ListForm.Command.font:*times-bold-r-*-*-120-*-*-*-*-iso8859-1
*Command*ListField.font:*courier-bold-r-*-*-120-*-*-*-*-iso8859-1
*WaveLabel.font:-Adobe-Symbol-Medium-R-Normal-*-140-**-*-*-fontspecific
*WaveUnit.font: -Adobe-Symbol-Medium-R-Normal-*-140-**-*-*-fontspecific

*font:                  *courier-bold-r-*-*-140-*-*-*-*-iso8859-1

*Exit.background:     PeachPuff
*Exit.foreground:     red
*Command*background:  aquamarine
*Command*foreground:  MidnightBlue
*Toggle.background:   aquamarine
*Toggle.foreground:   red

*IntegrationLabel.background: PeachPuff
*IntegrationLabel.foreground: red
*FilterLabel.background:      PeachPuff
*FilterLabel.foreground:      red
*DiaphragmLabel.background:   PeachPuff
*DiaphragmLabel.foreground:   red
*GratingLabel.background:     PeachPuff
*GratingLabel.foreground:     red
*PolarLabel.background:       PeachPuff
*PolarLabel.foreground:       red
*MosaicLabel.background:      PeachPuff
*MosaicLabel.foreground:      red

*Label*background: MistyRose
*Label*foreground: DarkGreen
*Text*background:  PaleGreen

*ExitLabel.font:    *helvetica-bold-r-*-*-140-*-*-*-*-iso8859-1
*ExitLabel.foreground:    MidnightBlue
*ExitForm.background:     red
*ExitForm.foreground:     MidnightBlue
*EXIT.background:         LightBlue
```

```
*EXIT.foreground:         red
*CANCEL.background:       green
*CANCEL.foreground:       MidnightBlue

*Command*ListForm.background:            green
*Command*ListForm.Dismiss.background:  aquamarine
*Command*ListForm.Dismiss.foreground:  MidnightBlue
*Command*ListForm.Default.background:  yellow
*Command*ListForm.Default.foreground:  MidnightBlue
*Command*ListField.foreground:         MidnightBlue
*Command*ListField.background:         PeachPuff

*Command*DisplayForm.background:       MistyRose
*Command*DisplayForm.Quit.background:  aquamarine
*Command*DisplayForm.Quit.foreground:  MidnightBlue
*Command*DisplayForm.Label*background: PeachPuff
*Command*DisplayForm.Label*foreground: red
*Command*DisplayForm*Text*background:  PaleGreen
*Command*DisplayForm*Text*foreground:  black
*Command*DisplayForm.Start SAOimage.background: yellow
*Command*DisplayForm.Start SAOimage.foreground: MidnightBlue

!for HwInitWidget
*KILL.background:              red
*KILL.foreground:              LightBlue
*DISMISS.background:           green
*DISMISS.foreground:           MidnightBlue
*Scrollbar*background:         yellow
*Scrollbar*foreground:         red
*Form*Scrollbar*background:    yellow
*Form*Scrollbar*foreground:    red

*integrationNumberLabel: Number of coadds per group
*integrationGroupsLabel: Number of groups

*IntegrationTimeValue:     0.40
*IntegrationNumberValue:     3
*IntegrationGroupsValue:     1
```

```
*FilterName: H   1.50  1.80
!*MosaicName: mosaic2.txt

*imageSend: 1
*imageXsize: 256
*imageYsize: 256
*imageX0: 0
*imageY0: 0
*imageMin: 0
*imageMax: 15000

*SocketBufferSize:    1040

*FocusStep: 6
*FocusSum: 3
*FocusLevel: .5

*BadPixelLevel:      32000
*SaturationLevel:    31000
*SaturationWarning:  257

*TelescopeName: VATT
*TelescopeAddress: 255.255.255.000
*TelescopePort: 1043

!*hardwareHost: arnica.unibe.ch
!*test: TRUE
!*internalAccount: arnica
!*rootDirectory: i:\\ir_soft
*hardwareRootDirectory:  d:\\ir_soft
*gTestMode: FALSE
```

The resource file can contain the **comments**, started by the exclamation mark (!) and the empty lines. The **wildcharacters** are allowed. If there are several statements concerning the same parameter in the competion (it happens frequently, because the parameters can be addressed as belonging to the group), that one with a **less general** address or the last one takes the priority.

The statement should contain three fields — the resource name, the character ":" and the resource value optionaly after one or several spaces.

First group of the statements, one can see above, is used for definition of the the fonts used for the different fields inside of the widgets created by the application. If such preferences are not provided or are not found by X11 server, the default setting of the X server is used. There are the precisely aimed statements, having the specific field as a target, as well as the statement of the most general type appliable to all the fonts used by the application. The observer who likes to use his own fonts should remember that the field sizes are usually initially predefined and some fonts could not fit into the available space or could be too tiny. The widget size generally can be tuned manually after the visualization by the use of the X server actions, hence no explicit size resources are provided.

The next group of the statements controls the colors of the fields.

There are also two statements re-defining the *contents* of two labels in the parameter panel of the main menu. So, it is an example what the user can do if he/she is not satisfied by the labels of the panels or the buttons used.

Then follows the initial setting of the parameters for the elementary integration (see the page 3). Then the filter to be used by default is provided. The same statement for the mosaic file is currently commented.

The next group defines whether the image should be provided during the measurement (1 means yes, 0 — no), the size of the segment of the image to show (in pixels, 1-256), the position of the left upper corner of the segment and the interval of the pixel values to be mapped into the color index range (0-127).

The statement `*SocketBufferSize` has as a value the optimized size of the buffer used for the image (and not only the image) transmition.

Important !

- **We strongly recommend do not touch this parameter. It is dangerous. One has to understand well not only the xnir program internal properties but the operating system and network organization before to try to manipulate this parameter.**

Next three statements are for the tuning of the focusing process. The semi-automatic focusing is yet not supported by this version, but one of parameters, *FocusLevel* is used. It defines the discrimination level as a part of the intensity maximum used for the calculation of the spot (star image) size.

Three next statements can be useful to change for the user. The program *xnir* provides the dynamic control of the presence of the bad and saturated pixels inside of the 16x16 zoomed square (see the page 20). If there are the bad pixels in that area, or the number of the pixels equal and above the saturation level is not less than the value provided in the *SaturationWarning* item, the corresponding *Warning message* (see an example Fig. 1) is popped-up any time the array acquisition is done. These statements provide the value limits when the pixel is regarded as bad or saturated. The current value of the *SaturationWarning* is above the total number of the pixels in the area to check (256), hence with such setting the warning about the saturation will not be issued at all.

There are three statements defining the telescope in use. Currently only three telescopes (TIRGO, NOT and VATT) are supported. The connection to the telsecope can be via the sequential port or across the network. In the last case to establish connection one needs to provide the internet address and the port on the server computer. The setting of all these values is the responsibility of the maintenance personal.

The last group of the statements, commented or not, concerns the internal organization of the *xnir* program or is aimed to define the special modes of the execution. The values should be set by the person responsible for the installation of the software.

Important !

- **The user is recommended do not touch these parameters, if he/she does not like to be in troubles**.

The observer is welcome to use his/her own preferences during the own sessions but is kindly asked to respect all our warnings and the preferences of other persons. So, the best way is to have the copy of this file for the personal use and **to restore the default file everytime when it was replaced**. We recommend too to look from time to time into the manual and the default resource, to be sure that your personal resource file corresponds to the current situation in that part, which is of the developer's responsibility. The correct behavior will be just to copy into your pesonal file without any change all the statements you do not know or do not understand completely their meaning.

2.2 The Description of the File with Saved Configuration xnir.ini

The file `xnir.ini` is created and updated by the program automatically, hence in the reality the user should not touch this file even if it is possible to edit it. Anyway it is useful to know what part of the environment is saved in and restored from this file. It can help to understand the situation after the re-start of the program. Below there is an example of this file:

```
device longsp#
observer undefined#
clock_time 3.0000000
int_time 0.4000000
int_number 3
groups_number 1
temp_disk e:#
filter K   #
diaphragm F2X  2 pixels Slit #
grating_order 2
grating_lambda 1.4582164
grating_position -3948
polar_on_axis 0
polar_angle 0
polar_inser 3300
polar_linea 32000
frame_name undefined#
meas_num 2
mosaic pt_std3.txt#
colormap gray scale#
arnica_speed 1000
longsp_speed 90
grating_speed 1000
image_x0 0
image_y0 0
image_nx 256
image_ny 256
zoom_centerx 128
zoom_centery 128
image_min 0
image_max 15000
image_status 1
saturation 31000
bad 32000
sat_warning 257
```

Many of the statements are self-explanatory, if the user have read this manual carefully before. Lot of them were discussed in the previous section, and have here just a little bit different appearance. The general statement consist of two fields: the parameter name and its value, with a space between. If the value is a string of characters, it is terminated by the special character "#" , which does not belong to the resulting parameter value.

The first line of this file declare what kind of device was supported when this file was created, ARNICA or LONGSP. It is referenced first of all and, if the current device for which the program was started, is different, the contents of this file is **skipped**.

There are some parameters not referenced before, but the best choise the user can make is **do not touch them, as well as this file as a whole**, so we will not describe them.

2.3 The Setup Menus

As we have remarked already above, some parameters can be changed also during the session. Using the button **Setup** from main menu the observer can pop-up the root menu of the setup tree (Fig. 2.1).

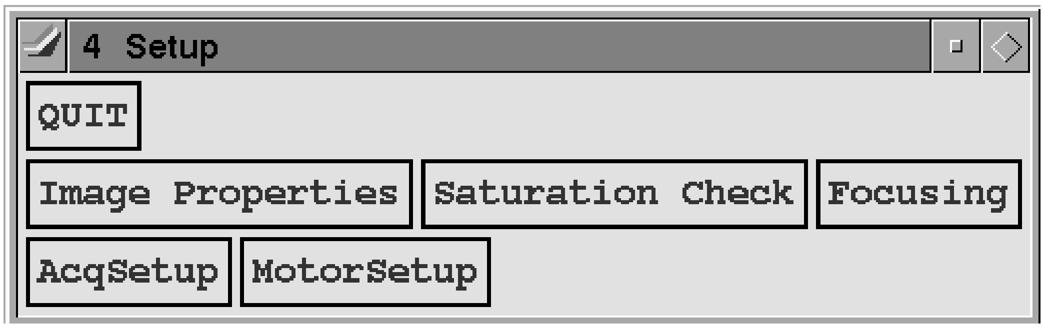

Figure 2.1: The Widget with Root Setup Menu

This menu contains several buttons which offer the possibility to go farther to the specific feature setup menus. Currently, of the user interest there are only the *Image Properties* setup, which is already described above (page 21) and the *Saturation Check* setup (Fig. 2.2).

The parameters accessible from this menu are already described too (page 29).

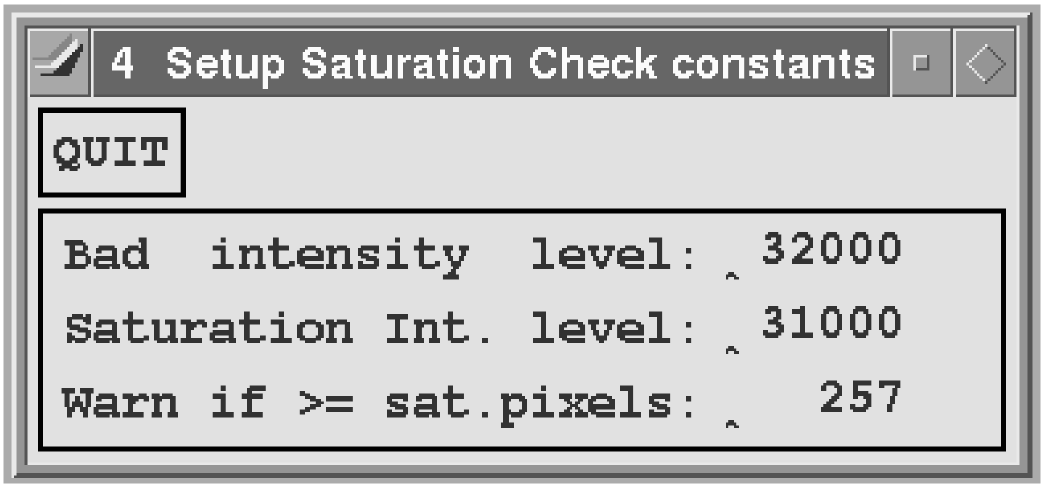

Figure 2.2: The Widget with Saturation Check Setup Menu

2.4 The Mosaic File

While the *Mosaic* file can not be strictly speaking regarded as a set of the parameters, it could be concerned as an important part of the environment in which the *xnir* program is working and which should be provided or chosen basically by the observer. Here we provide an example of one standard *Mosaic* file with a short description of its contents. Below you see the contents of `pt_std.txt` file:

```
  0.0   0.0 %  posizione "0"
-10.   10.  %  posizione "1"
 80.    0.  %  posizione "2"
  0.  -60.  %  posizione "3"
-80.    0.  %  posizione "4"
```

Its structure is rather simple. The text after “%” sign is a comment. There are 5 positions defined in this file, where the telescope should be placed **relatively** the position at which the telescope was directed **before** the start of the *mosaic* measurement task. Two coordinates, right ascention α and declination δ (measured in arcseconds), define the relative displacement to be performed for the next acquisition.

The specific mosaic file for each observetional program should be prepared in the same manner and placed into the directory
`D:\IR_SOFT\RESOURCE\MOSAIC`.

Only in this case it will be available during the session of *xnir*. It can be done at any moment, the program looks for the directory contents when the corresponding **List** button is activated.

2.5 The Filter File

As we have already mentioned before, when the **List** button inside the **Filter** sub-panel is pressed, the list of the filters appears. This list is read by the program from a text file that has to be stored in the directory `D:\IR_SOFT\RESOURCE`. Since the **xnir** program supports both the instruments ARNICA and LONGSP, two different filter files exist. The one used for ARNICA instrument is named `arnica.fil`, while the other used for LONGSP is named `longsp.fil`. Below there is an example of the `longsp.fil` file:

```
*******FILTERS**************************************
Filter name    Filter position
****************************************************
D                2844  Plug
J                948   (1.12-1.40) Standard Filter J
H                4740  (1.48-1.82) Standard Filter H
K                6636  (2.00-2.41) Standard Filter K
F1               0     Hole
EPS              3792  PinHole for Test
SJF              1896  (0.91-1.12) Band Pass Filter
FHK              5688  (1.76-2.12) Band Pass Filter
```

The first three lines constitute the file header and **they have to be always present** in the file. The list of filters and positions has to be written **starting from the fourth line**. The first column of this file contains the name of the filter (plug and holes too) mounted inside the instrument, while the second column contains the corresponding filters positions in steps (relative to the 0 position). It is not necessary that the filters are ordered on the basis of their position. They can be written in any order. The third column contains a comment with a general description of each filter. In the example it shows the filter wavelenght range and its complete name. While the presence of the first two columns is obviously indispensable, the content of the third column is not necessary for the correct execution of the program, but it can be useful for the user. The format of the two files `longsp.fil` and `arnica.fil` is the same, the difference is only relative to the list of filters and their positions.

Important !

- **The user is strongly asked not to touch these files. The modification of these files is required only when one or more filters are changed inside the instrument. For this reason it is a duty of the instrument staff to mantain update these files. The accidental absence of these files causes the program to create a file with a default list of filters whose corrispondence with the instrument filters configuration is not guaranted. In this case the user is warned about the absence of the file and that a default one is used.**

2.6 The Diaphragm File

The file, containing the list of available diaphrams, is named `longsp.dia` and, as for the filter files, is stored in the same directory `D:\IR_SOFT\RESOURCE`.

Below there is an example of the `longsp.dia` file:

```
*******DIAPHRAGMS***********************
Diaphr. name   Diaphr. position
***************************************
  EPS          6636  PinHole for Test
  F1X          2844  1 pixel Slit
  F2X          4740  2 pixels Slit
  PIN          948   PinHole for Test
  F4X          0     4 pixels Slit
  F1           5688  Hole
  WIN          3792  Field Stop
  F2           1896  Hole
```

This file has the same organization as the filter file. The description provided in the session 2.5 is valid for this case too.

2.7 The Grating File

This file holds the coefficients of the polynomial of second degree used to calculate the grating position once the order and the wavelength values are chosen by the user. The coefficient values of the calibration polynomial are determined with test measures. This file is stored in the directory `D:\IR_SOFT\RESOURCE` and it is named `grating.res`. Below there is an example of the `grating.res` file:

```
-9552
1994.0000000
-8.3900000
```

The three coefficients are written on different lines: the first value starting from the top is the coefficient of zero degree, the second is the coefficient of first degree and the last is the coeffiecient of second degree.

Important !

- **The user is asked not to touch this file. It is a duty of the instrument staff to mantain update this file. The accidental absence of this file causes the program to create a file with a default coefficient values whose corrispondence with the actual grating configuration is not guaranted. In this case the user is warned about the absence of the file and the usage of a default one.**

2.8 The Filter-Wavelength File

This file holds the list of the filters with the corresponding wavelenth range. This file is used only when the program controls the longsp instrument. Its name is `l_filter.wav` and it is stored in the directory `D:\IR_SOFT\RESOURCE`. Below we show an example of this file

```
 /* Filter wavelenght range      */
 Filter name     Wavelength range (micron)
****************************************
  J                   1.12-1.40
  H                   1.48-1.82
  K                   2.00-2.41
  SJF                 0.91-1.12
  FHK                 1.76-2.12
```

The first three lines constitute the file header and **they have to be always present** in the file. The list of filters and the corresponding wavelength range are to be written **starting from the fourth line**. The first column of this file contains the names of the filters mounted inside the instrument, the second column contains the corresponding filter wavelength range in μm.

Important !

- **The user is asked not to touch this file. The modification of these files is required only when one or more filters are changed inside the instrument. For this reason it is a duty of the instrument staff to mantain update this file. The accidental absence of this file causes the program to create a file with a default list of filters and corresponding wavelength range whose correspondence with the instrument filters configuration is not guaranted.**

2.9 The SAC file

The programming of the motor controller is done using an ascii file whose name is `sac_prog.txt`, stored in the directory `D:\IR_SOFT\RESOURCE`. This file contains all the motor parameters such as speed, acceleration, time delay etc., and the ICL language programs that executes the motors movements. Below we report the contents of this file.

```
del SETUP       ^ delete the old programs
del SETUP1
del SETUP2
del ARNFILT
del FILT_SET
del DIAF_SET
del GRAT_SET
del RESETP
del RESETM
del RESET
edit SETUP1     ^ setup for filter in ARNICA program
ENTER A  500
ENTER B  100
ENTER H 2000
ENTER M 1000
```

```
ENTER D  500
ENTER Q   50   ^ time delay required by the drive to return to full power
ENTER R   50
ENTER S   50
```

←

```
edit ARNFILT   ^ call the setup for filters in ARNICA program
CALL SETUP1
QUIT
```

←

```
edit SETUP2    ^ setup for the grating
ENTER A  500
ENTER B  100
ENTER H 5000
ENTER M 1000
ENTER D  500
ENTER Q   50   ^ time delay required by the drive to return to full power
ENTER R   50
```

←

```
edit SETUP     ^ set the system an movement variable of
ENTER A  500   ^ the filter and diaphram motor
ENTER B   30
ENTER H 5000
ENTER M   90
ENTER D  500
ENTER Q   50   ^ time delay required by the drive to return to full power
ENTER R   50
```

←

```
edit FILT_SET  ^ activate the output relative to the filter
CALL SETUP     ^ motor
SET 1          ^ set the output corresponding to the filter motor
WAIT 20        ^ wait 1/5  second for external device to recognize
QUIT
```

←

```
edit DIAF_SET  ^ activate the output relative to the diaphram
CALL SETUP     ^ motor
SET 2          ^ set the output corresponding to the diaphragm motor
WAIT 20        ^ wait half second for external device to recognize
QUIT
```

←

```
edit GRAT_SET  ^ activate the output relative to the grating
CALL SETUP2    ^ motor
SET 1 2        ^ to activate the grating motor each output (1 and 2)
WAIT 20
QUIT
```

←

```
edit OFF       ^ turn off motors
RESET 1 2       ^ turn off output 1 e 2
WAIT 10        ^ wait one tenth of second
QUIT
```

←

```
edit RESETP    ^ do the reset of the wheel,in the positive
EN P 0         ^ direction
+ SLEW         ^ move in positive direction
CALL RESET
QUIT
```

←

```
edit RESETM    ^ do the reset of the wheel,in the negative direction
EN P 0         ^ the parameter P is set at zero
- SLEW         ^ the motor move in negative direction
CALL RESET
QUIT
```

←

```
edit RESET
(A) UNTIL 11 A ^ until input 11 becomes active
STOP           ^ stop the motor when input 11 is active
MOVE 0         ^ wait until a motion is complete
EN LD 00100000
WAIT 20
- SLEW
(B) UNTIL 11 B
STOP
MOVE 0
EN LD 00000000
WAIT 20
+ RUN          ^ move at base speed in the positive direction
(C) UNTIL 11 C ^ until input 11 becomes
STOP           ^ stop the motor when input 11 is active
REPORT         ^ wait until the last motion is complete
REPORT         ^ twice to make check of 'movement end' the same
QUIT
```

←

Important !

- **The user is asked not to touch this file. It is a duty of the instrument staff or of the software developers to mantain update this file.**

Chapter 3

Tirgo CDR Handling

3.1 Data Structure

All data taken with Tirgo Nicmos3 instruments (ARNICA and LONGSP) are stored directly as standard *fits* files at 16 bits integer format.

Every file has an unique name formed from the last 4 digits of the integer part of the current Julian date and from 4 digit of a sequencial number. The acquisition program never rewrites over an old file. If the program is forced to send data to an existing file, it increments automatically the sequencial number until it get a not already used filename. Sequencial number is protected from the casual user, and should be changed **only** for the special purposes. There is the one line of the description, associated with each file and accessible by the *dir* command under DOS or DESQView/X. Under other systems one can find this description in the file `lista.txt`. The comment contains the source name, integration time, and, if telescope is connected, also the source coordinates with a limited precision. Below you can see an example of the report of a *dir* command (the data structure is created by *nirtest* program) :

```
 Volume in drive D is ULTERIORE      Serial number is 3713:19EE
 Directory of  d:\arnica\data\*.*

95130049.fts   135360   6-10-94  16:50 NGC4590 K -4.0s 12.39.30 -26.45.1
95130050.fts   135360   6-10-94  16:50 NGC4590#00 K -4.0s 12.39.30 -26.45.1
95130051.fts   135360   6-10-94  16:51 NGC4590#05 K -4.0s 12.39.30 -26.45.1
95130052.fts   135360   6-10-94  16:51 NGC4590#07 K -4.0s 12.39.30 -26.45.1
95130053.fts   135360   6-10-94  16:51 NGC4590#08 K -4.0s 12.39.30 -26.45.1
```

```
   1,218,708 bytes in 5 file(s)         1,257,472 bytes allocated
 120,770,560 bytes free
```

We have chosen to include in the fits header all relevant information available to the system, as source name, telescope position and so on. That choise is intended as an help in the reduction phase and as the backbone of the forthcoming Tirgo data archive. We give here an example.

```
SIMPLE  =                    T
BITPIX  =                   16
NAXIS   =                    2
NAXIS1  =                  256
NAXIS2  =                  256
ORIGIN  = 'T I R G O'             / TIRGO Observatory
BSCALE  =        1.0000000E+00    / real_value = tape_value * BSCALE + BZERO
BZERO   =        0.0000000E+00    / real_value = tape_value * BSCALE + BZERO
BUNIT   = 'COUNTS'                / Each count corresponds to  19  electrons
OBJECT  = 'NGC4590'               / Identifier of source
DATE-OBS= ' 10/06/94'             / UT date of data acquisition
TIME-OBS= ' 15:49:55'             / UT Time of data acquisition
J-DAY   =        9.5131597E+03    / Julia n date of acquisition minus 2440000
DATE    = ' 10/06/94'             / UT date this file was written
NUMBER  =                   49    / Incremental acquisition number
INSTRUME= 'ARNICA    '            / Device who measured data
OBSERVER= '    Carlo_Baffa'       / Observer name
FILTER  = ' K 1200 '              / Filter name and wheel position
TIME-INT=        1.0000000E+00    / Integration time on detector  1.0 * 4
COADDS  =                    4    / Number of frames added togheter
RA      = '12.39.30,0'            / Right ascension of source
DEC     = '-26.45.10'             / Declination     of source
EPOCH   =        1.9500000E+03    / Epoch of coordinates
AIRMASS =        2.3000000E+00    / Airmass when acquisition started
DATAMIN =        9.1000000E+01    / minimum value in array
DATAMAX =        9.9990000E+03    / maximum value in array
DOUBLESA= 'ON '                   / Double sampling enabled
END
```

3.2 Data Handling during the Measurements

Data are stored in a working directory. Due to limitation of DOS, the more files are present there, the slower is the processing of a new file. After having collected a between 50 to 100 data files, observer should use a 4Dos procedure ((*savedata*) to move data files to a different location. For *xnir* this procedure is started normally automatically. The command *savedata* has no arguments. If the acquisition program is running inside DESQView/X, this batch can be started from a normal dos window, precede (only one time) by the command *cnir*.

Data will be moved in a directory belonging to a tree of directories which permits to separate easily data of different dates, as in the followin table. Each execution of the *savedata* creates a new subdirectory in the present day directory. The present version limit the number of subdirectory of each day to 14.

```
D:.
|
+---ARNICA
    |
    +---DATA
        +---SAVE
            +---94_JUN
                |
                +---JUN_14
                |   +---1
                |   +---2
                |
                +---JUN_18
                    +---1
                    +---2
                    +---3
```

3.3 Permanent Data Archiving on CDR

At the end of each night, **data must be moved to the permanent CD Writable disk (CDR)**. Observer must place the current CDR cartridge in the PC and run the procedure *trandata* which will automaticaly copy the data to the cartridge. **We strongly suggest the observer to execute**

this procedure at the end of each night. The save of the data will take about 20 minutes.

The batch *trandata* must be executed **outside** of DESQView/X environment. During the execution, trandata displays the information about the quantity of data to be transferred. If CDR free space is low, data should go to the next cartridge. The screen should appare as:

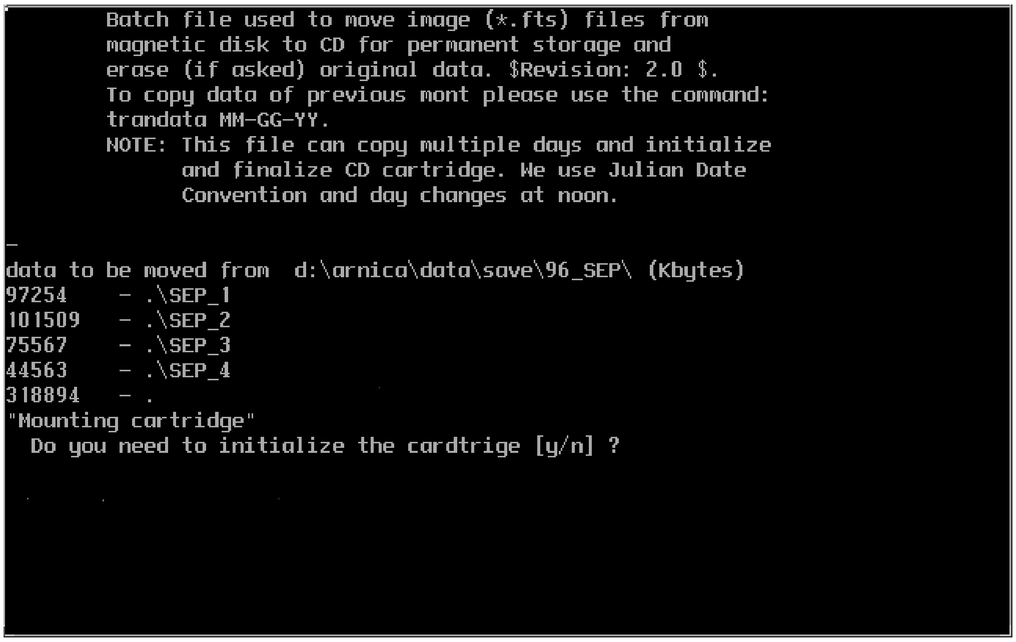

Figure 3.1: Starting messages of TRANDATA batch

The program then asks if there is need to initialize the cartridge. The initialization procedure is complitely automatized and take about 5 minutes.

Important !

- **Do not re-initialize a disk!**
- **Do not initialize a cartridge for a later use without writing some data on it: it is probable that computer will not mount it again!**

The cartridge is now initialized and mounted for writing. The program displays now a day based summary of the current month data. The data already transfered (even if partially) are marked by the label *moved*. The user can select the day or the days to be copied by the use of the button

space and can start execution with **enter**. In the following screenshot days SEP_1 and SEP_2 have been already processed and the day SEP_3 is selected to be copied.

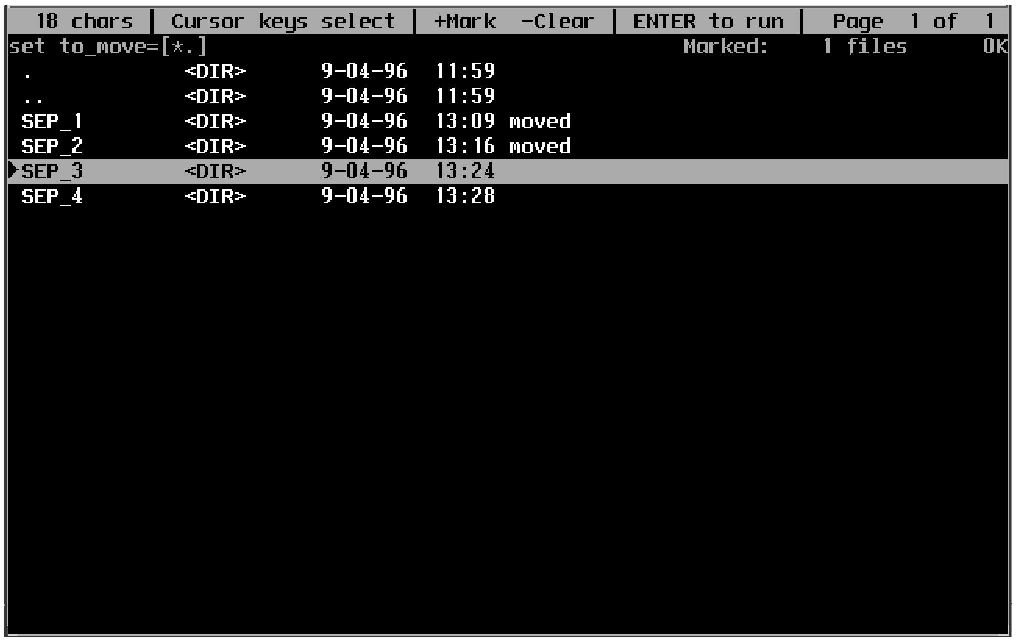

Figure 3.2: An Example of TRANDATA menu

At the end of the copy, which takes about 20 minutes for each day transferred, the procedure asks if the copy has been done properly and, whether it has to delete data from the hard disk. Too many data sitting on hard disk will slow down data acquisition till to stop, so it is observer's cure do not exeed the efficiency limit. Also the observer can choose to hold on the disk all his/her data and to dump all at once on the own cartridge. Remember to execute **alwais** the Tirgo own archive save! Note also, that CDR disks written by Tirgo computer can be read in the other system only after being finalized.

If the free space on cartridge is low, the program propose to finalize the cartdridge. The process will take a few minutes. The finalizing process will convert the CDR cartridge to the ISO-9660 format. The CDR can now be read in all standard CDROM reader by nearly all brand of computer. The finalizing process also *permanently write protects the cartridge.*

If the copy is interrupted, *trandata* can be restarted again and will not overwrite data already transferred. However, if the cartridge has been damaged, the program will warn the user. Normally in this case, it is necessary to mount manually the CDR by means of the program *c:/util/cdrex/cdrex*, using it in the interactive mode.

In case of trouble, **please contact** Carlo Baffa (*baffa@arcetri.astro.it*).

We mantain an archive of all data taken with our instruments. Data are stored in a series of CD Rom cartridges each containing from about 400 to 600 Mega bytes of data. Each cartridge contains the data of a few days of the observations, and has a progressive label which identify the sequence as in DATA_9A and DATA_9B (the two sides of an old worm cartridge) or DATA_26 (in the new CDR format). It is a wise idea to annotate on the ufficial logbook the name of the cartridge in use.

Data can be transfered to the tirgo sun (tirgo or toutatis) and from there saved on a DAT cartridge (borrow your own!). Small amount of data can be tranfered to the home instituton by ftp; remeber that data must be transferred in bin mode. Observer can also borrow a blank unformatted CDR and fill it with the data by means of trandata procedure.

Chapter 4

Messages

4.1 Initialization process messages

Below we report the list of the messages visualized during the initialization process. There are the regular messages with the information how the initialization proceeds. Other messages can appear in the case of some troubles. For such situations we suggest the actions.

The regular messages are usually self explanatory. Some are placed in the corresponding labels for a long time. Some appear only for a short time to be replaced then by others and are sometimes practically invisible for the user.

- **Hardware initialization process started**
- **Instrument initialization started**
- **Checking for the telescope presence ...**
- **There is NO telescope** when the telescope is assumed to be connected to the sequantial port.
- **There is NO telescope connection** in the case when the telescope is accessed through the network.
- **Telescope is OFF**: the test on telescope presence is not OK.
- **Mouse present on COM1, no telescope handling**: this message appears if the serial mouse is found on the port through which the computer should communicate with the telescope. This message should

not appear at all, because the computers that control hardware process are already equipped with bus mouse to avoid this case.

- **Telescope is present on COM1**: the test for presence of telescope is OK.
- **Telescope is OK**
- **Electronics initialization waits to start ...**
- **Standard instrument initialization**
- **INSTRUMENT: starting initialization**
- **INSTRUMENT—>Sending reset...wait**: the program started the electronic software reset.
- **Instrument DOES NOT respond**: the electronic initialization has failed or the program is executed with the test flag on.
- **Instrument respond: reset done**: the electronic has been initialized with a software reset.
- the next short leaving messages appear during the programming of the standard waveforms: **delay pattern**, **reset part 1**, **reset part 2**, **reset part 5**, **reset part 8**, **read part 3**, **read part 4**, **read part 6**, **read part 7**, **read part 9**, **Double reset and double read: standard mode**
- **Instrument is OK**
- **Initialization of polarimeter**
- **Time out with polarimeter**: polarimeter does not respond.
- **Polarimeter initialized**
- **Motor initialisation waits to start**
- **MOTOR CONTROLLER programming started**: start the programming of SAC. **Longsp SAC programming /Arnica SAC programming**: specifies which instrument controller is initialized.

- **Motor cotroller DOES NOT respond** It happens when the electronics is off,or the controller is not power up, or the electronics needs a software reset.

- **File sac_prog.txt does not exist**: the text file with the program that controls motors is not found. Look in the `resource` directory to see if this file is present.

- **Motor controller programmed**

- **Motor controller NOT programmed**: it appears when the program is started with the option *s* or (it means that we don't want to reprogram the sac) or when the electronic is off, or when the program runs in test mode.

- **Not enough memory**: there is not enough memory to allocate for the array where are going to be stored the data.

- **PING-PONG BOARD TEST: wait...**: the program starts the test on the ping-pong board.

- **PING-PONG TEST—-> Ping-pong board not Ready!**

- **Test ping-pong is NOT ok. We will simulate I/O**

- **Test ping-pong ok**

- **Initialization process is finished**: the process is terminated and the relative window can be destroyed using the DISMISS button.

There are some important messages which can appear in *DOS* window where *xnir* was started:

- **xhwinit failed to start! Problem with bs_socket**: this message appear if there are the problems in starting the initialization process from the server side. The only solution is to restart *xnir*. If the problem is not solved it is necessary to reload hardware PC. It is assumed also that the DVX environment is set correctly.

- **xhwinit failed to start! Problem with binding**: this message appears if there are the problems in starting the initialization process from the server side. The actions are the same as above.

- **xhwinit failed to start! Problem with client process: error or timeout on basic socket**: there are the problems in starting the initialization process on the server side. The actions are the same as above.
- **Connection is broken? pending >= 501** Appear when the inter-process communication is broken. May happen if the network traffic is havy. Kill the current acqusition job and start it again. May be necessery to kill by hands the hanged up processes in hardware PC.
- **Error in starting process**: the are some problems in the starting of the the process. This message appears when the file with the list of all the processes running on the computer can't be opened. If you are working under DVX, control the *.dvp* file using the *Customize Menu* voice from the DVX menu. The name of the process in the first label named *Program name* **has to be written in lower case letter**[1].

4.2 Motor process messages

These messages can appear in *DOS* window where *xnir* was started:

- **Motor process failed to start! Problem with bs_socket**
- **Motor process failed to start! Problem with binding**
- **Motor process failed to start! Problem with client process:error or timeout on basic socket**
- **Connection is broken? pending >= 501**
- **Error in starting process**: the are some problems in the starting of the process. This message is sent when the file with the list of all the processes running on the computer can't be opened. If you are working under DVX control the *.dvp* file using the *Customize Menu* voice from the DVX menu. The name of the process in the first label named *Program name* **has to be written in lower case letter**[2].
- **No selection corresponds to number -> #**: there are some problems in sending the number corresponding to the motor to move.)

[1] i.e.: it should appear *xhwinit* and not *Xhwinit* or *XHWINIT*
[2] i.e.:it should appear *x_motors* and not *X_motors* or *X_MOTORS*

- **Motor is already in this position**: the final and initial motor position are the same, so the motor is not moved.
- **Motor process stopped by user. Motor is in the same position**: the motor process is stopped before the starting of the movement.
- **Motor stopped by user**: the motor is stopped by the user during its movement.
- **Motor timeout–>motor stopped**: the movement is stopped because there is a timeout.
- **Motor timeout. Start INITIALIZATION**: if there is a timeout, the user is informed to re-start the initialization process.

The following messages can appear in the informative labels of the Motor process:

- **Filter/Diaphragm/Grating motor moving. Wait!**:one of these motors is moving.
- **Resetting filter/diaphragm/grating. Wait!**:one of these motors is moving to reset position.
- **Step done # they would be #**: the program informs the user how many steps have been done during reset and how many they should be.
- **Reset not done!**: this message appears if the motor is stopped by the user during the reset.
- **Motor process waits for the task**: the process is ready to start a new job.
- **polarimeter moving: wait!**: the program informs the user to wait because the polarimeter is moving to final position.
- **polarimeter traslating: wait!**: the program informs the user to wait because the polarimeter is moving to final position.
- **Motor process will be stopped in proper time.**: this message appears after the pressing of the STOP button.
- **Motor process is stopped**: this message appears after the real stop of the motor process.

- **Motor process is killed...**: this message appears after the real kill of the motor process.
- **Unrecognized message. Something is wrong with the link**: this message is sent when there are some problems on the link. Some informations are received on the link but they don't agree with the protocol messages.

4.3 Acquisition process messages

These messages can appear in *DOS* window where *xnir* was started:

- **Acquisition process failed to start! Problem with bs_socket**
- **Acquisition process failed to start! Problem with binding**
- **Acquisiton process failed to start!Problem with client process:error or timeout on basic socket**
- **Connection is broken? pending >= 501**
- **Error in starting process**: there are some problems in the starting of the process. This message is sent when the file with the list of all the processes running on the computer can't be opened. If you are working under DVX controls the *.dvp* file using the *Customize Menu* voice from the DVX menu. The name of the process in the first label named *Program name* **has to be written in lower case letter**[3].

The following messages can appear in the informative labels of the Acquisition process:

- **Acquisition process started**
- **Acquisition process waits for the task**: this message informs the user that the acquisiiton process is waiting for a new job.
- **Acquisition process will be stopped in proper time**: this message appears after the user has pressed the STOP button.
- **Acquisition process is stopped**:this message appears when the acquisition process is really stopped.

[3] i.e.: it should appear *xacquis* and not *Xacquis* or *XACQUIS*

- **Acquisition process is killed**: this message appears when the acquisition process is really killed.
- **...we read telescope position...**: the telescope position is read
- **...telescope position...**: the program is reading the telescope position.
- **Telescope is positioned at #**: the program informs the user about the source to which thelescope has been positioned.
- **...we start measurements...**: the acquisition measures are started
- **waiting for integrationg time....**: this message informs the user that he has to wait because the integration on signal is on.
- **Reading** : if double read is off:the program is reading the data array.
- **Reading (1)** : for double read, the program informs the user the the first array reading is on.
- **Reading (2)** : for double read, the program informs the user that the second array reading is on.
- **Computing** : the program informs the user it is collecting all the quantity to display.
- **Group # (#) Measurement # (#)is in progress. Wait**: the program informs the user about which coadd of the specified group it is doing.
- **Group # (#) Measurement # (#)is finished. Wait**: the program informs the user about which coadd of the specified group is finished.
- **Group # (#) is finished. Wait please...**: the program informs the user about which group is already finished.
- **Setting up**
- **Synchronizing.....**: syncronize with the read of array
- **...we send an image...**: the program informs the user that it is sending the image to display.

- **Handling Fits File**: the program is working on the fits file.
- **File # is written**: the program has finished to write inside the fits file and this file is stored in the data directory.
- **No focus value were returned**:
- **...focus setting, wait please....**: the program informs user to wait because the specified operation is on.
- **Insufficient memory!!!**: there is not enough memory to allocate the array to store data.
- **Timeout in focus setting.**
- **Ping Pong not Ready! (2)**
- **Ping Pong not Ready! (3)**
- **Ping Pong not Ready! (5)**
- **Ping Pong not Ready! (6)**
- **Ping Pong not Ready! (t0) Ping Pong status #**
- **Ping Pong not Ready! (t1) Ping Pong status #**
- **Ping Pong not Ready! (t2) Ping Pong status #**
- **Ping Pong not Ready! (t6) Ping Pong status #**
- **I cannot open temporary file! (1)**: it is not possible to open the temporary file where are stored the data. As default temporary directory it is used the *e* disk (generally the RAM disk). If this disk doesn' t exist on the PC where it is running the hardware, use the SETUP button (*Acquisition* menu) to set the correct temporary disk.
- **Error in closing temporary files!**: problems in closing the temporary file
- **I cannot flush temporary file! (1)**
- **I cannot rewind temporary file! (1)**
- **I cannot write temporary file! (1)**

- **Some errors!!**: there are problems during the mosaic acquisition (the process is stopped or some problems with the ping-pong board).

- **There was unrecognized message.Something was wrong with the link**: this message is sent when there are some problems on the link. Some informations are received on the link but they don't agree with the protocol messages.

These messages usually appear in their own widgets:

- **There are # bad pixels in 16x16 square centered at #,#**

- **There are # saturated pixels in 16x16 square centered at #,#**

- **Mosaic file not found!**: the mosaic file chosen by the user is not found. Look at the `resource/mosaic` directory. This message should not appear when the program xnir is run locally because the mosaic list is newly created when the button LIST of the mosaic panel is pressed.

4.4 List of resources

Below we report the complete application resource list for *xnir* program. The following list is hold inside the include file `xnir.h`. To know how to configure these application resources, look the section 2.1.

```
ButtonVertDistance     vertical distance between buttons
ObserverLabel          the label written inside the observer widget
ObserverName           the string with observer name

IntegrationLabel       the label written in the integration sub-panel
IntegrationTimeLabel   the label written in the first integration
                       sub-panel item
IntegrationTimeValue   the single integration time value
IntegrationTimeUnit    the string with time unit
IntegrationNumberLabel the label written inside the second integration
                       sub-panel item
IntegrationNumberValue the number of coadds
IntegrationGroupsLabel the label written inside the third integration
                       sub-panel item
IntegrationGroupsValue the number of groups
```

IntegrationTotalTimeLabel	the label written inside the third integration sub-panel item
FilterLabel	the label written in the Filter sub-panel
FilterName	the default filter name
FilterPath	the relative path of the arnica filters file
LFilterPath	the relative path for the longsp filters file
LfilterWavePath	the relative path for the longsp filters-wavelength range corrispondence file
DiaphragmLabel	the label written in the Diaphragm sub-panel
DiaphragmName	the default diaphragm name
DiaphragmPath	the relative path of diaphragm file
GratingLabel	the label written in the Grating sub-panel
PositionLabel	NOT USED
StepLabel	NOT USED
GratingOrderLabel	NOT USED
GratingLambdaLabel	NOT USED
MaxWave	max detector wavelength
MinWave	min detector wavelength
MaxOrder	max order used in user test mode
MinPolynomialOrder	minimun ordervalue used in the polynomial calibration
MaxOrderWave	maximun order-wavelength product value
Mirror	nirror position
FrameLabel	label written inside the Frame sub-panel
FrameName	the source name
MosaicLabel	label written inside the Mosaic sub-panel
MosaicName	the mosaic file name
PolarLabel	the label written inside the Polar sub-panel
PolarInsert	the value to insert polarimeter
PolarLinea	the value to put polarimeter off axis
TimerDelay	delay time for the button repetion

DisplayFileLabel	label written inside the first item of the display window
DisplayPath	relative path of the the file with the list of fits files
GratCoeffPath	the relative path of the file with the coefficients of the polynomial used for calibration
DisplayHostLabel	label written inside the second item of the display window
DisplayScreenLabel	label written inside the third item of the display window
TelescopeName	string with the telescope name
TelescopeAddress	string with the internet address of the server controlling the telescope
TelescopePort	the port number
HostsList	the relative path of the file with the list of the network accesible hosts
RootDirectory	string with the root directory name
HardwareHost	string with the name of the host where hardware processes are executed
HardwareScreenHost	string with the name of the scerrn host where the user interface is displayed on
HardwareRootDirectory	string with the root directory name for the host where hardware processes are executed
TempDisk	string with the ram-disk name
InternalAccount	the name of the account
PasswordSoftir	the account password
SacPath	the relative path of the file with the command to send to the motor control
Sac	flag to skip the SAC programming (True to skip, False not to skip)
Longsp	flag to start xnir for longsp instrument

GTestMode	flag to start longsp control for user (False)or test mode (True)
Polar	flag to start xnir with polarimeter control
Test	flag to execute xnir in test mode (without instruments and telescope)
ClockTime	clock time value
BitmapTelescope	name of the file with the telescope bitmap
BitmapInstrument	name of the file with the instrument bitmap
BitmapMotor	name of the file with the motor bitmap
BitmapPingPong	name of the file with the ping-pong bitmap
BitmapPolarimeter	name of the file with the polarimeter bitmap
BitmapStatusOn	name of the file with the status-on bitmap
BitmapStatusOff	name of the file with the status-off bitmap
BitmapR_row	name of the file with the right arrow bitmap
BitmapL_row	name of the file with the left arrow bitmap
Colormap	string with the default colormap name
ImageX0	upper left x coordinate of the square with the image
ImageY0	upper left y coordinate of the square with the image
ImageXsize	horizontal square side
ImageYsize	vertical square side
ZoomCenterX	center x coord. of the zoom square
ZoomCenterY	center y coord. of the zoom square
ImageMin	minimum image value
ImageMax	maximum image value
ImageSend	flag to send or not to send image
SocketBufferSize	buffer size for socket connection

```
FocusStep
FocusLevel
FocusSum

BadPixelLevel
SaturationLevel
SaturationWarning
```

For semplicity we have written only the class of the resource. The resource name is the same as the class resource but with the first letter lower case. For example:
ButtonVertDistance is the class of the resource buttonVertDistance is the name of the resource.

Acknowledgments

We are grateful to R. Stanga and F. Lisi for useful discussions and to prof. F. Pacini who have invited one of us (VG) to Arcetri.

References

[1] Lisi, F., C.Baffa, L.K.Hunt, 1993, SPIES's International Symposium on optical Engineering and Photonics in Aerospace and remote Sensing (Orlando, SPIE)
[2] Gennari S., Vanzi L., Ucla Conference, 1993
[3] L.Hunt et al., 1994, Arcetri Technical Report 4/1994
[4] DESQview/X User Guide, 1993, Quarterdeck