A User's Guide for ALFA

Draft

Stefan Hippler and Andreas Glindemann
Max-Planck-Institut für Astronomie
Heidelberg, Germany
July 1997
MPIA Release 1.1

Max-Planck-Institut für Astronomie
Königstuhl 17
69117 Heidelberg
Germany

http://www.mpia-hd.mpg.de/MPIA/Projects/ALFA

Chapter 1: Introduction

1. Overview

ALFA - the Adaptive optics with Lasers For Astronomy system for the Calar Alto 3.5m Telescope - consists of four major components:

1. The Telescope and its controls system (TCS)
2. The adaptive optics system (AO) consisting of:
   a. The adaptive optics Breadboard (figure 1.8) which holds all opto-mechanical parts
   b. the Wavefront Sensor (WFS) often also called Shack-Hartmann (SH) sensor
   c. the Tip-tilt Tracker (TTT)
   d. a TV based wide field guiding and acquisition system (TVG)
   e. a calibration and reference source (Laser)
   f. a compact interferometer (μPhase) to flatten the deformable mirror (DM)
   g. The adaptive optics controls system (AOS)
3. The Laser Guide Star (LGS) subsystem and its controls system
4. The science cameras (OMEGA-Cass, MAGIC) and their controls systems

This document describes how to set up the entire AO. Separate User's manuals exist for the Telescope controls system, the Laser Guide Star controls system and all science cameras. Only peculiarities of these instruments due to ALFA will be mentioned in this manual.

2. Software and Computer Overview

Most parts of the AO control software have been written in TCL/TK and C code. Parts of the Tip-tilt Tracker software are written in OCCAM. Other parts of the ALFA control software are written in EPICS sequencer (state notation language) code, as EPICS database records (db) and EPICS display list (dl) code.
All graphical user interfaces (GUI's) run on the SOLARIS (>2.4) operating system. The ALFA GUI's can be invoked on any X11 based display.

hbar and dospan: ALFA workstation and X-terminal

The recommended GUI hardware is either the standard ALFA workstation **hbar** which has two monitors or the standard ALFA X-terminal **dospan** with two monitors.

ALFA has several types of computer/processor hardware with software distributed across many processors.

trantor: the adaptive optics VME system

The core software to processing the adaptive optics (closed) loop runs on twenty 50 MHz Texas Instruments TMS320C40 Digital Signal Processors (DSP's). In each case four DSP's are in one HYDRA II board made by Ariel Corp. In total five HYDRA II boards and one Motorola MVME177 (50 MHz) board build **trantor**, one of several VME-bus based VxWorks/EPICS systems that control ALFA.
In order to compensate for tip-tilt distortions while guiding on a natural guide star (**NGS** mode), **trantor** has the same VME boards installed as **keahua** (see below)**.**
Additional boards installed in **trantor** are a UFORIA/VME board and a MVME712 distribution panel.

keahua: the tip-tilt Tracker VME system

The tip-tilt Tracker VME system **keahua** consists of a Motorola MVME177 (50 MHz) board, a MVME712 distribution panel, a INMOS B016 Transputer board and a TRAM motherboard MB2L8S-VME made by MSC Vertriebs GmbH. The tip-tilt (closed) loop runs on a Motorola 68060 CPU. Running the tip-tilt loop on **keahua** is mandatory when the fast AO loop guides on the artificial Laser star.

tvg2: the TV wide field acquisition and guiding system

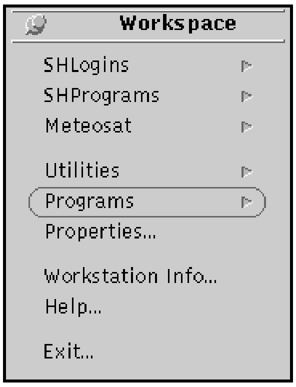
ALFA has one of the standard Calar Alto TV-Guider cameras installed on its breadboard. The field of view is about 4.6 x 3.6 arcmin. **tvg2** is a standard UNIX computer with two monitors attached. One monitor displays video images of the sky, the other monitor is used to display the TV-Guider control panel (GUI). A separate TV-Guider User's manual is available at Calar Alto.
The TV-Guider home page is http://www.caha.es/CAHA/Instruments/TVG/index.html at Calar Alto or http://www.mpia-hd.mpg.de/MPIA/Projects/TVG/index.html elsewhere.

uphase: the interferometer control laptop

A PC (laptop) that runs the FISBA µPhase Interferometer control software. The Twyman-Green Interferometer allows an *eye guided* flattening of the DM.

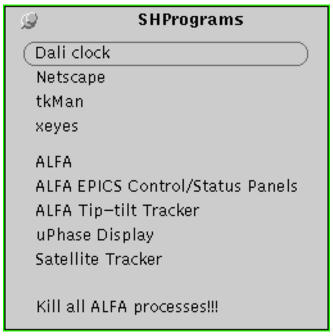
3. First steps

The beginning



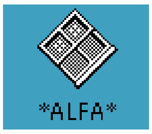
1. Everything is connected properly
2. Everything on the breadboard is switched off
3. **hbar** the primary ALFA workstation is running
4. Login to **hbar** as user **alfa** (for login/account information ask local staff)
   Note: If you are using the X-terminal **dospan**, select **hbar** from the login chooser menu. If the login chooser is not available give <ALT-GRAPH><HELP> on the keyboard. This starts the launcher program from which you can start a new X-terminal (xdm) session (via the login chooser menu)
5. Click with the right mouse button on the root window[1] (background on **data:6**[1]). This opens the **Workspace** menu. Select **console@keahua** and **console@trantor** from the **SHLogins** sub-menu
6. Turn on everything on the breadboard except the UFORIA-R (used with the DM) and the DM (see DM start-up procedure below)
7. Turn on **trantor** and **keahua** and check the messages that appear in the corresponding **console@...** windows
8. Wait for the lower red light on the **trantor** UFORIA board to go out
9. Turn on UFORIA-R which is located in the DM electronics rack
10. Turn on the DM
    a. Check that the master breaker in the back of the DM electronics is on
    b. Turn on the top-front circuit breaker. The boot light will come on for about 0.5 second. The other light may not always have the same pattern
    c. Check that the voltage knobs are fully counter clockwise
    d. Check that the current knobs are set to their default position (2.5 turns clockwise)

1.Controlling ALFA requires a lot of GUI windows. Two monitors/displays do not always offer enough space to view everything simultaneously. The **Virtual Desktop** offers more virtual screen space in order to run all GUI's simultaneously. Just double click with the left mouse button inside a virtual desktop field and you get a new screen. After having passed the login procedure you will see the virtual desktop window with 6 screen fields. So, you have 2 monitors with 6 virtual screens on each monitor makes 12 screens in total. For more information on virtual desktop see `man olvwm.` The monitor which displays the login sequence is called the **console monitor** (or console display), the other monitor is called the **data monitor** (or data display). In order to switch the input focus from the console monitor to the data monitor just move the mouse pointer over the right border of the console monitor (and back over the left border). Directing output to a particular monitor is controlled by the environment variable DISPLAY (e.g. `setenv DISPLAY dospan:0.0` for the console monitor or `setenv DISPLAY dospan:0.1` for the data monitor). The DISPLAY variable changes automatically when you move the mouse pointer from one monitor to the other.
Suggested window positions are specified as **monitor:virtualDesktopNumber**, where virtual desktops are numbered from left to right and top to bottom, e.g. the top left virtual desktop number is 1 and the bottom right virtual desktop number is 6. The reference **console:1** suggests that you create the given window (GUI) on the console monitor in virtual desktop 1.



e. Turn on both high voltage power supplies
f. Simultaneously turn the voltage knobs until both voltages read **40** Volts
g. Keep the voltages within +-5 volts
h. Raise the 80 Volt power supply voltage (bottom) until **70** volts are reached

11. Back to **hbar**, click with the right mouse button on the root window (background on **data:2**). This opens the **Workspace** menu. Select **ALFA EPICS Control/ Status Panels** from the **SHPrograms** sub-menu and wait until the main menu (alfaMenu.dl) appears. The display manager **dm** will open a second window that you don't need here. Close (do not quit or exit) it.
12. Click with the right mouse button on the root window (usually on **console:1**) and select **ALFA** from the **Workspace -> SHPrograms** menu. An xterm icon with the name ***ALFA*** appears on the root window and the ALFA main GUI comes up.
13. Open the ***ALFA*** icon (double click on it). Everything is fine when you see the messages below:
    Initializing Tcl...
    Initializing Tk...
    Initializing Tix...
    Initializing BLT...
    Initializing the AOA libraries...
    Initializing the HBAR common libraries...
    Initializing the AO libraries...
    Initializing the Stage libraries...
    Connecting to AO server...Done.
    Initializing AO subsystem...Done.
    Building the GUI...Done.
    Initializing Stage subsystem...Done.
    Connecting to Stage server...Done.
    %



14. Close (do not quit or exit) the ***ALFA*** window.
15. Click with the right mouse button on the root window (background on **data:4**). This opens the **Workspace** menu. Select **ALFA Tip-tilt Tracker** from the **SHPrograms** sub-menu and wait until the ALFA Tip-Tilt Tracker GUI appears
16. Click on the **Refresh CCD** button and check the System and Command status fields. The **Refresh CCD** command takes about 15 seconds for a successful operation
17. Click on the **data:2** window and click with the right mouse button into the **ALFA EPICS Panels** field. A pop-up menu comes up. Select **MPIA Stages (f10 ...)**, **MPIA Stages (dichroics ...)** and **AO Status Display.** Position all windows such that you can view and access them simultaneously. Select now the **Tracker Real Time Display** entry and move the up coming window into **data:4.** Do that by dragging the window into the **Virtual Desktop** field which contains the **ALFA Tip-tilt Tracker** GUI (**data:4**)
18. That's it for the moment

Figure 1.1: The ALFA Adaptive Optics Main GUI

First initializations and operational checkouts

Stages Part 1

Click on the **Stages** button within the Adaptive Optics Subsystem Main GUI (figure 1.1). A new window comes up (figure 1.2) that controls the following 12 stages (figure 1.8):

1. **fsmTilt**, the field select mirror Tilt-axis
2. **fsmTip**, the field select mirror Tip-axis (as well as Shutter and Laser Control)
3. **fsrfz,** field stop reference fiber Z-axis (focus)
4. **fsrfs**, field stop reference fiber selector with 3 distinct positions:

   a. Reference position
   b. Small hole position
   c. Big hole position
5. **hlas**, the hartmann lens array selector or Monolithic Lenslet Module (MLM) array selector. Five different MLM's are available:
   a. 3x3 sub-apertures (MLM 1)
   b. 5x5 sub-apertures (MLM 2)
   c. 9x9 sub-apertures (MLM 3)
   d. 10x10 sub-apertures (MLM 4)
   e. 12x12 sub-apertures (MLM 5)
6. **rlm**, the relay lens module
7. **tcx**, the Tip-tilt Tracker camera X-axis
8. **tcy**, the Tip-tilt Tracker camera Y-axis
9. **tcz**, the Tip-tilt Tracker camera Z-axis
10. **tfw**, the Tip-tilt Tracker filter wheel
11. **wfs**, the Wavefront sensor camera position
12. **wfsfw**, the Wavefront sensor camera filter wheel

Initialize stages!

The first thing you should do is initializing all stages. Click on the radio button of the stage that you want to initialize and select the **Initialization Sequence** entry in the **Operations** menu. You can also **Initialize ALL stages** at once. Initializing all stages takes up to 12 minutes

Stages Part 2

Go to the MPIA stages control panel (mpiaStages1.dl, figure 1.3). This panel gives allows you to control 4 stages:

1. **TV Guider**, the TV guider camera Y-axis (the one that is perpendicular to the TV-Guider focus axis) with two characteristic positions:
   a. **On axis**
   b. Off axis
2. **Tip-tilt mirror**, the tip-tilt mirror is the first mirror in ALFA that sees light from the Telescope (tertiary mirror). This mirror has three characteristic positions:
   a. **ALFA in**: Tip-tilt mirror is in the on-axis position
   b. **Out**: Tip-tilt mirror is completely out off the f/10 beam
   c. **TV in**: Tip-tilt mirror moves the fixed connected TV mirror to the on-axis position
3. **dichroic (Tv)**, one of two dichroics with two positions: **in** our **out**. Only one of the two dichroics can be in its **in** position
4. **dichroic (Tip)**, see dichroic (Tv). The additional names Tv and Tip only indicate on which side of the Breadboard the mirrors are: either close to the TV Camera or close to the tip-tilt mirror

Figure 1.2: The ALFA Adaptive Optics Subsystem Stages Panel

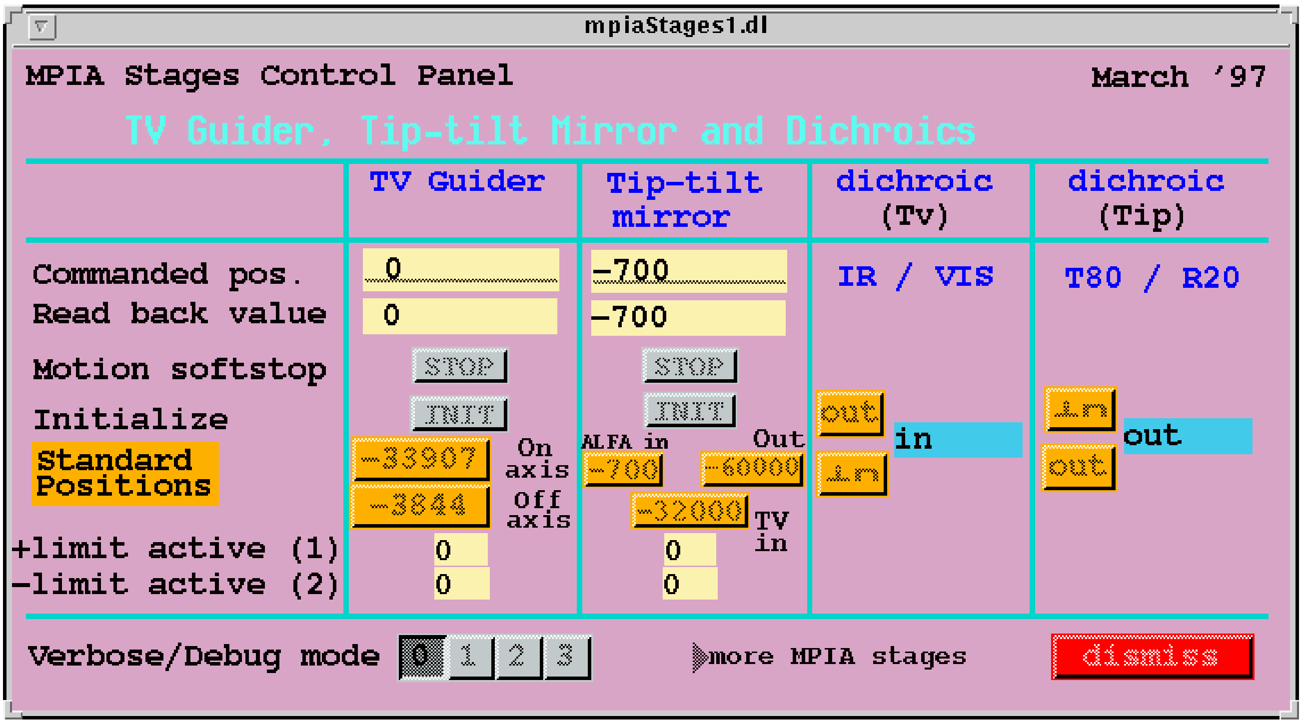


Figure 1.3: MPIA Stages (TV Guider etc.) Control Panel

Initialize stages!

The **INIT** button blinks to indicate that this stage has not yet been initialized. Press the **INIT** button to start the initialization. The **STOP** button blinks while the stage is initializing. You can press all **INIT** buttons without waiting for an initialization process to complete.

Stages Part 3

Go to the MPIA stages control panel (mpiaStages.dl, figure 1.4). This panel allows you to control 5 stages:

1. f10 parabola **focus**, the f10 parabola mirror focus position
2. f10 parabola **tip**, the f10 parabola mirror tip-axis
3. f10 parabola **tilt**, the f10 parabola mirror tilt-axis
4. f10 reference fiber **x**, the f10 reference fiber x-axis with two characteristic positions:
   a. **in**, move reference fiber into the f10 beam (optical path)
   b. **out**, move reference fiber out of the f10 beam

   The f10 reference fiber is switch on and off via the **Utilities** panel
5. f10 reference fiber **focus** with two characteristic positions:
   a. **NGS**, focus position to simulate a Natural Guide Star
   b. **LGS**, focus position to simulate a Laser Guide Star

   The f10 reference fiber is switch on and off via the **Utilities** panel

Initialize stages!

The **INIT** button blinks to indicate that this stage has not yet been initialized. Press the **INIT** button to start the initialization. The **STOP** button blinks while the stage is initializing. You can press all **INIT** buttons without waiting for an initialization process to complete.

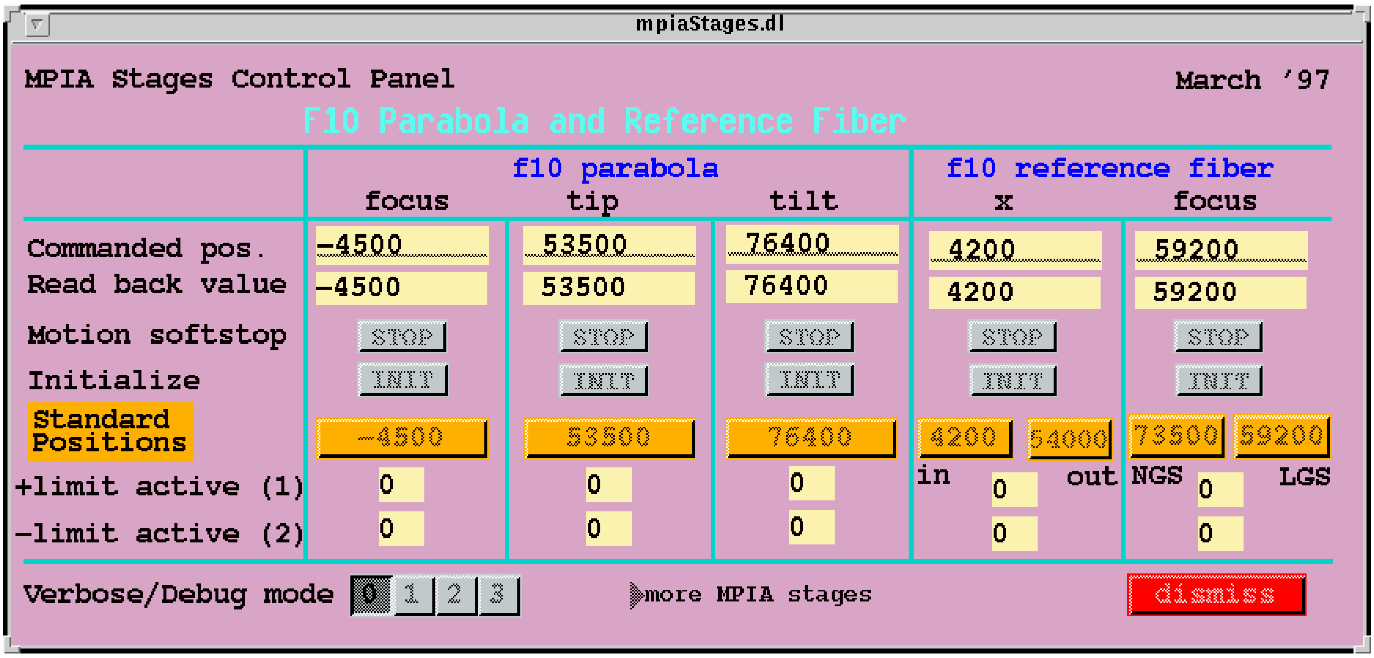


Figure 1.4: MPIA Stages (F10 Parabola etc.) Control Panel

The Wavefront Sensor (Shack-Hartmann Sensor)

Clicking on the **Camera** button inside the AOS main GUI raises the **LLCCD Camera** window. This window (see figure 1.5) provides both control and diagnostic settings for the Lincoln-Lab CCD camera inside the Wavefront sensor.

Check the LLCCD Video Display:

Check the **CRT ON** radio button (even if it is already marked on!). The LLCCD video control monitor (RS-170/NTSC display) starts displaying live images. Defaults settings are:

1. CRT: **ON**
2. TE (ThermoElectric) Cooler: **ON**
3. Camera Gain: **1** (possible selections: 1 or 2)
4. CRT Gain: **2** (possible selections: 0,1,2,3,4)
5. Frame Rate: **301** (possible selections: 100, 201, 301, 408, 500, 600, 702, 804, 902, 1000, 1018, 1058, 1078, 1100, 1122, 1144, 1168, 1194, 1206)
6. TE Current: usually around 0.2 A
7. TE Voltage: usually around 1 V
8. TE Temperature: usually around -5 degrees centigrade

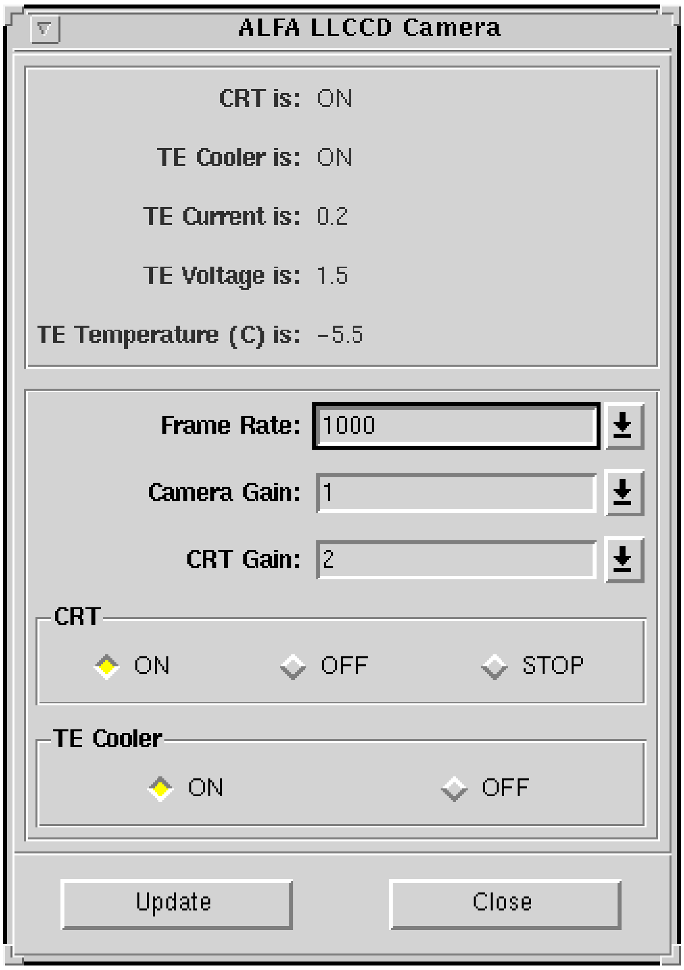


Figure 1.5: LLCCD Camera Panel

Move all stages to their default/ standard positions

After all stages are initialized the next step to do is moving them into their default positions. Some of the MPIA stages (mpiaStages.dl, mpiaStages1.dl) have more than one default position. These default buttons usually have numbers as label. In order to calilbrate the AO system for use with a natural guide star select the **in** and **NGS** positions of the **f10 reference fiber**. One of the dichroics has to be in its **in** position. Before a dichroic can be moved **in**, the other dichroic has to be in its **out** position.

Open then **Utilities** panel (from the ALFA main GUI, figure 1.6) and select the filter of the **WFS Filter Wheel** you need in order to run the AO loop. The **f/10 Reference** fiber can either be switched on via the **f/10 Reference On** button or externally on the Breadboard. Select the **Big Hole Field Stop**. The **Tracker Filter Wheel** position can be selected later.

If all stages have reached their default position the image of the optical fibre should appear on the Shack-Hartmann sensor (SH sensor, WFS) and in the center of the tip-tilt sensor (TT sensor)

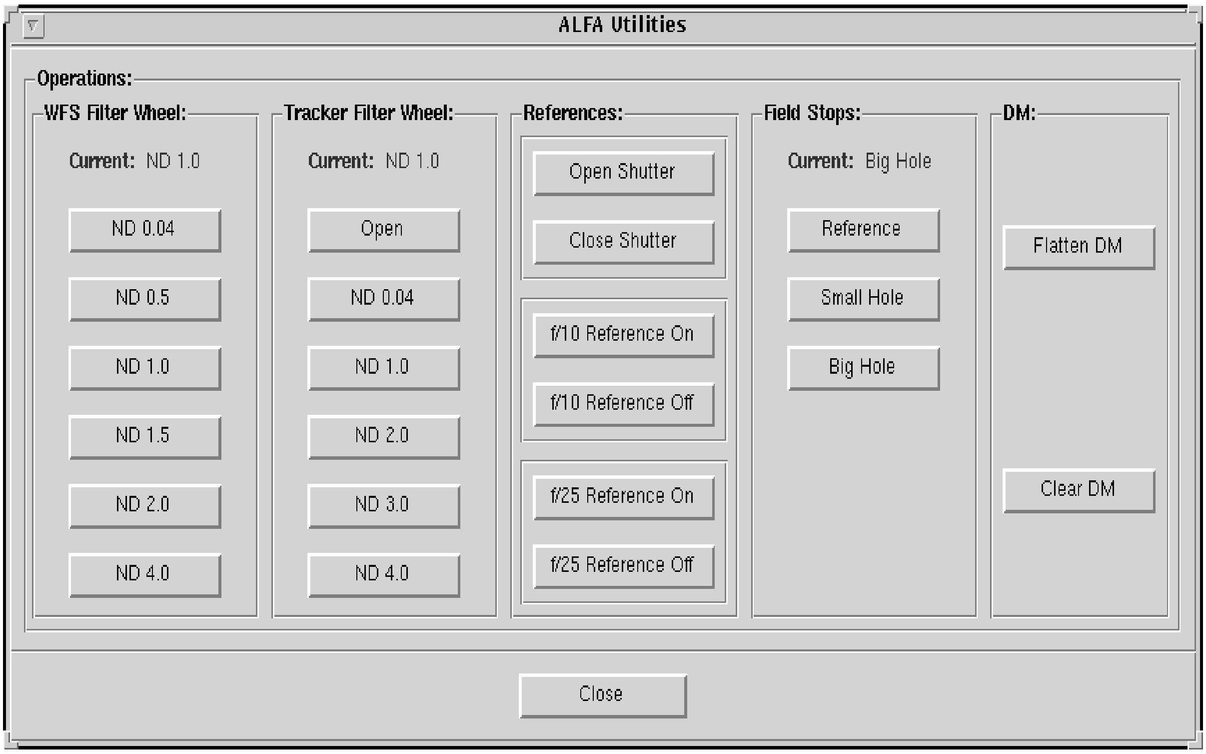


Figure 1.6: The ALFA Utilities Panel

Chapter 2 describes the calibration procedure of the SH sensor, and Chapter 3 deals with the TT sensor.

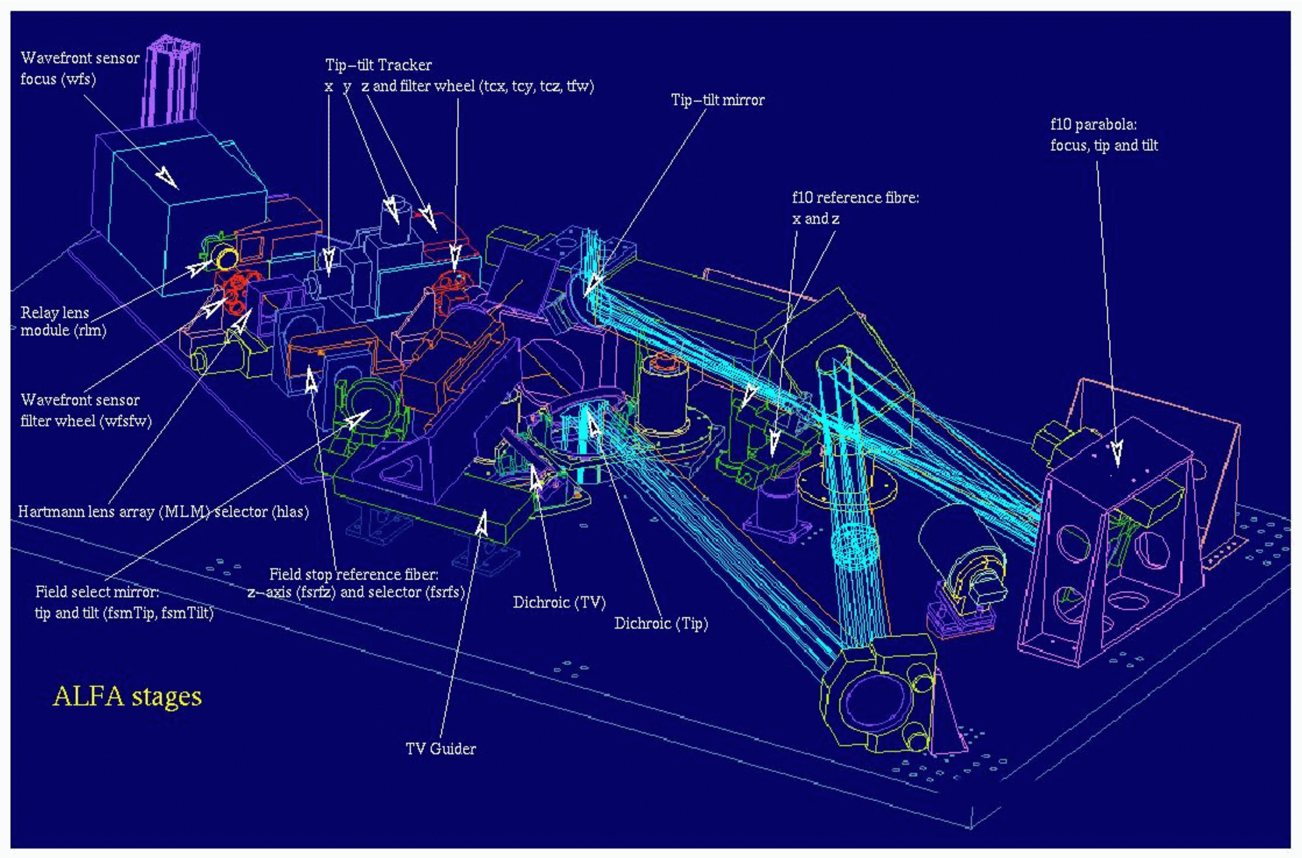


Figure 1.7: Location of ALFA's motorized stages on the Breadboard

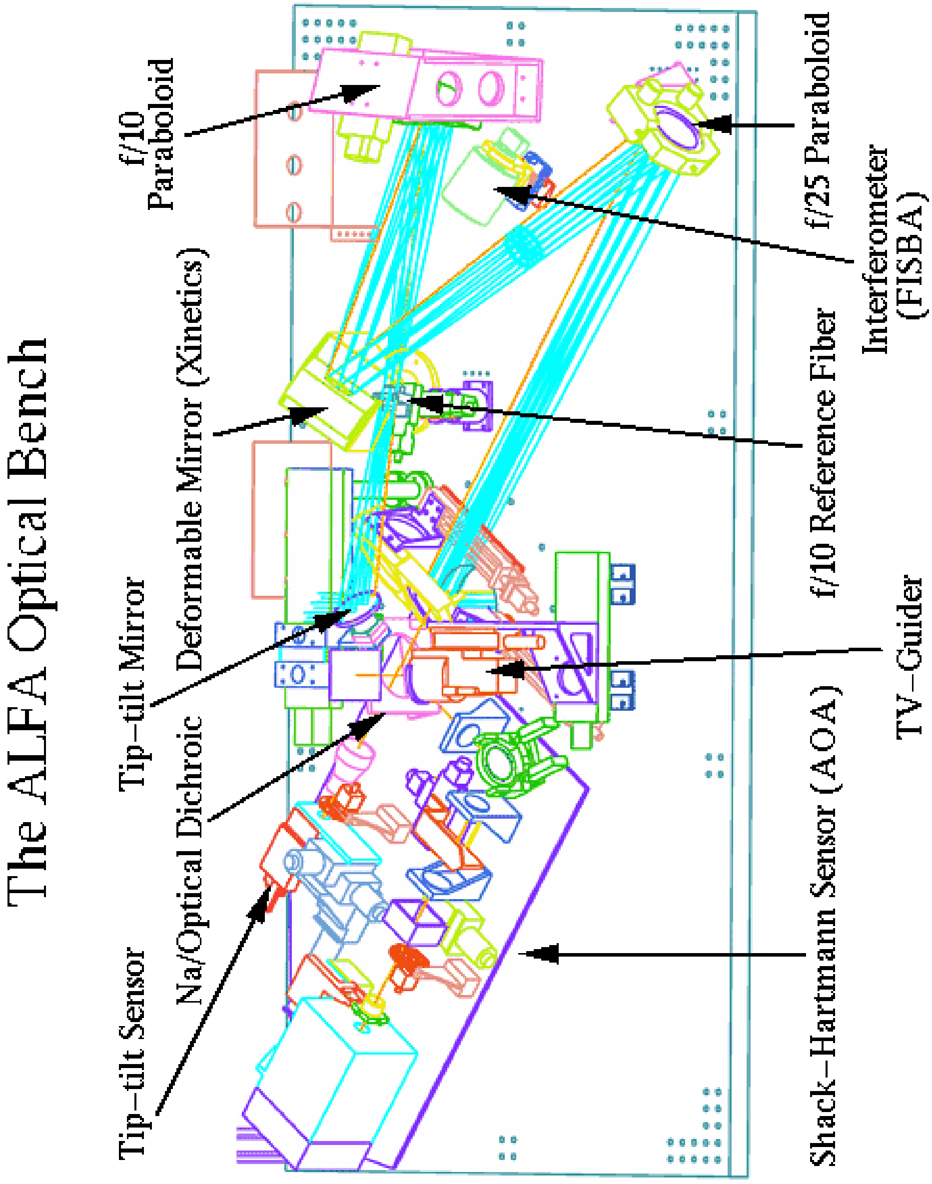
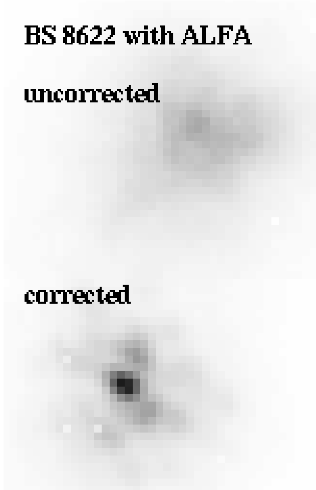


Figure 1.8: The ALFA Optical Bench

Chapter 2: Adaptive Optics Correction



1. Flatten DM Procedure

Starting the PC program: ki

A micro interferometer based on the Twyman-Green system is used to measure the surface roughness. Start the measuring program on the **uphase** PC:

1. Switch on **uphase** (located below the ALFA breadboard)
2. **C:<Enter>**
3. **CD uphase<Enter>**
4. **ki<Enter>**
5. Press **<F10>** to enter the live fringe adjustment display
6. ... adjust the fringe pattern ...
7. Press **<Esc>** to exit the adjustment display
8. Press **<F7>** to enter the calibration menu
9. ... select **delay 30** if you want automatic measurements with a 30 s delay between measurements
10. Press **<Enter>**
11. ... this loads the calibration profile from disk ...
12. ... the message "Starting with automatic measurement" appears
13. ... check the result on the laptop screen
14. ... if everything looks fine, close the laptop

All data coming from the interferometer are stored on **hbar** in the directory **/data/alfa/uphase**. All filenames begin with the prefix **pdat1**. Numbers are added to this prefix in increasing order. It's sometimes a good idea to remove all interferometer data files **before** you start the PC program **ki** because **ki** can handle no more than 999 data files (e.g. pdat1999.dat and pdat1999.pcx are the last files **ki** can create)**.** Remove the data files with the command **cd /data/alfa/uphase; rm pdat1*.pcx pdat1*.dat**

Display interferograms on the X-display

Open the **Workspace** popup menu, select **SHPrograms** and there select **uPhase Display**. A new window appears which displays the last interferometer image (pdat1*.pcx) created by **ki**. The **uPhase Display** automatically updates the screen when new interferograms are available (see figure 2.1). Type **q** inside the uPhase Display to quit the program. For more information about see **man display**.

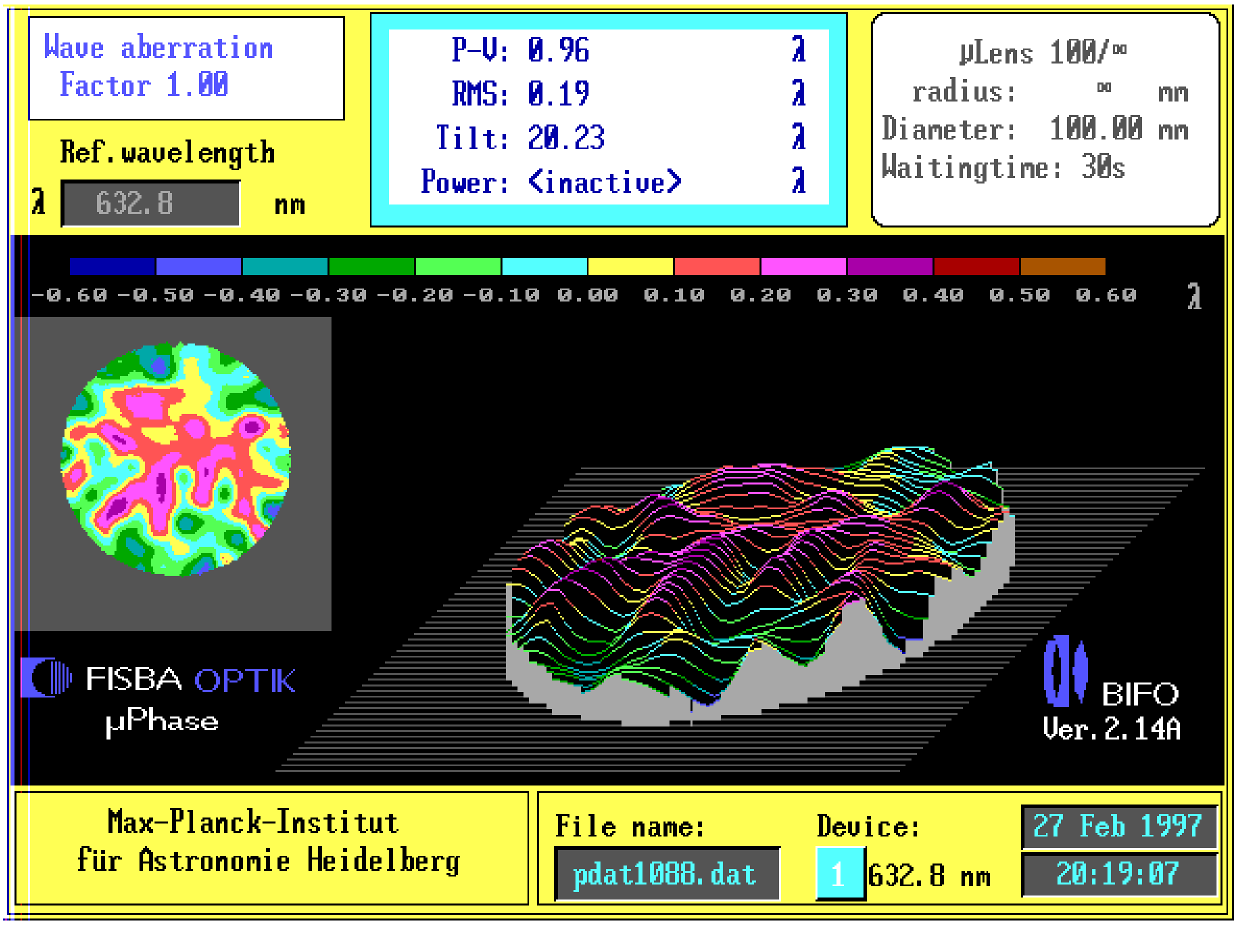
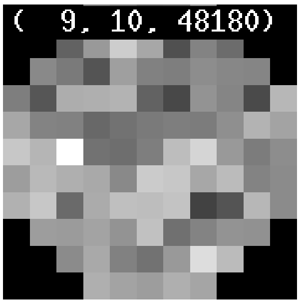


Figure 2.1: uPhase Display

Flatten the Deformable Mirror



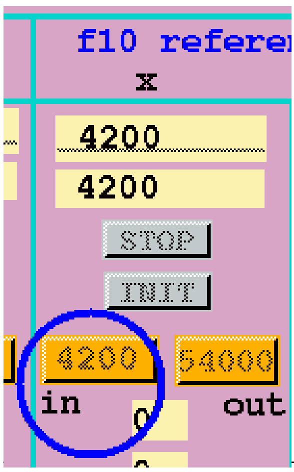
Open the ***ALFA*** window and type in the commands:
cd $HOME/FLATDM
makeFlat
The messages:
makeFlat-Info: using /usr/aos/alfa/config/DMBias as input profile
makeFlat-Info: creating DMBias.new
appear inside the ***ALFA*** window and a new window pops up showing the actual "flatten" pattern of the DM. Each square (besides the squares in the four edges) is mapped to one of the 97 DM actuators. Before you can start poking individual actuators you have to select an actuator by moving the mouse cursor onto the square (=actuator) you want to poke and then press either **<space>** or **m.** As long as the mouse cursor is inside the makeFlat GUI it is possible to poke the selected actuator with the keystrokes described below (poke values range from 0 to 65535 and are displayed as third number in the makeFlat GUI):

1. + (add 1)
2. - (subtract 1)
3. <Arrow up> (add 10)
4. <Arrow down> (subtract 10)
5. <Page Up> (add 100)
6. <Page Down> (subtract 100)
7. <space> or m (select actuator at cursor position)
8. r (refresh the **makeFlat** GUI)

When ever you press one of the keys 1-6 the entire "flatten" profile is **immediately** send to the DM and the new profile is stored in the file DMBias.new (default). If you want to keep an intermediate pattern you have to save the DMBias.new file (e.g. **cp DMBias.new DMBias.new.00**). The "flatten" pattern given by **makeFlat** and the contour plot of the uPhase display correspond approximately one to one.

2. Closed Loop Operation

Calibration of the AO system



The fiber image on the SH sensor is used to calibrate the AO system. That means that the wavefront provided by the fiber is considered to be the "perfect" plane wave and the position of the subimages in the SH sensor are the reference positions when running the system in closed loop. Therefore it is necessary to align the off-axis paraboloids and the focus position of the fiber such that the fiber image on MAGIC is optimised. Usually, fiddling with the f/10 tilt and the fiber focus is enough to create a spot with a FWHM <2 pixels on MAGIC. Now, the shape of the wavefront is sufficiently good and the pattern of the subimages on the SH sensor provides the reference positions for the closed loop operation.

This procedure eliminates the influence of all aberrations introduced between the IR/Vis dichroic and the SH sensor.

Before you start the calibration procedure go to the MPIA Stages Control Panel and move the f10 reference fiber to the **in** position.

In order to start the calibration procedure click on the **Calibration** button in the main GUI. Then the Calibration GUI (figure 2.2) will appear. Click on **Take Offset** first to create the offset image suitable for the frame rate in use. Clicking on **Take Reference** opens the Reference GUI (figure 2.3) where clicking on **Reference Image** takes an image of the subimage pattern on the SH sensor. Now check the signal level by clicking on **Info** and **Open...** in the displayed image (figure 2.4). If the maximum is below about 3700 the light level of the fiber is all right. (The maximum signal is 4095 counts but in the reference image the offset values of a few hundred are subtracted.) If not, either change the frame rate (in the LLCCD **Camera** Panel) or the ND filters (in the **Utilities** GUI). Changing the frame rate requires to take a new offset as the levels are slightly different.

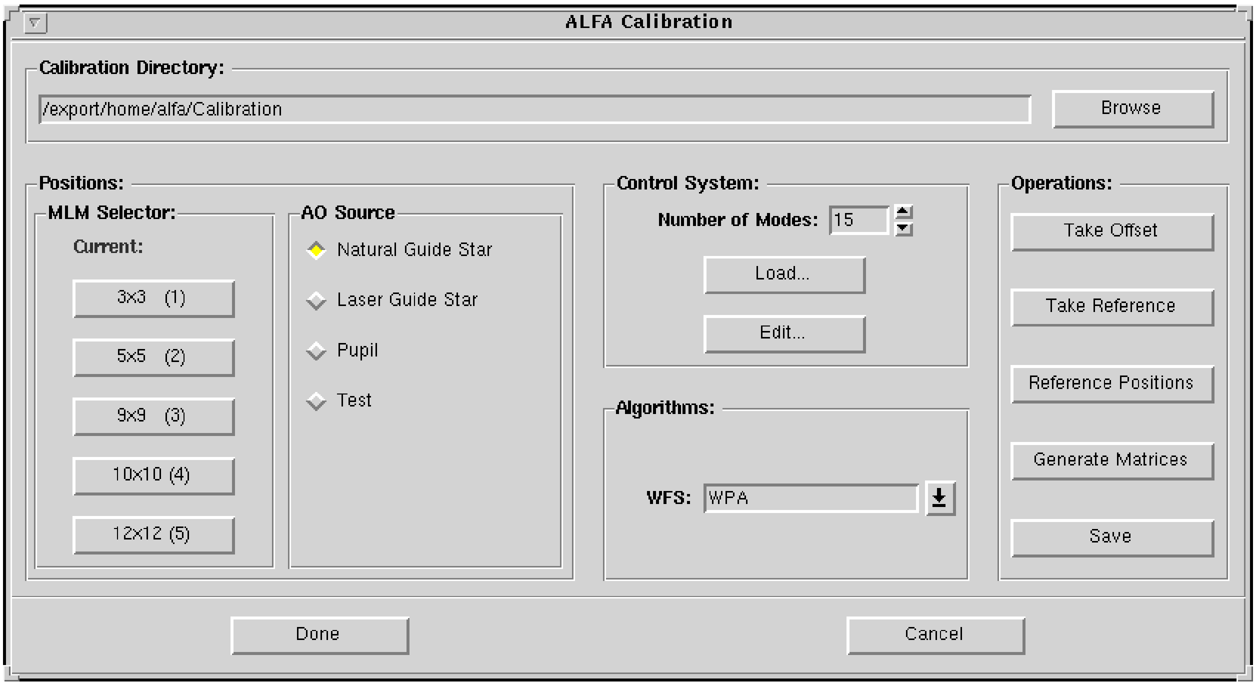


Figure 2.2: The ALFA Calibration GUI

If the signal level is satisfactory select the **Hexagonal Grid** type and click on **Compute Rectangles** which forces the system to search for those spots that are brighter than 0.2 times the peak intensity. (This number is adjustable with **Brightness Threshold** in the Reference GUI.) ALFA calculates the most suitable positions of rectangles around the spots. These define the area on the CCD within which the centroid is calculated. Playing with the parameters **Min Pixels between Centers** and **Max Pixels between Centers**, pretty quickly gives you a feeling of how to adjust them in order to produce the desired size of rectangles. When the calculated rectangles are displayed as green squares clicking inside them with the left mouse button unselects them. This is important for the central obscuration of the telescope pupil. If in doubt which central boxes to unselect look at a star first. Also, if the rectangles are too close to or partially outside of the CCD array they have to be unselected. After unselecting the rectangles click on **Redraw** to create a new pattern of subarrays.

Now, the main calibration of the system can start. Back in the **Calibration** GUI (the Reference GUI can remain open) click on **Reference Positions** in the **Calibration** GUI to calculate and display the centroid position in each box. Then, after having selected the

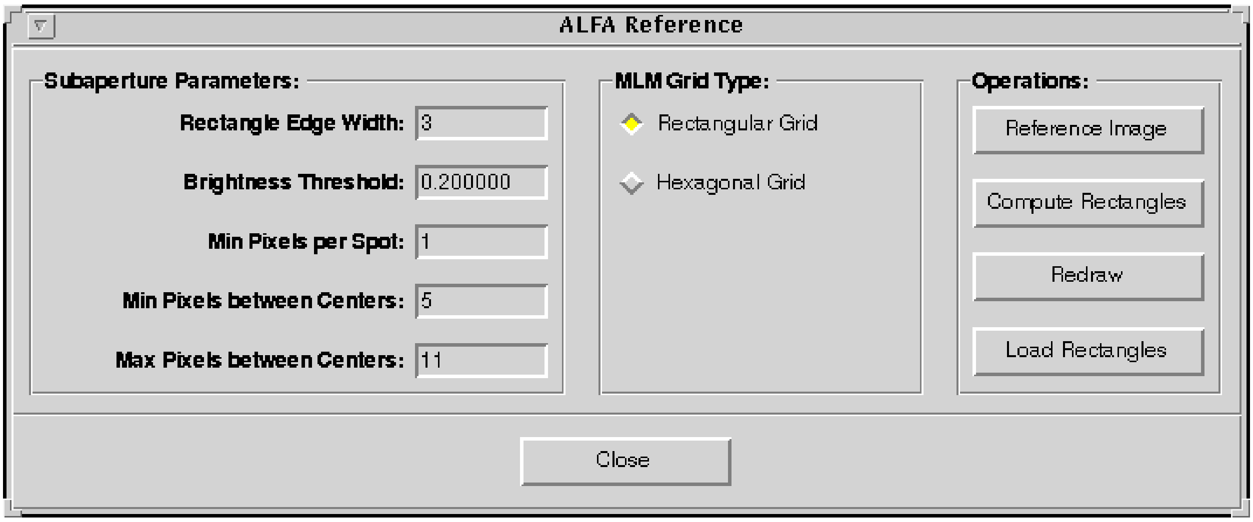


Figure 2.3: The ALFA Reference GUI

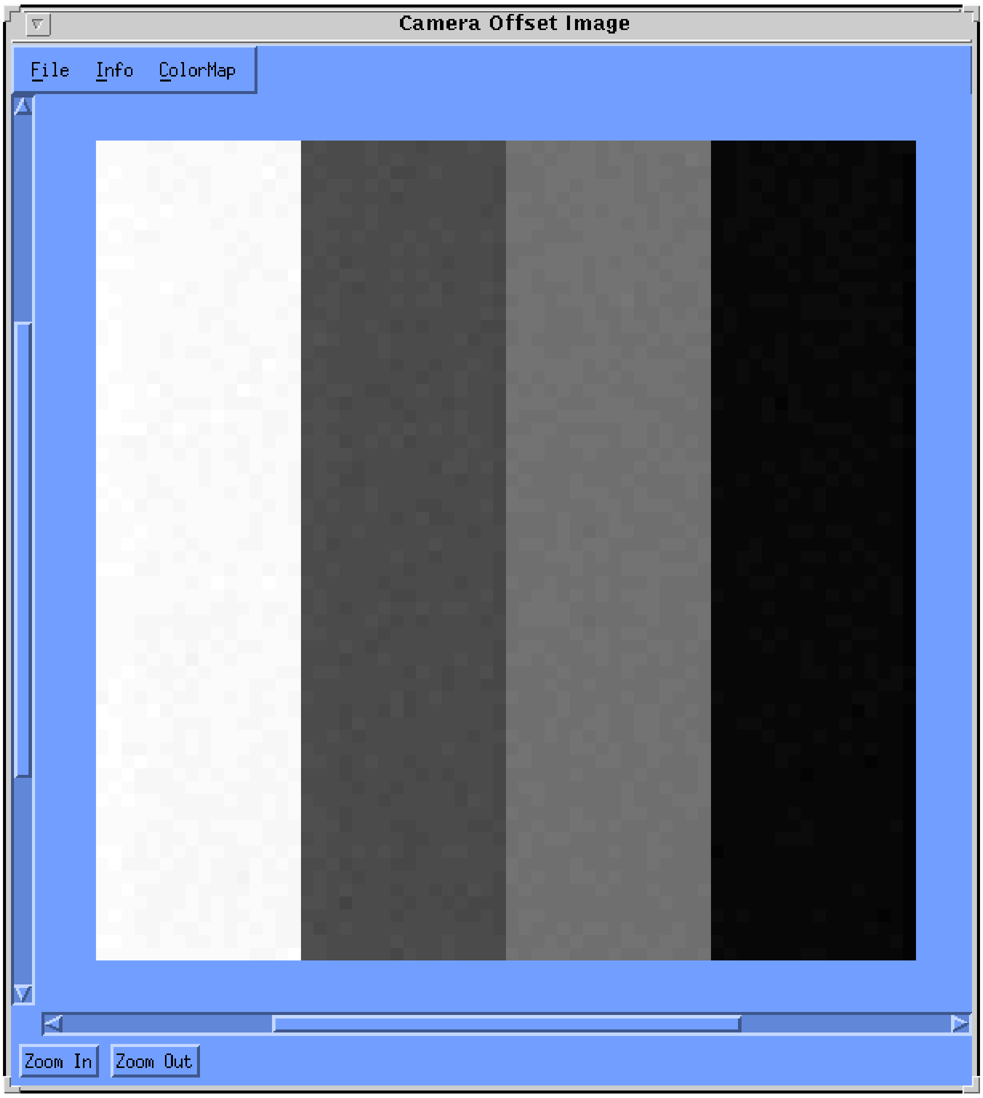


Figure 2.4:
Take Offset display

Number of Modes and the **WFS Algorithm**, hit **Generate Matrices** to send the Zernike modes one by one to the DM and register the change in centroid positions in the subarray for each Zernike mode. This procedure of "measuring" the change that goes with the Zernike modes rather than "calculating" the centroid positions of the SH sensor for each mode greatly facilitates the alignment of the system as accurate knowledge of the aperture shape and position with respect to the lenslet array is not required.

You can choose between two algorithms for the wavefront analysis: 1) **WPA** or weighted pixel averaging which computes the spot center as "center of gravity" of the intensity distribution and 2) **Spot Fitting** which calculates the spot center by finding the pixel with the highest intensity in the given subaperture and performing then a fit over a 3x3 area centered around this pixel.

After the calibration is done the results that are stored in a number of files have to be saved by clicking on **Save**. Pressing the **Done** button is equivalent to pressing the **Save** and then the **Cancel (**dismiss) button. In the upper part of the Calibration GUI the directory can be chosen where the files are supposed to be saved. The default name is Calibration but by creating other directories in an x-term window one can create a variety of data sets, e.g. for different numbers of subimages. A feature of the system requires that you **Edit...** the number of modes, and explicitly click on **OK** in the ALFA Modal Weights window (figure 2.5) although the proper number of modes was used for the calibration procedure. All actions above the Save button require to click on **Save** or **Done** in order to save the files in the directory Calibration. The modal weights are saved by clicking on **OK** in the ALFA Modal Weights GUI. Thus, changing the weighting of the modes requires an **OK** to confirm and save the new parameters.

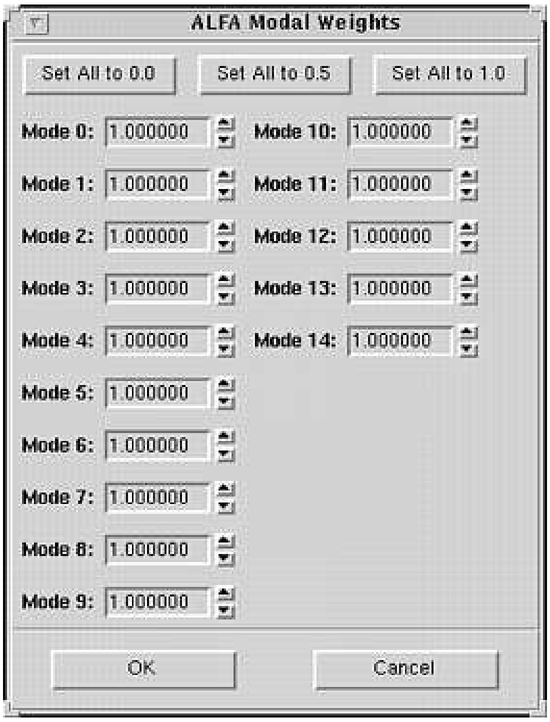


Figure 2.5: The ALFA Modal Weights (Edit ...) GUI

The calibration procedure is finished now. In order to run the system with the parameters determined in the calibration procedure go the main GUI, click on **Closed Loop** and, if you want to use the new calibration (rather then a set that was saved earier) hit **Create**

New in the Corrective Loop Setup GUI (figure 2.6) and click **OK** when being asked whether the Setup directory should be overwritten. If you didn't **Edit...** the modes after changing the number of modes with respect to an earlier calibration the system complains by moaning about the number of modes not fitting the other parameters. Go back to the **Calibration** GUI, click on **Edit...** and **OK**. Return to the main GUI and **Closed Loop** and **Create New** should now create a new set of files in the Setup directory that is being sent to the VME crate right away. If you want to make sure that this Setup directory is not overwritten again give it a different name in an x-term window. Using it again, one simply has to chose the proper directory name for the Setup directory in the Corrective Loop Setup GUI (that pops up after clicking on **Closed Loop** in the main GUI) and click on **Use Existing**. The parameters are now in the VME system and the loop can be started by clicking on the first radio button labelled **Start** in the middle row of buttons in the main GUI. The system than runs in open loop. The SH sensor image, the gradients, the DM signals and the Zernike modes can be displayed at a rate of about 1Hz in the **Data** Handler GUI (see figure 2.7).

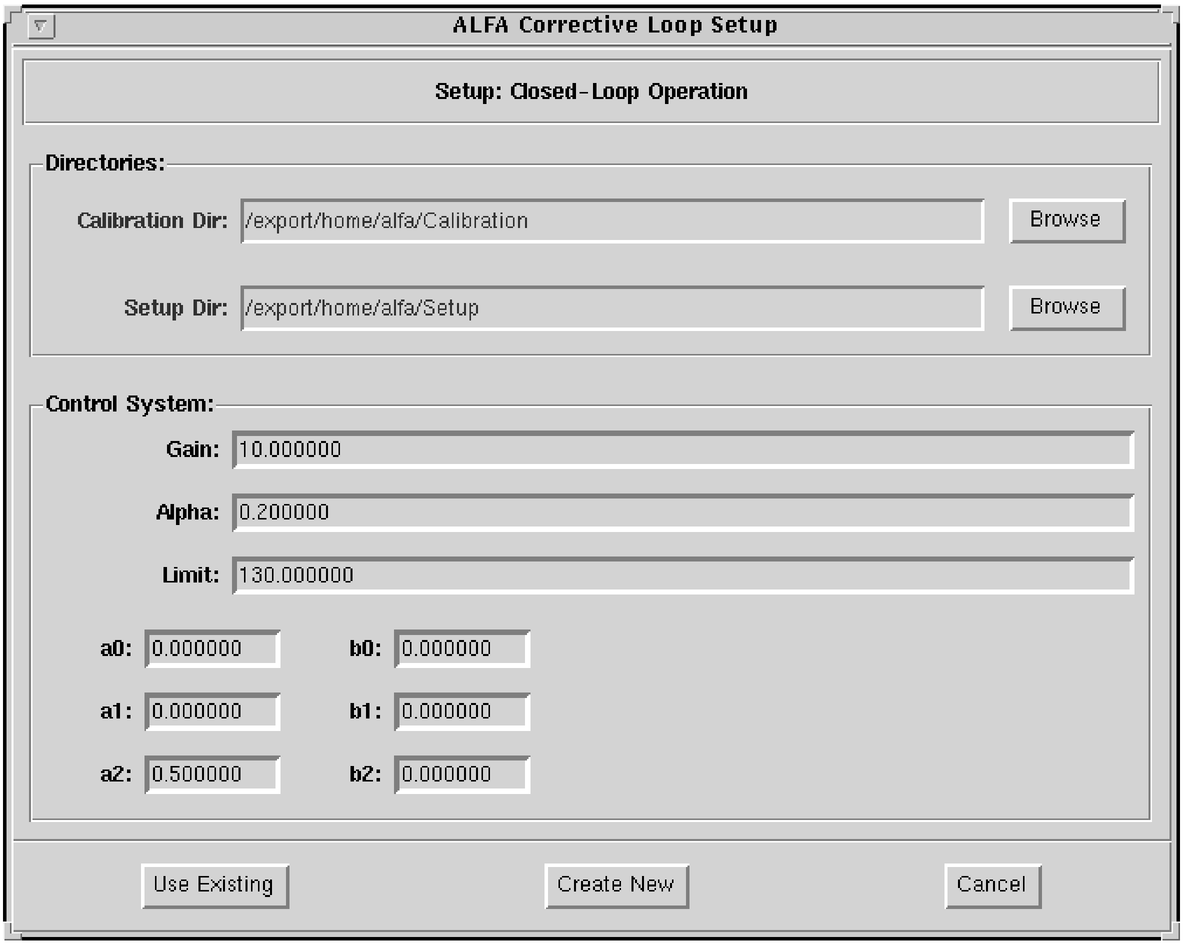


Figure 2.6: ALFA Corrective Loop Setup

Running in open loop on the fiber image should show very small gradients (i.e. very short arrows) and closing the loop by clicking on **Close** in the right column of the main GUI should have no visual effect except that the signals sent to the DM should have non-zero values. These values might look pretty large but bear in mind that the grey scale is adjusted according to the maximum and minimum values being sent.

If this is all satisfactory the loop can be opened by clicking on **Open** or **Freeze**. The only difference between the two is that **Open**, in addition to opening the loop, also sets the DM actuators to their default (i.e. flat) positions while **Freeze** keeps all actuators at their last positions. Only in **Open** loop mode it is possible to modify parameters of the **Control System** like Gain, Alpha, Limit, a0, a1, a2, b0, b1 and b2.

Now the fiber can be moved out and the AO correction on a star can begin.

3. Wavefront Sense

Selection of the **Wavefront Sense** button will bring up the same panel as for **Closed Loop** operation but without the **Control System** part. **Wavefront Sense** requires a calibrated system but you don't have to **Generate Matrices** for a closed loop operation. This mode is useful for measuring turbulences in the atmosphere. Up to **6000 gradient data sets** can be collected in one run (-> ALFA Data Handler -> **Start Collection**).

4. Image Only

The third choice in the Setup section of the ALFA Main GUI will raise raise the **Setup: LLCCD Image Collection** subpanel. Click **OK** to select this mode or **Cancel** the operation. In **Image Only** no calibration or setup directories are required. The ALFA Data Handler (figure 2.7) can display and save LLCCD images at a rate of about 2 Hz.

5. Diagnostics

This mode is for maintenance only. It runs diagnostic/test routines on the DSP's.

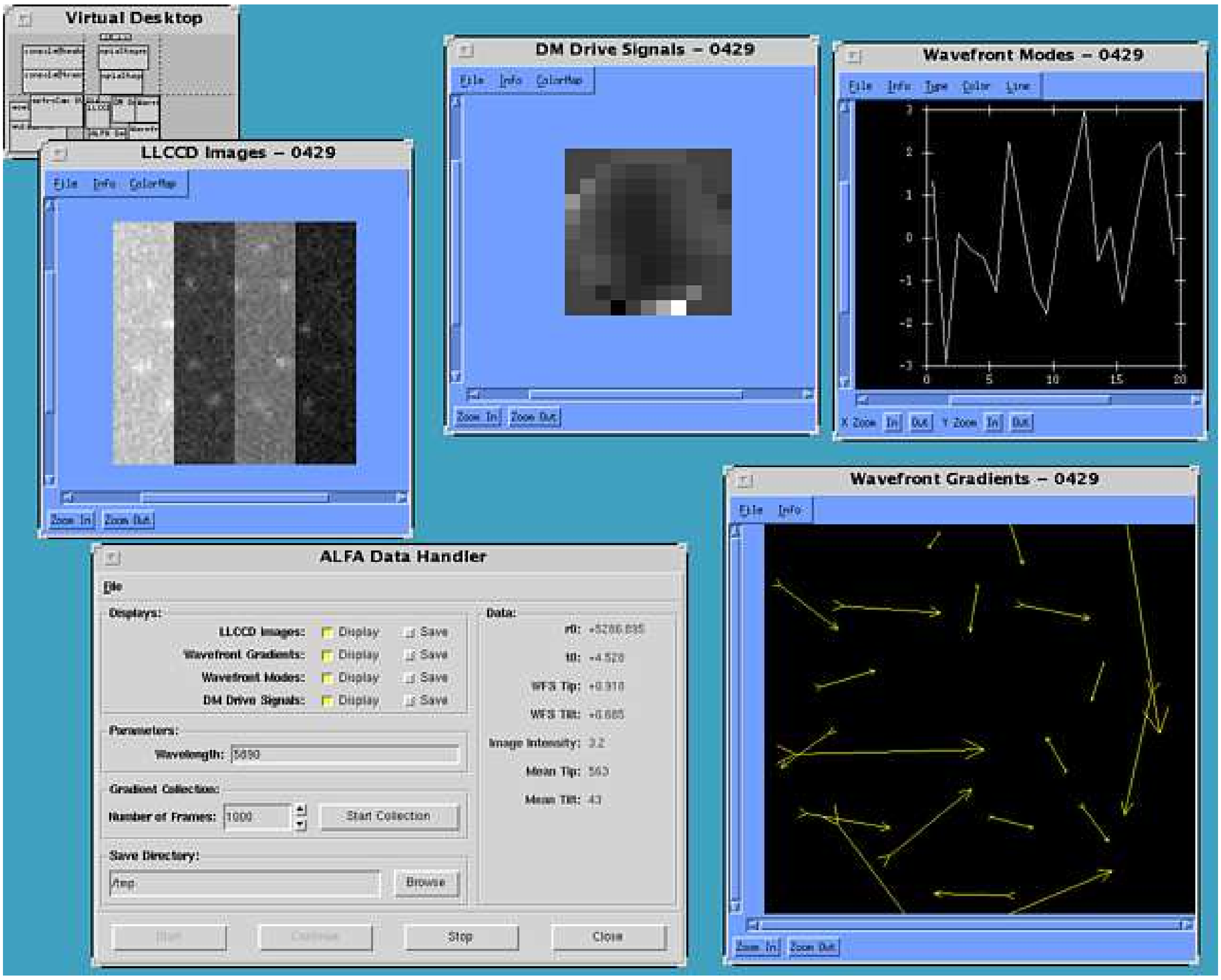


Figure 2.7: ALFA Data Handler Windows

Chapter 3: Tip-tilt Correction

8. Setup

9. Imaging modes

10. Other business

11. Data files

12. Adjusting and verifying scale factors

Chapter 4: What to do when ...

1. Questions and answers about using ALFA software

Chapter 5: Log Files

1. console@trantor boot log

VxWorks System Boot

Copyright 1984-1994 Wind River Systems, Inc.

CPU: Motorola MVME177
Version: 5.2
BSP version: 1.0
Creation date: Thu Jul 20 17:05:04 PDT 1995

Press any key to stop auto-boot...
 1 0
auto-booting...

boot device : ei
processor number : 0
host name : hbar
file name : /usr/aos/hbar/vw/config/mv177/vxWorks
inet on ethernet (e) : 150.214.222.22:ffffff00

```
host inet (h)        : 150.214.222.20
gateway inet (g)     : 150.214.222.1
user (u)             : alfa
flags (f)            : 0x8
target name (tn)     : trantor
startup script (s)   : /usr/aos/alfa/etc/HBARStartup.cmd

Attaching network interface ei0... done.
Attaching network interface lo0... done.
Loading... 497440 + 30600 + 27590
Starting at 0x1000...

Attaching network interface ei0... done.
Attaching network interface lo0... done.
Loading symbol table from hbar:/usr/aos/hbar/vw/config/mv177/vxWorks.sym ...done


 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
      ]]]]]]]]]]]  ]]]]     ]]]]]]]]]]       ]]              ]]]]         (R)
 ]     ]]]]]]]]]  ]]]]]]     ]]]]]]]]       ]]               ]]]]
 ]]     ]]]]]]]  ]]]]]]]]     ]]]]]] ]       ]]                ]]]]
 ]]]     ]]]]] ]    ]]]  ]     ]]]] ]]]   ]]]]]]]]]  ]]]] ]] ]]]]  ]]   ]]]]]
 ]]]]     ]]]  ]]    ]  ]]]     ]] ]]]]] ]]]]]]   ]] ]]]]]]] ]]]] ]]   ]]]]
 ]]]]]     ]  ]]]]     ]]]]]      ]]]]]]]] ]]]]   ]] ]]]]    ]]]]]]]    ]]]]
 ]]]]]]      ]]]]]     ]]]]]]    ]  ]]]]]  ]]]]   ]] ]]]]    ]]]]]]]]    ]]]]
 ]]]]]]]    ]]]]]  ]    ]]]]]]  ]    ]]]   ]]]]   ]] ]]]]    ]]]] ]]]]    ]]]]
 ]]]]]]]]  ]]]]]  ]]]    ]]]]]]]      ]    ]]]]]]]  ]]]]    ]]]]  ]]]] ]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]      Development System
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]      VxWorks version 5.2
 ]]]]]]]]]]]]]]]]]]]]]]]]]]      KERNEL: WIND version 2.4
 ]]]]]]]]]]]]]]]]]]]]]]]]]      Copyright Wind River Systems, Inc., 1984-1995

                       CPU: Motorola MVME177.  Processor #0.
                      Memory Size: 0x1000000.  BSP version 1.0.

Executing startup script /usr/aos/alfa/etc/HBARStartup.cmd ...
#============================================================#
#
# Adaptive Optics Associates
#   54 CambridgePark Drive
#  Cambridge, MA 02140-2308
#     USA
#    (Phone) 617-864-0201
#     (Fax) 617-864-1348
#
#   Copyright (C) 1995 AOA
#     All Rights Reserved
#
#============================================================#
```

```
#
# PROJECT:
#
#HBAR-4265-036
#
# FILE:
#
# HBARStartup.cmd
#
# MODULE:
#
# ETC
#
# DESCRIPTION:
#
# Master VxWorks startup command file
#
#  $Id: HBARStartup.cmd,v 1.10 1996/09/12 06:20:33 andy Exp $
#
#==========================================================#

#
# Environment Variables
#
putenv("AOA_ROOT=/usr/aos/alfa")
value = 0 = 0x0

#
# Hosts
#
hostAdd("hbar","150.214.222.20")
value = -1 = 0xffffffff = _end + 0xfff77591

# ----- change the shell prompt
shellPromptSet("alfa@trantor> ")
value = 507476 = 0x7be54 = _shellHistSize + 0x4

#
# File Systems
#
nfsMount("hbar","/usr/aos", 0);
value = 0 = 0x0
nfsMount("hbar","/data", 0);
value = 0 = 0x0
nfsMount("hbar","/export/home", 0);
value = 0 = 0x0
nfsMount("hbar","/tmp",0)
value = 0 = 0x0

#
# Hydra
#
cd getenv("AOA_ROOT")
```

```
value = 0 = 0x0
< etc/HydraStartup.cmd
#==============================================================#
#
# Adaptive Optics Associates
#   54 CambridgePark Drive
#  Cambridge, MA 02140-2308
#     USA
#    (Phone) 617-864-0201
#     (Fax) 617-864-1348
#
#   Copyright (C) 1995 AOA
#     All Rights Reserved
#
#==============================================================#
#
# PROJECT:
#
#HBAR-4265-036
#
# FILE:
#
# HydraStartup.cmd
#
# MODULE:
#
# ETC
#
# DESCRIPTION:
#
# VxWorks startup command file for the Ariel Hydra software
#
#  $Id: HydraStartup.cmd,v 1.3 1996/07/01 19:40:29 joe Exp $
#
#==============================================================#

#
# Environment Variables
#
putenv("VC40PATH=/usr/aos/alfa/hydra/bin.x40")
value = 0 = 0x0

#
# Load and install the driver
#
cd getenv("AOA_ROOT")
value = 0 = 0x0
cd "hydra"
value = 0 = 0x0

ld < driver/vx68k/hydra-ii/h2dsp.vx
value = 16275036 = 0xf8565c = h2dsp.vx_bss + 0x28
h2DrvInstall()
```

```
value = 8 = 0x8


#
# Add the Hydra Devices
#
h2DevAdd("/vc40a", 0x55800000, h2Pri(5), h2Vec(0xd0))
value = 0 = 0x0
h2DevAdd("/vc40b", 0x55820000, h2Pri(5), h2Vec(0xd4))
value = 0 = 0x0
h2DevAdd("/vc40c", 0x55840000, h2Pri(5), h2Vec(0xd8))
value = 0 = 0x0
h2DevAdd("/vc40d", 0x55860000, h2Pri(5), h2Vec(0xdc))
value = 0 = 0x0
h2DevAdd("/vc40e", 0x55880000, h2Pri(5), h2Vec(0xe0))
value = 0 = 0x0


#
# Load the support modules
#
ld < lib.vx68k/libvc40.o
value = 16314996 = 0xf8f274
ld < bin.vx68k/blinkc40.vx
value = 16244108 = 0xf7dd8c = libvc40.o_bss + 0x540
ld < bin.vx68k/copyc40.vx
value = 16231704 = 0xf7ad18 = blinkc40.vx_bss + 0x590
ld < bin.vx68k/emurst.vx
value = 16232316 = 0xf7af7c = _vc40jtag + 0x624
ld < bin.vx68k/infoc40.vx
value = 16227216 = 0xf79b90 = emurst.vx_bss + 0x594
ld < bin.vx68k/jtagtest.vx
value = 16222608 = 0xf78990 = _option_arg + 0x1cc
ld < bin.vx68k/loadc40.vx
value = 16200252 = 0xf7323c = jtagtest.vx_bss + 0x540
ld < bin.vx68k/trapc40.vx
value = 16199296 = 0xf72e80 = jtagtest.vx_bss + 0x184
ld < bin.vx68k/wipec40.vx
value = 16194548 = 0xf71bf4 = trapc40.vx_bss + 0x590

#
# Start the remote Ariel debugger
#
emurst

Hydra-II jtag circuitry is reset

value = 1 = 0x1
jtagtest

    jtagtest passed -- 1

value = 0 = 0x0
```

```
#ld < bin.vx68k/raxdsd.vx
#sp raxdsd


#
# AOA Modules
#

# Utility
cd getenv("AOA_ROOT")
value = 0 = 0x0
< etc/UtilStartup.cmd
#==============================================================#
#
# Adaptive Optics Associates
#   54 CambridgePark Drive
#  Cambridge, MA 02140-2308
#     USA
#    (Phone) 617-864-0201
#     (Fax) 617-864-1348
#
#   Copyright (C) 1995 AOA
#     All Rights Reserved
#
#==============================================================#
#
# PROJECT:
#
#HBAR-4265-036
#
# FILE:
#
# UtilStartup.cmd
#
# MODULE:
#
# Util
#
# DESCRIPTION:
#
# VxWorks command file to initialize the Utility modules
#
#  $Id: UtilStartup.cmd,v 1.9 1996/08/15 21:57:11 joe Exp $
#
#==============================================================#

cd getenv("AOA_ROOT")
value = 0 = 0x0
cd "bin"
value = 0 = 0x0
```

```
#
# Load the Client/Server library
#
ld < ClientServer.vx.o
value = 16291764 = 0xf897b4


#
# Load the Event Logger
#
ld < AOALog.vx.o
value = 16288764 = 0xf88bfc
AOALog_Init()
value = 0 = 0x0
AOALog_SetRemoteHostname("150.214.222.20")
value = 0 = 0x0
#AOALog_EnableRemoteLogging()
#AOALog_EnableFileLogging("/tmp/aoalog",1)


#
# HydraCom Library
#
cd "/usr/aos/bin"
value = 0 = 0x0
ld < HydraCom.vx.o
value = 16183556 = 0xf6f104
HydraCom_SetTimeout(1)
value = 3 = 0x3

#
# Done
#

# Adaptive Optics subsystem
cd getenv("AOA_ROOT")
value = 0 = 0x0
< etc/AOStartup.cmd
#=============================================================#
#
# Adaptive Optics Associates
#   54 CambridgePark Drive
#  Cambridge, MA 02140-2308
#     USA
#    (Phone) 617-864-0201
#     (Fax) 617-864-1348
#
#   Copyright (C) 1995 AOA
#     All Rights Reserved
#
#=============================================================#
#
# PROJECT:
```

```
#
#HBAR-4265-036
#
# FILE:
#
# AOStartup.cmd
#
# MODULE:
#
# AO
#
# DESCRIPTION:
#
# VxWorks command file to start the AO Subsystem
#
#  $Id: AOStartup.cmd,v 1.6 1996/06/10 19:38:08 joe Exp $
#
#==============================================================#

#
# Port LLCCD camera is connected to
#
putenv("LLCCD_PORT=/tyCo/2")
value = 0 = 0x0


#
# Load the modules
#
cd getenv("AOA_ROOT")
value = 0 = 0x0
cd "bin"
value = 0 = 0x0

ld < AODevice.vx.o
value = 16163204 = 0xf6a184
ld < AOServer.vx.o
value = 16160196 = 0xf695c4


#
# Initialize
#
AO_Init()
INFO: AO: (0x0000): AO_Init: Initializing...
INFO: AO: (0x0000): AO_Init: Cleanly resetting...
INFO: AO: (0x0000): AO_Term: Terminating...
INFO: AO: (0x0000): AO_Term: Terminating LLCCD Camera...
INFO: AO: (0x0000): AO_Term: Terminating Camera I/F...
INFO: AO: (0x0000): AO_Term: Terminating Offset & Gain...
INFO: AO: (0x0000): AO_Term: Terminating WFS...
INFO: AO: (0x0000): AO_Term: Terminating RTR...
INFO: AO: (0x0000): AO_Term: Terminating Control System...
```

```
INFO: AO: (0x0000): AO_Term: Terminating DM I/F...
INFO: AO: (0x0000): AO_Term: Done.
INFO: AO: (0x0000): AO_Init: Loading DSP executables from: /usr/aos/alfa/bin
INFO: AO: (0x0000): AO_Init: Initializing LLCCD Camera...
INFO: AO: (0x0000): AO_Init: Starting LLCCD Camera...
INFO: AO: (0x0000): AO_Init: Stopping LLCCD Camera...
INFO: AO: (0x0000): AO_Init: Initializing Camera I/F...
INFO: AO: (0x0000): AO_Init: Initializing Offset & Gain...
INFO: AO: (0x0000): AO_Init: Initializing WFS...
INFO: AO: (0x0000): AO_Init: Initializing RTR...
INFO: AO: (0x0000): AO_Init: Initializing Control System...
INFO: AO: (0x0000): AO_Init: Initializing DM I/F...
INFO: AO: (0x0000): AO_Init: Spawning DSP monitor...
INFO: AO: (0x0000): AO_Init: Done.
value = 0 = 0x0
AO_Server_Start()
value = 0 = 0x0


#
# Done
#

# Stages
cd getenv("AOA_ROOT")
value = 0 = 0x0
< etc/StageStartup.cmd
#===============================================================#
#
# Adaptive Optics Associates
#   54 CambridgePark Drive
#  Cambridge, MA 02140-2308
#     USA
#    (Phone) 617-864-0201
#     (Fax) 617-864-1348
#
#   Copyright (C) 1995 AOA
#     All Rights Reserved
#
#===============================================================#
#
# PROJECT:
#
#HBAR-4265-036
#
# FILE:
#
# StageStartup.cmd
#
# MODULE:
#
# Stage
#
```

```
# DESCRIPTION:
#
# VxWorks command file to start the Stage Subsystem
#
#  $Id: StageStartup.cmd,v 1.8 1996/09/12 06:26:36 andy Exp $
#
#===========================================================#

#
# Port IMS483 Steppermotor Driver is connected to
#
# This is serial Port 2 on the 712M breakout card.
#
putenv("IMS_PORT=/tyCo/1")
value = 0 = 0x0


#
# Load the modules
#
cd getenv("AOA_ROOT")
value = 0 = 0x0
cd "bin"
value = 0 = 0x0

ld < StageDevice.vx.o
value = 16047868 = 0xf4defc = _stageInitSync + 0x230
ld < StageServer.vx.o
value = 15911996 = 0xf2cc3c = StageServer.vx.o_bss + 0x94
ld < Stage.vx.o
value = 15913776 = 0xf2d330 = Stage.vx.o_bss + 0x8
ld < Stage_debug.vx.o
value = 15914376 = 0xf2d588 = Stage.vx.o_bss + 0x260


#
# Initialize
#
stage_debugFlag = 0
_stage_debugFlag = 0xf33870: value = 0 = 0x0
Stage_Init()
INFO: Stage: (0x0000): Stage_Init(): initializing Stage Subsystem
INFO: Stage: (0x0000): Stage_Init(): starting the device driver
INFO: Stage: (0x0000): Stage_Init(): starting the stage server
value = 0 = 0x0

#
# The following command is necessary only if the previous command is
# commented out.  Stage_Init() already does
# StageDevice_Init(port)
# Stage_Server_Stop()
# Stage_Server_Start()
#
```

```
#Stage_Server_Start()

#
# Done
#

CS_DisableDebugLog()
value = 3 = 0x3

#
# EPICS
#
cd getenv("AOA_ROOT")
value = 0 = 0x0
< epics/startup.trantor
#
# VxWorks startup script to load and execute system (iocCore) software.
#
# First, vxWorks startup issues:

# Network stuff: Change to your default router if required
# routeAdd "0", "127.0.0.1"

# Login stuff: If you use vxWorks security you might use one of the following
loginUserAdd "alfa", "bQbyyRbdzS"
value = -1 = 0xffffffff = _cofferr + 0xff07a3a3

# < /usr/local/VxWorks/etc/passwd

# Make delete key work for backspace
tyBackspaceSet 0177
value = 3 = 0x3

# cd to application install directory (beware automounter problems)
cd "/usr/aos/alfa/epicsR3.12.2"
value = 0 = 0x0
pwd
/usr/aos/alfa/epicsR3.12.2
value = 27 = 0x1b

# Load EPICS core and record stuff
ld < epics/base/bin/mv177/iocCore
value = 15947008 = 0xf35500 = _asLock + 0x234
ld < epics/base/bin/mv177/drvSup
value = 15902396 = 0xf2a6bc = _epvxiResourceMangerOK + 0x324
ld < epics/base/bin/mv177/recSup
value = 16145632 = 0xf65ce0 = _epvxiLibDeviceList + 0x938
ld < epics/base/bin/mv177/devSupIms
value = 15348736 = 0xea3400 = _ims_command_qid + 0x52c

# If using state programs, load the sequencer
ld < epics/base/bin/mv177/seq
value = 15346224 = 0xea2a30
```

```
# Load any application records, device support or drivers here,
# or code for subroutine records
# ld < ./bin/mv177/testsub.o

# Load sequences here
ld < ./bin/mv177/imsseq
value = 15343716 = 0xea2064

# Configure the Timestamp system, 1=Master IOC
TSconfigure 1, 0, 0, 0, 0, 0
value = 0 = 0x0

# Load one or more ASCII databases here
dbLoad "./data/default.dctsdr"
value = 0 = 0x0
dbLoadRecords "./data/aoaStages.db"
value = 0 = 0x0
dbLoadRecords "./data/aoaStatus.db"
value = 0 = 0x0

# This turns off logging
iocLogDisable=1
_iocLogDisable = 0xf0e788: value = 1 = 0x1

iocInit "./data/resource.def"
############################################################################
###  @(#)EPICS IOC CORE
###  @(#)Version R3.12.2patch1 Date: 1996/04/02
############################################################################
devIms483: Initializing stage fsmTilt (b)
devIms483: Initializing stage fsmTip (a)
devIms483: Initializing stage fsrfs (i)
devIms483: Initializing stage fsrfz (j)
devIms483: Initializing stage hlas (c)
devIms483: Initializing stage rlm (d)
devIms483: Initializing stage tcx (f)
devIms483: Initializing stage tcy (g)
devIms483: Initializing stage tcz (h)
devIms483: Initializing stage tfw (k)
devIms483: Initializing stage wfs (e)
devIms483: Initializing stage wfsfw (l)
0xf91f6c (tShell): iocInit: All initialization complete
value = 0 = 0x0

# Start sequences
seq &imsseq
@(#)SEQ Version 1.9.0(3.12.1): Tue Jul 23 12:09:37 MET DST 1996
tShell 11/20/96 12:24:49: Spawning state program "imsseq", task name = "imsseq"
tShell 11/20/96 12:24:49:   Task id = 14378276 = 0xdb6524
value = 14378276 = 0xdb6524
# end of AOA EPICS startup
```

```
#
# Done
#
printf("\n**\n** DONE WITH SYSTEM INITIALIZATION\n**\n\n")

**
** DONE WITH SYSTEM INITIALIZATION
**

value = 43 = 0x2b = '+'

imsseq 11/20/96 12:24:49: Spawning task 16313092: "imsseq_1"
imsseq 11/20/96 12:24:49: Spawning task 14327324: "imsseq_2"

Done executing startup script /usr/aos/alfa/etc/HBARStartup.cmd
alfa@trantor>
```

2. console@keahua boot log

```
                    VxWorks System Boot


Copyright 1984-1994  Wind River Systems, Inc.





CPU: Motorola MVME177
Version: 5.2
BSP version: 1.0
Creation date: Thu Jul 20 17:05:04 PDT 1995




Press any key to stop auto-boot...
 1 0
auto-booting...


boot device          : ei
processor number     : 0
host name            : hbar
file name            : /vw/vw-5.2/config/mv177/vxWorks
inet on ethernet (e) : 150.214.222.104:ffffff00
host inet (h)        : 150.214.222.20
gateway inet (g)     : 150.214.222.1
user (u)             : alfa
```

flags (f)   : 0x8
target name (tn)   : keahua
startup script (s)   : /EPICS/charmAppl/startup.keahua

Attaching network interface ei0... done.
Attaching network interface lo0... done.
Loading... 489404 + 30260 + 27582
Starting at 0x1000...

Attaching network interface ei0... done.
Attaching network interface lo0... done.
Loading symbol table from hbar:/vw/vw-5.2/config/mv177/vxWorks.sym ...done

```
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
     ]]]]]]]]]]]  ]]]]     ]]]]]]]]]]       ]]              ]]]]         (R)
 ]     ]]]]]]]]]  ]]]]]]     ]]]]]]]]       ]]              ]]]]
 ]]     ]]]]]]]  ]]]]]]]]     ]]]]]] ]       ]]              ]]]]
 ]]]     ]]]]] ]    ]]]  ]     ]]]] ]]]   ]]]]]]]]]  ]]]] ]] ]]]]  ]]   ]]]]]
 ]]]]      ]]]  ]]    ]  ]]]     ]] ]]]]] ]]]]]]   ]] ]]]]]]] ]]]] ]]   ]]]]
 ]]]]]     ]  ]]]]     ]]]]]      ]]]]]]]] ]]]]   ]] ]]]]    ]]]]]]]    ]]]]
 ]]]]]]      ]]]]]     ]]]]]]    ]  ]]]]]  ]]]]   ]] ]]]]    ]]]]]]]]    ]]]]
 ]]]]]]]    ]]]]]  ]    ]]]]]]  ]    ]]]   ]]]]   ]] ]]]]    ]]]] ]]]]    ]]]]
 ]]]]]]]]  ]]]]]  ]]]    ]]]]]]]      ]     ]]]]]]]  ]]]]    ]]]]  ]]]] ]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       Development System
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks version 5.2
 ]]]]]]]]]]]]]]]]]]]]]]]]]]       KERNEL: WIND version 2.4
 ]]]]]]]]]]]]]]]]]]]]]]]]]       Copyright Wind River Systems, Inc., 1984-1995
```

CPU: Motorola MVME177. Processor #0.
Memory Size: 0x1000000. BSP version 1.0.

Executing startup script /EPICS/charmAppl/startup.keahua ...
#
VxWorks startup script to load and execute EPICS system (iocCore) software
and other CHARM (alfa release) software
Stefan Hippler
Change log:
17-Sep-1996, ALFA, hbar integration started
19-Sep-1996, stages software added
23-Sep-1996, load esters software
1-Oct-1996, home directory now /export/home/alfa/CHARM
19-Nov-1996, ut, rdate and tnetDev added
AOA_ROOT set to /usr/aos/alfa (was /usr/aos/hbar)
#
#

----- create user, password table

```
loginUserAdd "alfa", "bQbyyRbdzS"
value = 0 = 0x0
loginUserAdd "hippler", "cQQcRez9Qz"
value = 0 = 0x0

hostAdd "hbar", "150.214.222.20"
value = -1 = 0xffffffff = _end + 0xfff79651

# ----- setup an User ID for Remote File Access via NFS
# uid of alfa
nfsAuthUnixSet("hbar", 220, 200, 0)
value = 0 = 0x0

# ----- change the shell prompt
shellPromptSet("alfa@keahua> ")
value = 499236 = 0x79e24 = _shellHistSize + 0x4

# ----- mount hbar disks
nfsMount "hbar","/"
value = 0 = 0x0
nfsMount "hbar","/data"
value = 0 = 0x0
nfsMount "hbar","/tmp"
value = 0 = 0x0
nfsMount "hbar","/export/home"
value = 0 = 0x0
nfsMount "hbar","/usr/aos"
value = 0 = 0x0
# -----

# define EPICS application root
EROOT = "/EPICS/charmAppl"
new symbol "EROOT" added to symbol table.
EROOT = 0xf95f00: value = 16328708 = 0xf92804

# define AOA root
putenv("AOA_ROOT=/usr/aos/alfa")
value = 0 = 0x0

# define ESTERS device (temperature measure gizmo) application root
putenv ("ESTERS_ROOT=/export/home/hippler/ALFA/ESTERS")
value = 0 = 0x0

#
# First, vxWorks startup issues:

# Network stuff: Change to your default router if required
# routeAdd "0", "127.0.0.1"

# Login stuff: If you use vxWorks security you might use one of the following
# loginUserAdd "user", "GaRbAgEpWd"
# < /usr/local/VxWorks/etc/passwd
```

```
# Make delete key work for backspace
tyBackspaceSet 0177
value = 3 = 0x3

# load AOA LOG utility
cd getenv("AOA_ROOT")
value = 0 = 0x0
< etc/UtilStartup.cmd.MPIA
#============================================================#
#
# Adaptive Optics Associates
#   54 CambridgePark Drive
#  Cambridge, MA 02140-2308
#     USA
#    (Phone) 617-864-0201
#     (Fax) 617-864-1348
#
#   Copyright (C) 1995 AOA
#     All Rights Reserved
#
#============================================================#
#
# PROJECT:
#
#HBAR-4265-036
#
# FILE:
#
# UtilStartup.cmd.MPIA
#
# MODULE:
#
# Util
#
# DESCRIPTION:
#
# VxWorks command file to initialize the Utility modules
#
#  $Id: UtilStartup.cmd,v 1.9 1996/08/15 21:57:11 joe Exp $
#
#============================================================#

cd getenv("AOA_ROOT")
value = 0 = 0x0
cd "bin"
value = 0 = 0x0


#
# Load the Client/Server library
#
ld < ClientServer.vx.o
value = 16252844 = 0xf7ffac = ClientServer.vx.o_bss + 0x8
```

```
#
# Load the Event Logger
#
ld < AOALog.vx.o
value = 16293612 = 0xf89eec
AOALog_Init()
value = 0 = 0x0
AOALog_SetRemoteHostname("150.214.222.20")
value = 0 = 0x0
#AOALog_EnableRemoteLogging()
rm "/tmp/MPIA_STAGES.log"
value = 0 = 0x0
AOALog_EnableFileLogging("/tmp/MPIA_STAGES.log",1)
value = 0 = 0x0

#
# Done
#

# load stages control software
cd getenv("AOA_ROOT")
value = 0 = 0x0
< etc/StageStartup.cmd
#==============================================================#
#
# Adaptive Optics Associates
#   54 CambridgePark Drive
#  Cambridge, MA 02140-2308
#     USA
#    (Phone) 617-864-0201
#     (Fax) 617-864-1348
#
#   Copyright (C) 1995 AOA
#     All Rights Reserved
#
#==============================================================#
#
# PROJECT:
#
#HBAR-4265-036
#
# FILE:
#
# StageStartup.cmd
#
# MODULE:
#
# Stage
#
# DESCRIPTION:
#
```

```
# VxWorks command file to start the Stage Subsystem
#
#  $Id: StageStartup.cmd,v 1.8 1996/09/12 06:26:36 andy Exp $
#
#==========================================================#

#
# Port IMS483 Steppermotor Driver is connected to
#
# This is serial Port 2 on the 712M breakout card.
#
putenv("IMS_PORT=/tyCo/1")
value = 0 = 0x0


#
# Load the modules
#
cd getenv("AOA_ROOT")
value = 0 = 0x0
cd "bin"
value = 0 = 0x0

ld < StageDevice.vx.o
value = 16261176 = 0xf82038 = _stageInitSync + 0x230
ld < StageServer.vx.o
value = 16257696 = 0xf812a0
ld < Stage.vx.o
value = 16184436 = 0xf6f474 = Stage.vx.o_bss + 0x8
ld < Stage_debug.vx.o
value = 16185036 = 0xf6f6cc = Stage.vx.o_bss + 0x260


#
# Initialize
#
stage_debugFlag = 0
_stage_debugFlag = 0xf759b4: value = 0 = 0x0
Stage_Init()
INFO: Stage: (0x0000): Stage_Init(): initializing Stage Subsystem
INFO: Stage: (0x0000): Stage_Init(): starting the device driver
INFO: Stage: (0x0000): Stage_Init(): starting the stage server
value = 0 = 0x0

#
# The following command is necessary only if the previous command is
# commented out.  Stage_Init() already does
# StageDevice_Init(port)
# Stage_Server_Stop()
# Stage_Server_Start()
#
#Stage_Server_Start()
```

```
#
# Done
#

# load esters software
cd getenv("ESTERS_ROOT")
value = 0 = 0x0
< estLoad
# -----------------------------------
# estLoad, Stefan Hippler, 15-Jul-96
# -----------------------------------
#
unld "estIniExternals.o"
value = -1 = 0xffffffff = EROOT + 0xff06a0ff
unld "estIo.o"
value = -1 = 0xffffffff = EROOT + 0xff06a0ff
unld "estHl.o"
value = -1 = 0xffffffff = EROOT + 0xff06a0ff
unld "estStat.o"
value = -1 = 0xffffffff = EROOT + 0xff06a0ff
#
ld <estIniExternals.o
value = 16269584 = 0xf84110 = estIniExternals.o_bss + 0x8
estIniExternals(2)
value = 0 = 0x0
#
ld <estIo.o
value = 16270204 = 0xf8437c = estIniExternals.o_bss + 0x274
ld <estHl.o
value = 16218348 = 0xf778ec = estHl.o_bss + 0x8
ld <estStat.o
value = 16266572 = 0xf8354c = estIo.o_bss + 0x54c
#

# load astromed user interface (aui) and CHARM/PZT serial control modules
< /vw/STARTUP.keahua
#*******************************************************************
# STARTUP.keahua, Stefan Hippler
# AstroCam, vxWorks/VME/MV167-33MHz System --- VxWorks Startup Procedure
# loaded either from VxWorks boot script or from EPICS startup script
# "startup.keahua"
# Change log:
#   13-Oct-95, first release without LQS package
#   26-Oct-95, new directory structure added
#    4-Nov-95, cleanup
#    8-Jan-95, Calar Alto Implementation
#   29-May-96, modified to new waipi disk layout
#    1-Sep-96, CHARM/SERIAL now on /EPICS/charmAppl/src/SERIAL
#   17-Sep-96, hbar setup started
#    1-Oct-96, new home directory: /export/home/alfa
#    4-Oct-96, nfs mounts now in /EPICS/charmAppl/startup.keahua
#   19-Nov-96, host epsalfa1 added
#-----------------------------------------------------------------------
```

```
# ----- create user, password table

#loginUserAdd "alfa", "bQbyyRbdzS"
#loginUserAdd "hippler", "cQQcRez9Qz"

#hostAdd "hbar", "150.214.222.20"
hostAdd "epsalfa1", "150.214.222.85"
value = 0 = 0x0

# ----- setup an User ID for Remote File Access via NFS
# uid of alfa
#nfsAuthUnixSet("hbar", 220, 200, 0)

# ----- change the shell prompt
shellPromptSet("alfa@keahua> ")
value = 499236 = 0x79e24 = _shellHistSize + 0x4

# ----- mount hbar disks
#nfsMount "hbar","/"
#nfsMount "hbar","/data"
#nfsMount "hbar","/export/home"
#nfsMount "hbar","/usr/aos"

# ----- setup ASTROMED environment variable
putenv("ASTROMED=/ASTROMED")
value = 0 = 0x0

# ----- setup INMOSROOT environment variable
putenv("INMOSROOT=/INMOSROOT")
value = 0 = 0x0

# ----- setup VLTROOT environment variable
putenv("VLTROOT=/VLTROOT")
value = 0 = 0x0

cd getenv("VLTROOT")
value = 0 = 0x0
cd "vw/bin"
value = 0 = 0x0

# ----- install the VLT lculog library
ld < lculog
value = 16268464 = 0xf83cb0 = estIo.o_bss + 0xcb0

# ----- install the VLT lcudrv library
ld < lcudrv
value = 16265364 = 0xf83094 = estIo.o_bss + 0x94

# ----- install ESO's inmos software for VxWorks
cd getenv("INMOSROOT")
value = 0 = 0x0
cd "bin"
```

```
value = 0 = 0x0
<b016Install.sh > /export/home/alfa/LOG/keahua/b016Install.sh.LOG
ld <timsSetUTC.o
value = 15983608 = 0xf3e3f8 = timsSetUTC.o_bss + 0x8
ld <timsGetUTC.o
value = 15982720 = 0xf3e080 = timsGetUTC.o_bss + 0x8
<inmosInstallIserver.sh > /export/home/alfa/LOG/keahua/inmosInstallIserver.sh.LOG
<inmosInstallTool.sh > /export/home/alfa/LOG/keahua/inmosInstallTool.sh.LOG

# install CHARM software
cd "/EPICS/charmAppl/src/SERIAL"
value = 0 = 0x0
<charmLoad
# -----------------------------------
# charmLoad, Stefan Hippler, 9-Jan-96
# -----------------------------------
#
ld <charmIniExternals.o
value = 15988072 = 0xf3f568 = charmIniExternals.o_data + 0x308
#
charmIniExternals(2)
value = 1 = 0x1
#
ld <charmIni.o
value = 15954940 = 0xf373fc = charmIni.o_bss + 0x324
ld <charmSend.o
value = 15952864 = 0xf36be0 = charmSend.o_bss + 0x8
ld <charmCrash.o
value = 15956736 = 0xf37b00 = charmCrash.o_bss + 0x8
ld <charmPS.o
value = 15957348 = 0xf37d64 = charmCrash.o_bss + 0x26c
ld <charmPSNowait.o
value = 15954292 = 0xf37174 = charmIni.o_bss + 0x9c
ld <charmTipTilt.o
value = 15950328 = 0xf361f8 = charmTipTilt.o_bss + 0x8
ld <testPort1_1.o
value = 15949488 = 0xf35eb0 = testPort1_1.o_bss + 0x118
ld <testPort1_2.o
value = 15947496 = 0xf356e8 = ITOOLS + 0x5ec
ld <testPort1_3.o
value = 15946984 = 0xf354e8 = ITOOLS + 0x3ec
ld <cpm.o
value = 15951276 = 0xf365ac = charmPSNowait.o_bss + 0xa4
ld <rpm.o
value = 15941060 = 0xf33dc4 = iserver_bss + 0x978
#

# install 42gci interface and boot TRAM and Astromed transputer
cd getenv("ASTROMED")
value = 0 = 0x0
cd "42gci"
value = 0 = 0x0
```

```
<42gciInstall.sh.CHARM > /export/home/alfa/LOG/keahua/
42gciInstall.sh.CHARM.LOG
#<42gciInstall.sh.CHARM

# oOo

# cd to application install directory (beware automounter problems)
cd "/export/home/hippler/CHARM/charmEpicsR3.12.2/Sep96"
value = 0 = 0x0
pwd
/export/home/hippler/CHARM/charmEpicsR3.12.2/Sep96
value = 51 = 0x33 = '3'

# Load EPICS core and record stuff
ld < epics/base/bin/mv177/iocCore
value = 16000984 = 0xf427d8 = _asLock + 0x84
ld < epics/base/bin/mv177/drvSup
value = 16002788 = 0xf42ee4 = _epvxiResourceMangerOK + 0x324
ld < epics/base/bin/mv177/recSup
value = 15809468 = 0xf13bbc
ld < epics/base/bin/mv177/devSupIms
value = 15806948 = 0xf131e4 = _ims_command_qid + 0x52c

# If using state programs, load the sequencer
ld < epics/base/bin/mv177/seq
value = 15025016 = 0xe54378

# Load any application records, device support or drivers here,
# or code for subroutine records
# ld < ./bin/mv177/testsub.o

# Load sequences here
ld < ./bin/mv177/auiseq
value = 15022508 = 0xe539ac = _stopPressed + 0x244
#ld < ./bin/mv177/charmseq
ld < ./bin/mv177/imsseq
value = 14131768 = 0xd7a238 = auiseq_bss + 0x7c4

# Configure the Bancom driver (read documentation)
# BCconfigure 1, 0, 200, 2, 25

# Configure the Timestamp system, 1=Master IOC
TSconfigure 1,0,0,0,0,0
value = 0 = 0x0

# Load one or more ASCII databases here
dbLoad "./data/default.dctsdr"
value = 0 = 0x0
dbLoadRecords "./data/auidb.db"
value = 0 = 0x0
dbLoadRecords "./data/charmdb.db"
value = 0 = 0x0
dbLoadRecords "./data/aoaStages.db"
```

```
value = 0 = 0x0
dbLoadRecords "./data/mpiaStages.db"
value = 0 = 0x0

# This turns off logging
iocLogDisable=1
_iocLogDisable = 0xe1fa4c: value = 1 = 0x1

iocInit "./data/resource.def"
############################################################################
###  @(#)EPICS IOC CORE
###  @(#)Version R3.12.2patch1 Date: 1996/04/02
############################################################################
devIms483: Initializing stage f10Focus (m)
devIms483: Initializing stage dichroicTip (u)
devIms483: Initializing stage dichroicTv (t)
devIms483: Initializing stage f10Tilt (q)
devIms483: Initializing stage f10Tip (o)
devIms483: Initializing stage fibreX (r)
devIms483: Initializing stage fibreY (p)
devIms483: Initializing stage tiptiltMirrorY (n)
devIms483: Initializing stage tvGuiderY (s)
0xf919fc (tShell): iocInit: All initialization complete
value = 0 = 0x0

# Start sequences

seq &auiseq, "stack=50000"
@(#)SEQ Version 1.9.0(3.12.1): Tue Jul 23 12:09:37 MET DST 1996
tShell 11/20/96 12:58:13: Spawning state program "auiseq", task name = "auiseq"
tShell 11/20/96 12:58:13:   Task id = 13321636 = 0xcb45a4
value = 13321636 = 0xcb45a4
taskDelay(sysClkRateGet())
auiseq 11/20/96 12:58:13: Spawning task 13271236: "auiseq_1"
auiseq 11/20/96 12:58:13: Spawning task 13220836: "auiseq_2"
auiseq 11/20/96 12:58:13: Spawning task 13170436: "auiseq_3"
auiseq 11/20/96 12:58:13: Spawning task 13120036: "auiseq_4"
value = 0 = 0x0

#seq &charmseq
#taskDelay(sysClkRateGet())

seq &imsseq
@(#)SEQ Version 1.9.0(3.12.1): Tue Jul 23 12:09:37 MET DST 1996
tShell 11/20/96 12:58:14: Spawning state program "imsseq", task name = "imsseq"
tShell 11/20/96 12:58:14:   Task id = 13069636 = 0xc76d44
value = 13069636 = 0xc76d44
taskDelay(sysClkRateGet())
imsseq 11/20/96 12:58:14: Spawning task 13022028: "imsseq_1"
imsseq 11/20/96 12:58:14: Spawning task 13011628: "imsseq_2"
imsseq 11/20/96 12:58:14: Spawning task 13001228: "imsseq_3"
imsseq 11/20/96 12:58:14: Spawning task 12990828: "imsseq_4"
imsseq 11/20/96 12:58:14: Spawning task 12980300: "imsseq_5"
```

```
imsseq 11/20/96 12:58:14: Spawning task 12949448: "imsseq_6"
imsseq 11/20/96 12:58:14: Spawning task 12969848: "imsseq_7"
imsseq 11/20/96 12:58:14: Spawning task 12938976: "imsseq_8"
imsseq 11/20/96 12:58:14: Spawning task 12856492: "imsseq_9"
value = 0 = 0x0

# terminal server module
ld < ./bin/mv177/tnetDev
value = 15816176 = 0xf155f0

#
_dmINIOK = 0
new symbol "_dmINIOK" added to symbol table.
_dmINIOK = 0xc4032c: value = 0 = 0x0
ld < ./bin/mv177/dmBoot
value = 12846028 = 0xc403cc = _dmINIOK + 0xa0

# time / UT
ld < ./bin/mv177/ut
value = 12845372 = 0xc4013c = ut_bss + 0x8

# rdate
ld < ./bin/mv177/rdate
value = 12843644 = 0xc3fa7c = rdate_bss + 0x33c
#

# Start fabry perot software
# pwd
# < STARTUP.FPI

# cd to CHARM's working directory
cd "/export/home/alfa/CHARM"
value = 0 = 0x0
pwd
/export/home/alfa/CHARM
value = 24 = 0x18


Done executing startup script /EPICS/charmAppl/startup.keahua
alfa@keahua>
```