

DESCRIPTION OF VOODOO V1.7 FEATURES

By Scott Streit
June 27, 2003

VOODOO INSTALLATION PROCEDURES AND ISSUES

Voodoo requires a JAVA 2 runtime environment to function. We currently use JDK1.3. To ease the installation process, install scripts are provided for all operating systems. The current scripts have greatly improved installation.

Solaris and Linux Installation

- Make sure the \$PATH environment variable in your .cshrc or .login file contains the path to your JAVA 2 bin directory:

```
/opt/JDK_1.2.2_05/bin/
```

- Create a temporary directory and unpack the Voodoo files (**do not untar the voodoofiles.tar file**):

```
gunzip voodoo.tar.gz
tar xvf voodoo.tar
```

- Become superuser and run the install script:

```
./Install
```

- Enter the directory where you want Voodoo installed. The default is /opt/voodoo.
- Modify your .cshrc or .login to contain the following, where `$install_dir` is the directory where the Voodoo program files are installed and `$system` is either "unix" or "linux":

```
setenv LD_LIBRARY_PATH $install_dir/Voodoo/Clib/$system/lib
setenv CLASSPATH
$install_dir/Voodoo/classes:$install_dir/packages
```

- To un-install Voodoo, just delete the directory where Voodoo was installed. The default is /opt/voodoo.

Windows 2000 Installation

- Create a temporary directory and unzip the Voodoo files.
- Run the file "Setup.exe".
- Make sure your systems \$PATH environment variable contains the path to your JAVA 2 bin directory:

C:\JDK_1.2.2_05\bin\

Installation Issues

Despite the easy installation, errors may still occur and the most common are:

1. UNIX and LINUX users must manually set the CLASSPATH and LD_LIBRARY_PATH environment variables on their system. Setting an incorrect path will result in the JAVA classes not being found.

```
setenv CLASSPATH /opt/voodoo/Voodoo/classes:/opt/voodoo/packages:.
setenv LD_LIBRARY_PATH /opt/voodoo/Voodoo/Clib/unix/lib:.
```

2. LINUX users must remember to manually set aside image buffer RAM by editing /etc/lilo.conf. More on this later.
3. My computer works fine with Version 1.4, but crashes now that I downloaded and installed the Version 1.7 driver and voodoo?
 - The PCI and Timing board ROMS need to be upgraded. This involves running PciRomBurn and TimRomBurn.

4. I get an *Error 31* after installing the Windows 2000 driver?

- The *astropci.sys* file is in the wrong directory. Move it from C:\WINNT to C:\WINNT\System32\drivers.

4. The result of running Voodoo on Windows 2000 is:

```
java -Xrs Voodoo C:\Voodoo\Voodoo ARCPNP1
java.lang.IllegalArgumentException: Signal already used by VM:
SIGINT
    at sun.misc.Signal.handle(Unknown Source)
    at java.lang.Terminator.setup(Unknown Source)
    at java.lang.Shutdown.add(Unknown Source)
    at java.lang.Runtime.addShutdownHook(Unknown Source)
    at sun.awt.windows.WToolkit.run(Unknown Source)
    at java.lang.Thread.run(Unknown Source)
```

Try removing the "-Xrs" from the "voodoo.bat" file. Or try changing ARCPNP1 to astropci1.

6. When I run Voodoo I get an "Unsatisfied Link Error"?

- This means the path to Voodoo's C library is not set. On UNIX and LINUX, make sure you have the following defined in your .cshrc file, where \$install_dir is your installation path and \$system is either "unix" or "linux":

```
setenv LD_LIBRARY_PATH $install_dir/Voodoo/Clib/$system/lib
```

On Windows 2000, make sure your "path" environment variable contains the following:

```
$install_dir/Voodoo/Clib/win2k/lib
```

7. When I run Voodoo I get "CLASSPATH not found"?

- This means the path to the Voodoo classes is not properly set. On UNIX and LINUX, make sure you have the following defined in your .cshrc file, where \$install_dir is your installation path:

```
setenv CLASSPATH  
$install_dir/Voodoo/classes:$install_dir/packages
```

On Windows 2000, make sure you create an environment variable called CLASSPATH that contains the following:

```
$install_dir/Voodoo/classes;$install_dir/packages
```

8. I have SAOImage/Ximtool/DS9 installed, but when I click "Display Image" nothing happens?

- You must have IRAF installed for this feature to work. IRAF is not available for MS Windows.

DEVICE DRIVER INSTALLATION PROCEDURES AND ISSUES

Solaris Installation

- ★ Unpack the device driver files, this will create a directory called astropciV1.7 (or something similar):

```
gunzip astropci_solaris.tar.gz  
tar xvf astropci_solaris.tar
```

- ★ Become superuser and run the install script:

```
./Install
```

- ★ Enter the PCI slot number when prompted. For multiple boards, continue to enter PCI slot numbers as prompted. Hit "return" when finished.

- ★ To remove the driver, use the unix command: `rem_drv astropci`

Linux Installation

- ★ This driver has been tested under Redhat 7.2 (kernel 2.4.2-2) and Redhat 9.0 (kernel 2.4.20-8). It WILL NOT work with older kernels.
- ★ In order for this driver to function, the following line must be set in LILO (/etc/lilo.conf):
append="mem=xxxM", where xxx is the amount of RAM you DO NOT want to use for an image buffer. So, if your computer has 128Mb of RAM and you want to have a 28Mb image buffer, you must have the following line in LILO (/etc/lilo.conf):
append="mem=100M".

Example:

1. Become superuser/root.

2. Edit /etc/lilo.conf to have the append="mem=xxxM" line:

```
image=/boot/vmlinuz-2.4.2-2
label=linux
read-only
root=/dev/hda5
append="mem=100M"
```

3. From a tcsh, execute the command: lilo -v

4. Reboot the computer.

- ★ Create the directory (/xxx) where you want to keep the driver files.

- ★ Copy the driver tar file to the new directory and unpack it:

```
gunzip astropci_x_x_x.tar.gz
tar xvf astropci_x_x_x.tar
```

- ★ Become superuser and run the install script:

```
./astropci_load
```

IMPORTANT NOTE: The Linux driver currently has no support for loading the driver at system startup. So the ./astropci_load script must be run after every system boot.

- ★ To unload the driver, type:

```
./astropci_unload
```

Windows 2000 Installation

★ Unzip the device driver files.

IMPORTANT: If you are re-installing or upgrading the driver you must first delete the windows cached .inf and .pnf files. During installation, Windows stores its own copy of the driver, which it will use the next time you try to re-install. To prevent Windows from using this old copy you must delete it before re-installing or upgrading. The files are generally called oem0.inf, oem1.inf, ... and oem0.pnf, oem1.pnf, ... And can be found in C:\Windows\inf\. **Be sure to only delete the oem# files that contain references to Astronomical Research Cameras, Inc.**

★ The Win2k driver supports Plug-N-Play. So there are two methods for installation.

1. Install the PCI board and startup the computer, the new board should be detected and you will be prompted for the driver installation.
2. From control panel, choose "Add New Hardware", again you will be prompted for the driver installation.
3. Reboot.

★ NOTE: The driver install will modify the Windows registry to support a large image buffer. This MUST be done. The default amount of memory to set aside is 32M (0x2000000). If you want more memory, you can modify it by doing the following:
WARNING: Incorrectly modifying registry values can cause your machine to stop working.

1. Run *regedit* from *Start->Run*.
2. In the registry, modify
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session
Manager\Memory Management\NonPagedPoolSize to be a value larger than the
buffer you want to use. DO NOT make the value more than 80% of your total
RAM. For example, to set aside 32M of nonpaged memory, set the value to
0x2000000.
3. Reboot for changes to take affect.

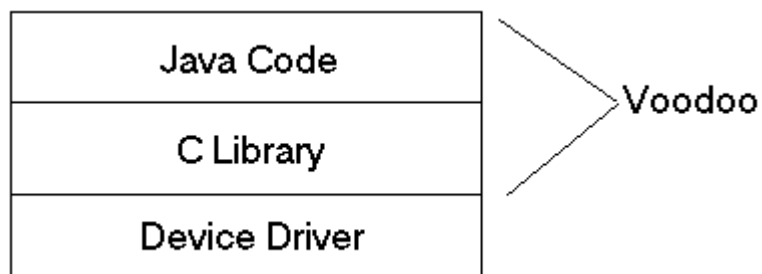
REVISION COMPATIBILITY

Software revisions are completely INCOMPATIBLE. If Voodoo is upgraded from an older version to a newer one, then all existing DSP code must also be upgraded to match the version number of the newer Voodoo. All software (DSP, device driver, and Voodoo) are released together in one matching version. Mixing code from different versions may have unpredictable results. For example, migrating from version 1.4 to 1.7 requires users to download Voodoo V1.7, device driver V1.7, and V1.7 PCI, timing, and utility files. The PCI and timing boards must have their ROMs burned with the new code. To aid in this process, we supply two programs, PciRomBurn and TimRomBurn. PciRomBurn will allow users to burn new PCI code on the PCI EEPROM. Similarly, TimRomBurn will allow new timing board code to be burned.

IMPORTANT: For all versions before 1.7 (2003), all *.setup*, *.par*, and *.VoodooChild.curse* files must be tossed away and new ones generated. These files are not compatible among revisions before 2003 versions of V1.7.

APPLICATION STRUCTURE

Voodoo consists of two layers, the JAVA layer and the C library layer. Voodoo's C library supports device driver level system calls (because JAVA doesn't) and any other operations which may be difficult or too slow with JAVA (for example, memory mapping of the image buffer). While JAVA is platform independent, the C system calls are not. So the C library must be compiled separately on the supported operating systems. Likewise, the device driver is also system dependent. While the underlying system calls and structure are different for the device drivers on the UNIX, LINUX, and Windows 2000 operating systems, the interface between Voodoo and the device drivers is identical across all systems.



CONTROLLER SETUP

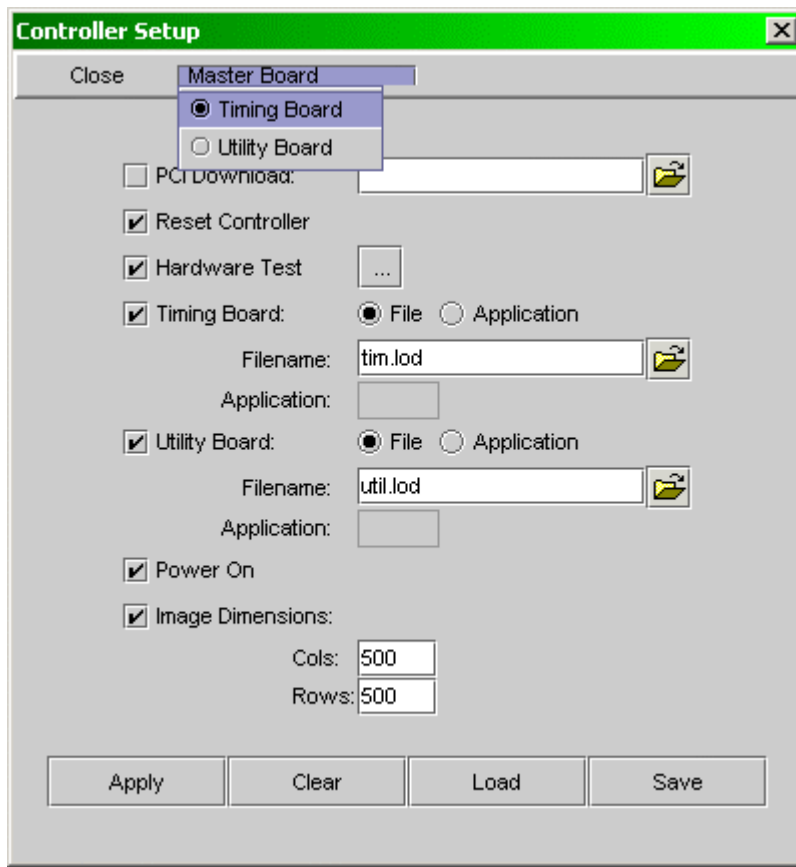
The setup procedure follows the Setup Window components from top to bottom. A minimal setup of the image dimensions must be applied before an exposure can be started.

Timing or Utility – Who is Master?

The Setup Window allows users to designate either the Timing or Utility board as “master”. The master board is the one that will control the exposure sequence. The following commands are dependent on the “master” switch:

1. OSH – Open Shutter (manually)
2. CSH – Close Shutter (manually)
3. PON – Power On
4. POF – Power Off
5. AEX – Abort Exposure
6. PEX – Pause Exposure
7. REX – Resume Exposure
8. SEX – Start Exposure
9. SET – Set Exposure Time (timing board only, utility uses RDM/WRM)
10. RET – Read Elapsed Time (timing board only, utility uses RDM/WRM)
11. Setting Shutter Position (during an exposure)

Example: Master equals timing board.



CONTROLLER CONFIGURATION PARAMETERS

The timing board DSP code on every controller contains a 24-bit word known as the *Controller Configuration*. The bits of this word determine what hardware options are available. These options are ONLY available after a timing board file or application has been downloaded on the controller.

Voodoo reads the controller configuration word by sending an RCC command to the timing board at the end of a controller setup sequence. A tabbed window is then created based on the word bits. Each tab pane corresponds to one of the available hardware options and allows parameters to be modified. For the parameters to be sent to the timing board, users must click "Apply Above" for each parameters tab they wish to modify. This prevents unintentional modification of values that may potentially cause harm to the system.

NOTE: If no configuration word exists, the default values are used by Voodoo, which are NONLINEAR TEMPERATURE CONVERSION, TIMING BOARD REV4B, UTILITY BOARD REV3, and SHUTTER available.

The file TIMHDR.ASM contains the bit specification for the controller configuration and is as follows:

; The bit is set (=1) if the capability is supported by the controller

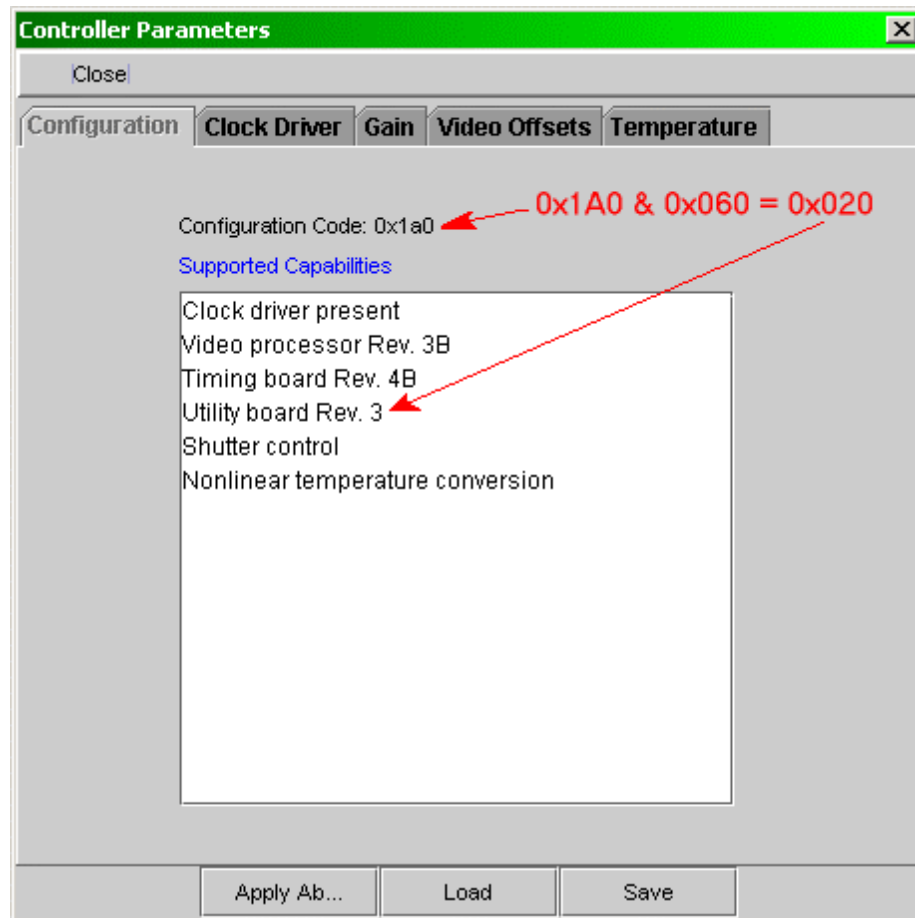
BIT #'s	FUNCTION	BIT #'s	FUNCTION
2,1,0	Video Processor	11	Binning
000	CCD Rev. 3	0	Not supported
001	CCD Gen I	1	Yes supported
010	IR Rev. 4		
011	IR Coadder	12	Split-Serial readout
		0	Not supported
4,3	Timing Board	1	Yes supported
00	Rev. 4, Gen II		
01	Gen I	13	Split-Parallel readout
		0	Not supported
6,5	Utility Board	1	Yes supported
00	No utility board		
01	Utility Rev. 3	14	MPP = Inverted parallel clocks
		0	Not supported
7	Shutter	1	Yes supported
0	No shutter support		
1	Yes shutter support	16,15	Clock Driver Board
		00	Rev. 3
9,8	Temperature readout	11	No clock driver (Gen I)
00	No temperature readout		
01	Polynomial Diode calibration		
10	Linear sensor calibration	19,18,17	Special implementations
		000	Somewhere else
10	Subarray readout	001	Mount Laguna Observ.
0	Not supported	010	NGST Aladdin
1	Yes supported	xxx	Other

The timing board DSP file contains default values for all available hardware configurations. Voodoo allows users to modify these default values.

- **IMPORTANT:** Modified values are not permanently saved on the timing board EEPROM and will need to be applied every time one of the following occurs:
 - Controller power is cycled.
 - The controller setup is applied and the "Load Controller Parameters" checkbox is not selected.
- To expedite this process, Voodoo allows users to save all parameters to the controller setup file, which can later be reloaded by checking the "Load Controller Parameters" in the Setup window before applying the setup. All non-blank controller parameters are automatically saved in the controller setup file when that file is saved.
- It is important to note that the parameters only directly affect Voodoo in two minor ways, which are:
 - The deinterlace option on Voodoo's Main Window may be changed. For example, selecting quad readout will automatically cause the deinterlace option to select quad deinterlacing.
 - The image dimensions in the Setup Window may be changed by some parameters. Binning, for example, would do this.

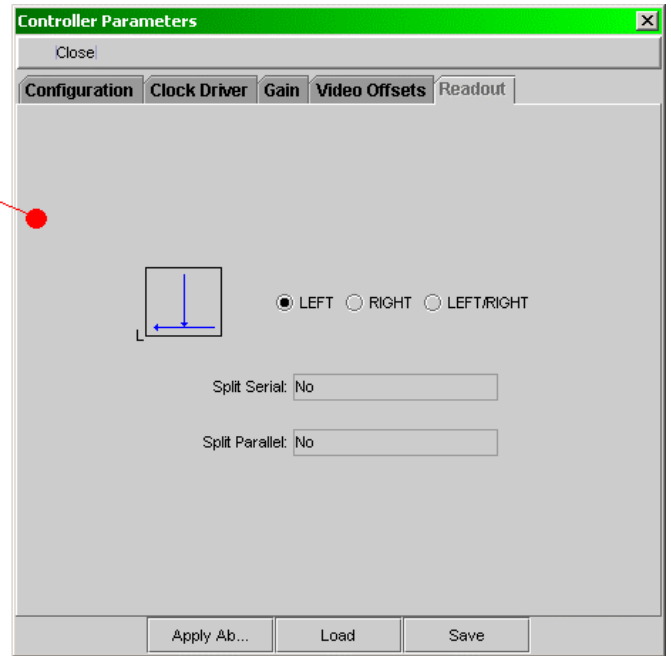
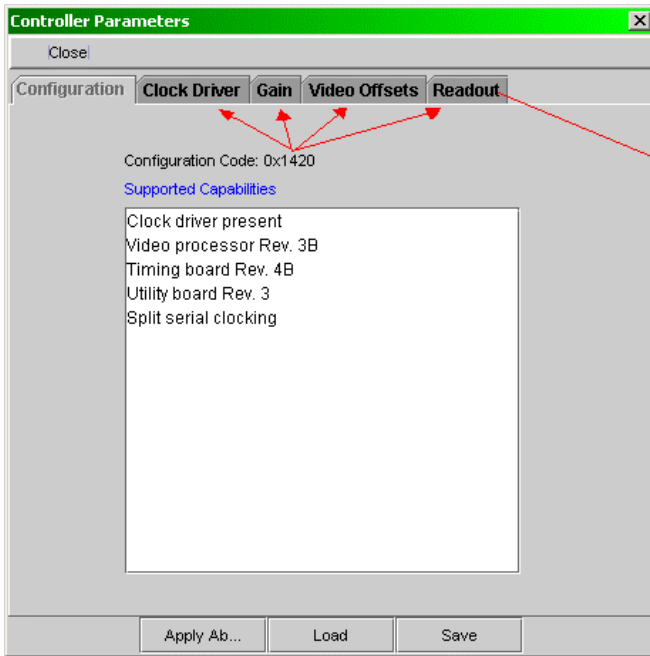
Example 1: Utility Board Exists

The controller returned 0x1A0 as the controller configuration. The utility board support is found by masking 0x1A0 with 0x060. A utility board is supported if: $0x1A0 \& 0x060 = 0x020$. And in this case it does.



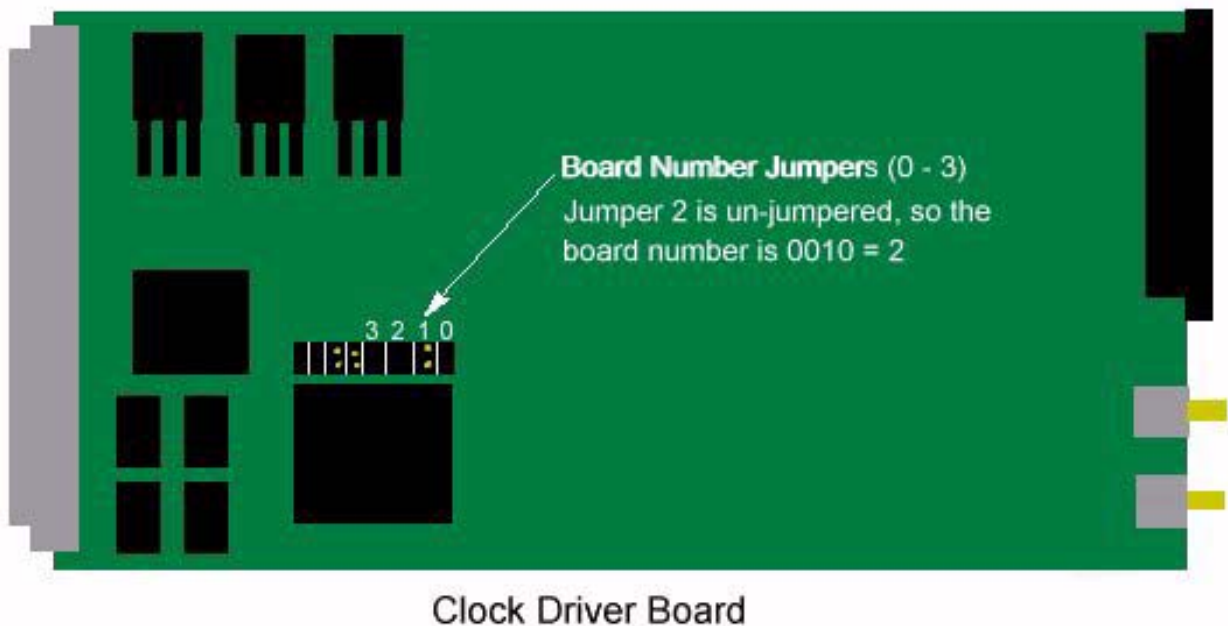
Example 2:

The controller has returned 0x1420, which will cause Voodoo to create the tabbed parameter panes for a Clock Driver (0x0), Video Processor Rev. 3B (0x0), Timing Board Rev. 4B (0x0), Utility Board Rev. 3 (0x20), Split Serial Readout options (0x1000), and Subarray is supported (0x400). $0x20 | 0x1000 | 0x400 = 0x1420$.



Example 3:

The Clock Driver Parameters window allows users to modify the array clock voltages. A simple mistake here is to not select the correct board number. The board number is determined by a set of jumpers on the clock driver board (see diagram). The Clock fields are labels for the user's convenience. The minimum and maximum voltage values must be set before any other voltages can be sent to the controller. Finally, the M1 and M2 buttons determine which of the 24 clocks are associated with the two data signal inputs.



CAMERA BUTTONS

Voodoo has several buttons that allow direct control of certain camera functions. These are primarily used for diagnostics, except for "Read Temperature", which is useful to camera users.

Open Shutter – This button manually opens/closes the shutter (if one exists).

Clear Array – This instructs the timing board to clear (or destructively readout) the array.

Idle Array – The timing board continuously clocks the CCD if it's in idle mode. This button will toggle idle mode on/off.

Power On – This turns the controller power on/off. This is here for diagnostic purposes.

Reset Controller – This turns the controller power off and loads the DSP boot code from ROM. This does not load any DSP application code.

Reset PCI – This instructs the PCI board to reset its program counter (PC) back to its initial position, thus “restarting” the DSP firmware on the PCI board.

Write To Disk – This causes Voodoo to write the contents of the current image buffer to a FITS file. The image dimensions will be those specified in the setup window. How much of the image buffer is actually written to disk depends on the image dimensions specified in the setup window.

Read Temp – This instructs the controller to return the current digital value that represents the arrays temperature. Voodoo converts this digital value into a temperature value. The conversion method used depends on the type of temperature sensor, linear or non-linear.

GLOBAL SETTING AND AUTOMATIC STARTUP

Voodoo allows the user to save and automatically reload all window sizes, exposure options, image buffer size, and the controller setup filename. All this information is saved in the *current directory* to a text file called “.VoodooChild.curse”, by clicking the “Save Settings” button, which is located on the “Global Settings” window under the “Program” menu option. Once this file has been saved, every subsequent start of Voodoo, from that same directory, will cause the file to be read and all values reloaded.

- **IMPORTANT ISSUES:**

- The controller setup file **will** be automatically downloaded to the controller. The controller parameters file **will not** be automatically downloaded to the controller, unless the “Load Controller Parameters” item is checked in the Controller Setup window. Otherwise, the parameters will be loaded into Voodoo, but the user must manually apply the desired settings.
- All *.VoodooChild.curse* files must be deleted and regenerated when switching Voodoo versions, otherwise, you are doomed to suffer incorrect parameter loading or even crashes. This is only true for versions before 1.7.
- All *.VoodooChild.curse* files should be deleted or removed if Voodoo fails to start or behaves strangely during startup.
- If a valid *.VoodooChild.curse* file exists, Voodoo will reallocate the image buffer using the dimensions contained within that file.

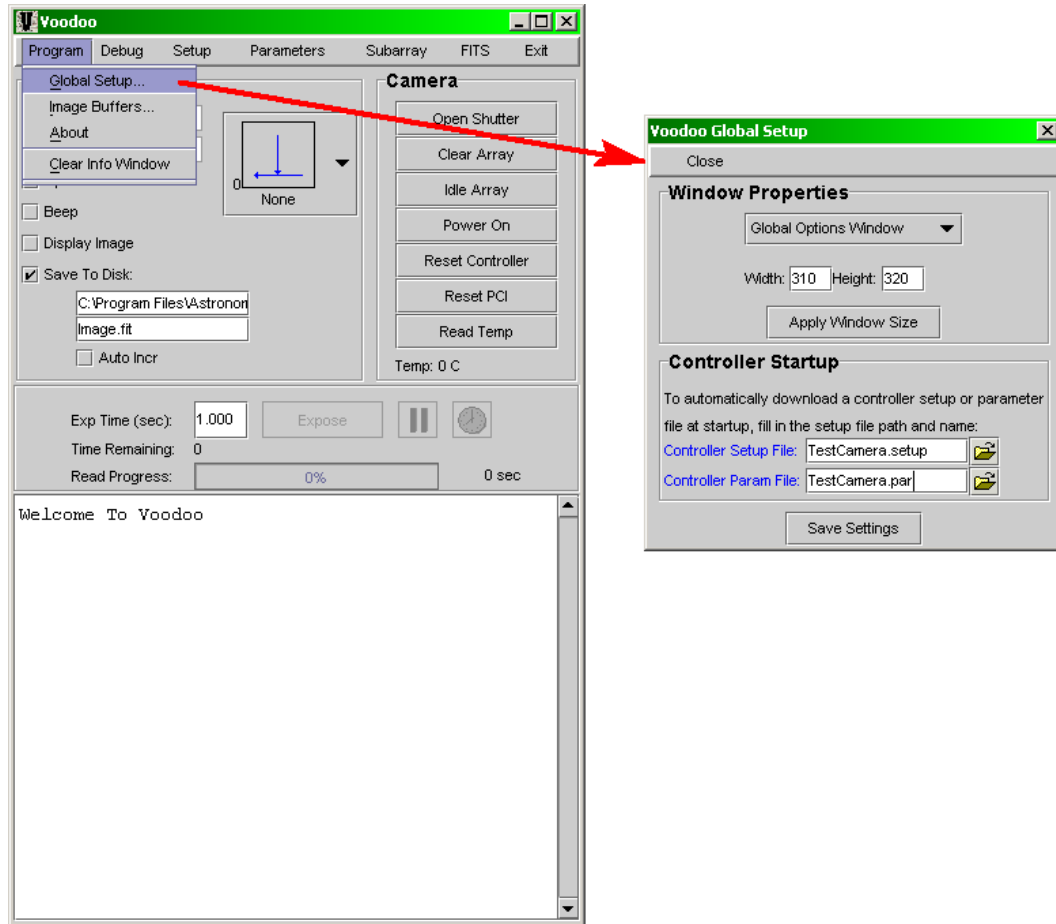


IMAGE BUFFERS

Voodoo has default image buffer dimensions of 2200x2200 pixels. This value can be customized for the duration of the program via the Image Buffers Window. To have Voodoo automatically modify the image buffer dimensions at startup, the user should save a Global Setup File (.VoodooChild.curse) after setting the desired dimensions.

- On Windows 2000, two values limit the maximum image buffer size that can be allocated by Voodoo.
 1. See the document "*Windows 2000/XP Device Driver DMA Limits*" for more info on Windows imposed memory limits.
 2. The device driver has a hard-coded maximum image buffer size of 4200x4200 pixels. This value can only be changed by modifying the device driver and re-compiling.

3. The device driver modifies the registry value "*NonPagedPoolSize*" to be approximately 35Mb, which is large enough to hold a 4200x4200 pixel image. If this value exceeds 60% of your total RAM your machine may become unstable.
- LINUX users must manually set aside RAM for Voodoo to use. This is accomplished by adding the line: "append=mem=100M" to the file /etc/lilo.conf. The 100M is the amount of RAM you **do not** want to use for an image buffer. So this would allow a 28Mb image buffer on a 128Mb RAM machine.

Example:

1. Become superuser/root.
2. Edit /etc/lilo.conf to have the append="mem=xxxM" line:

```
image=/boot/vmlinuz-2.4.2-2
label=linux
read-only
root=/dev/hda5
append="mem=100M"
```

3. From a tcsh, execute the command: lilo -v
4. Reboot the computer.

DEBUG TOOLS

Voodoo provides several debug tools. These can be useful for diagnosing PCI, controller, and software errors.

Disable Read Elapsed Exp Time prevents Voodoo from sending commands to read the controllers elapsed exposure time. This will not prevent Voodoo from reading the image; it will only prevent reading of the elapsed time.

Disable Exposure Timeout prevents Voodoo from timing out if it cannot read the elapsed exposure time from the controller.

Disable Readout Timeout prevents Voodoo from timing out if the current pixel count cannot be read from the controller. Voodoo will never come out of expose if this box is checked and the controller fails to finish image readout. The current pixel count is the number of pixels that have been transferred to the host computer.

Disable Readout Progress prevents Voodoo from reading the current pixel count. The current pixel count is the number of pixels that have been transferred to the host computer.

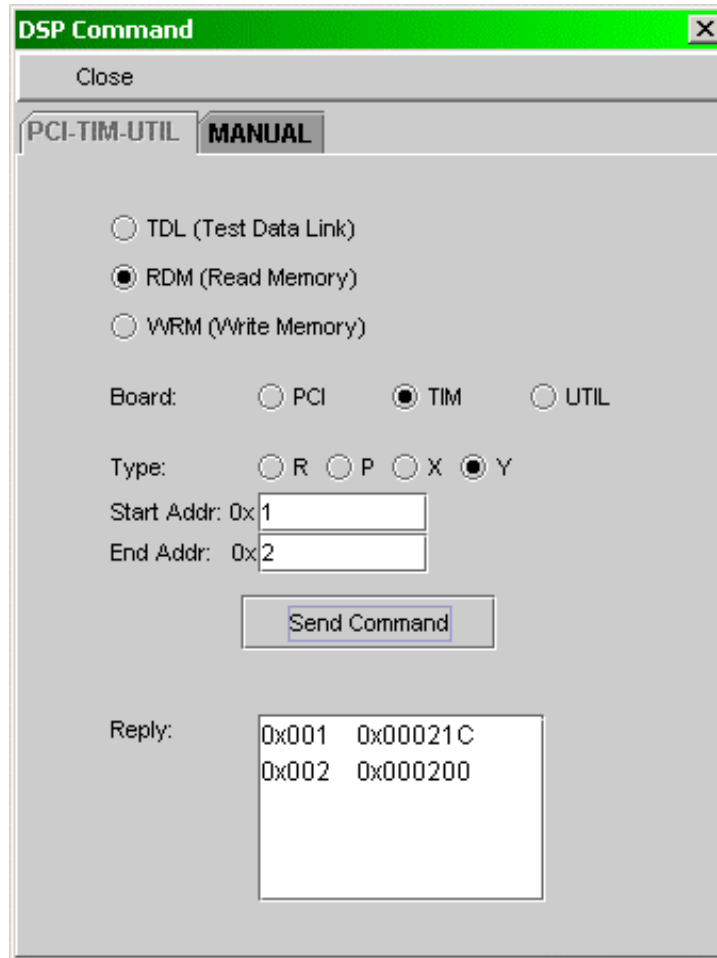
Show Commands will instruct Voodoo to display the details of every command as it is sent. This will display the command, header, parameter and reply values for every command sent. The format is as follows:

```
<header> <command> <arg1> <arg2> <arg3> <arg4> <arg5> <arg6> -> <reply>
```

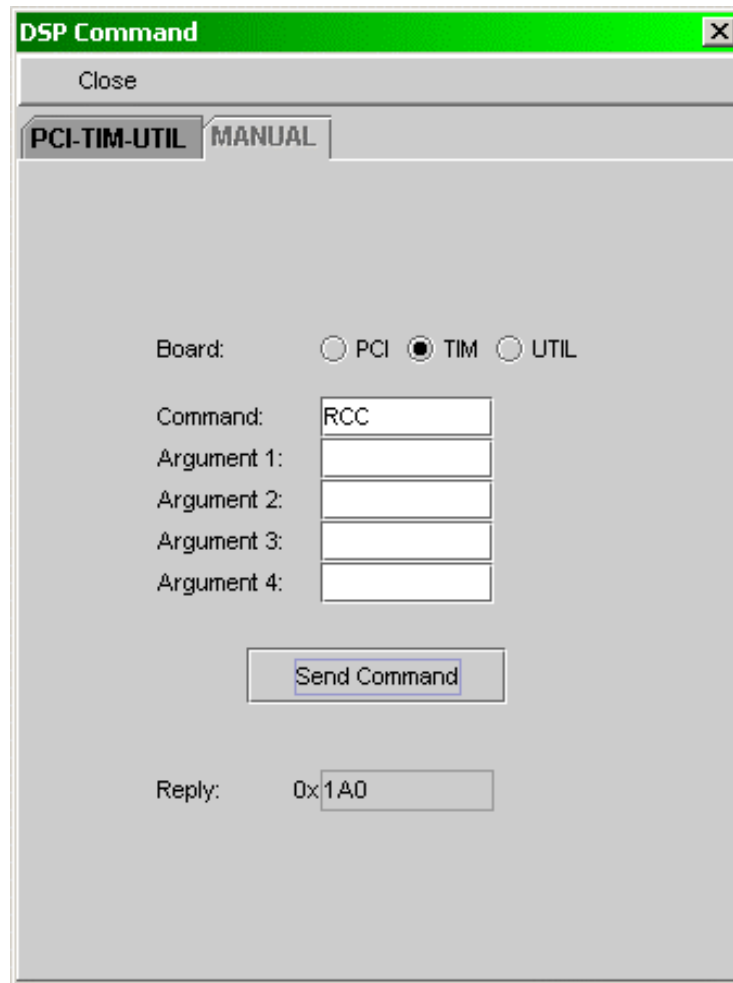
The DSP Command window allows users to send both fixed and user defined commands to the controller and PCI board. The fixed commands that can be sent are TDL (Test Data Link), RDM (Read Memory), and WRM (Write Memory).

- Test Data Link – Tests DSP communications by sending a specified data value to the specified board (Timing, Utility, PCI). The DSP board responds by sending the data value back to Voodoo.
- Read Memory – Allows users to read memory locations on the specified board's DSP (Timing, Utility, and PCI).
- Write Memory – Allows users to write to any valid DSP memory location on the specified board (Timing, Utility, and PCI).
- User Defined Commands – These are 3 character ASCII commands that can be user defined in the controller DSP code or they can be one of the predefined commands, such as IDL (Idle array).

Example: Reading the image dimensions on the controller can be done by sending RDM to the timing board. The image columns are at Y:1 and the image rows are at Y:2.

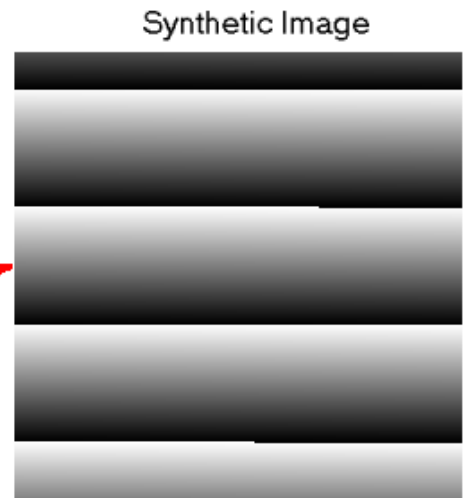
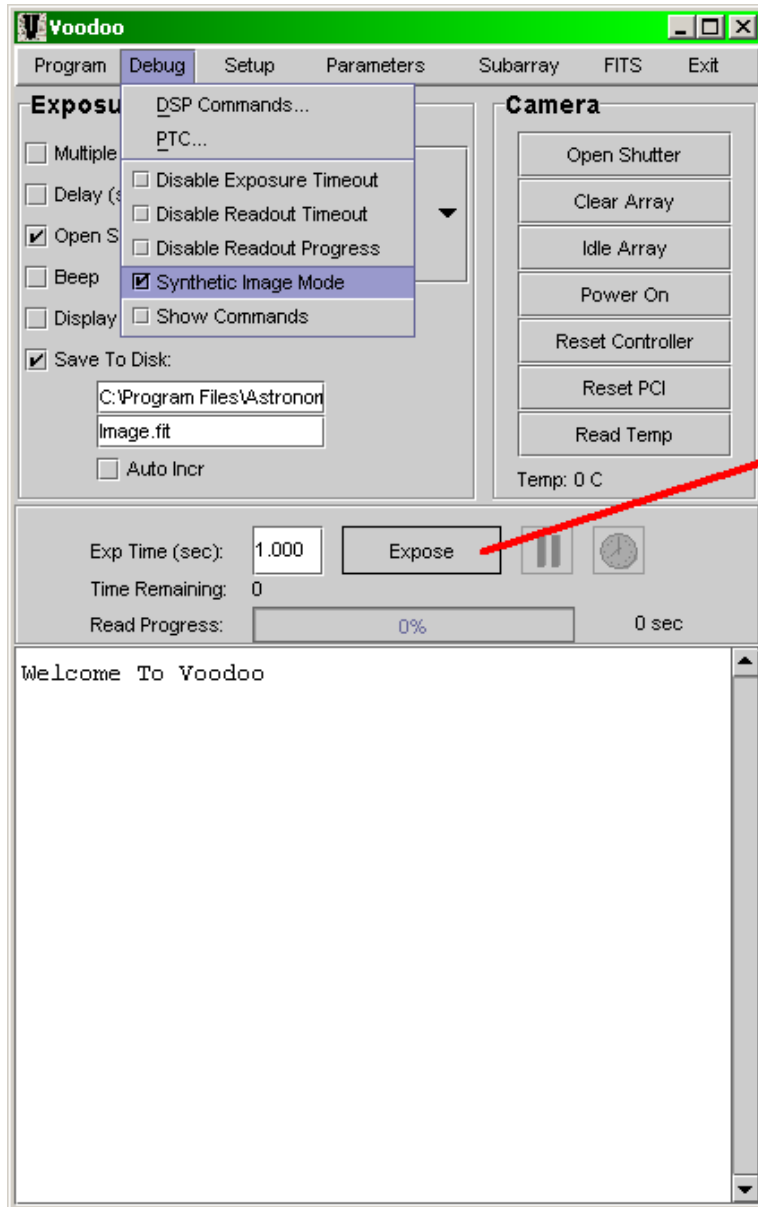


Example: Reading the controller configuration word. This requires sending the RCC command to the timing board. No arguments. This can be used to verify that the correct hardware options are available.



Synthetic Image Mode is used to verify image readout. Selecting this Mode will cause the controller, and in particular the timing board, to generate a preset image on exposure. This allows users to determine if the controller is properly reading out. For example, if no image or an incorrect image is returned from an array on expose, then using synthetic image mode will allow users to know that there may be a problem with the array readout code. Exposure time has no affect on the appearance of synthetic image. The dimensions of the synthetic image will be equal to the set image dimensions. Also, the number of gradients will depend on the set image dimensions. The larger the image, the more gradients there will be.

It is important to note that this is a test of the controller and not the computer or Voodoo. Also, it is possible that your controller may not contain the DSP code required to generate the synthetic image.



Photon Transfer Curve (PTC) plots image signal level (mean counts in ADU's) against the variance (in ADU's) measured between pairs of flat field images of identical exposure times. One of each pair serves as a flat field correction for the other image. The conversion gain is also calculated and is determined from the following equation for total noise:

$$\sigma_i^2 = \sigma_r^2 + k \cdot N_i$$

where σ_i is the variance, σ_r the Poisson readout noise, N_i the number of ADU's detected and $1/k$ the conversion gain in e^-/ADU . The variance is measured as the root mean square of the number of ADU's in the corrected image divided by $\sqrt{2}$ to account for the noise introduced by flat fielding. The conversion gain is obtained from the slope of a linear least squares fit to the plotted

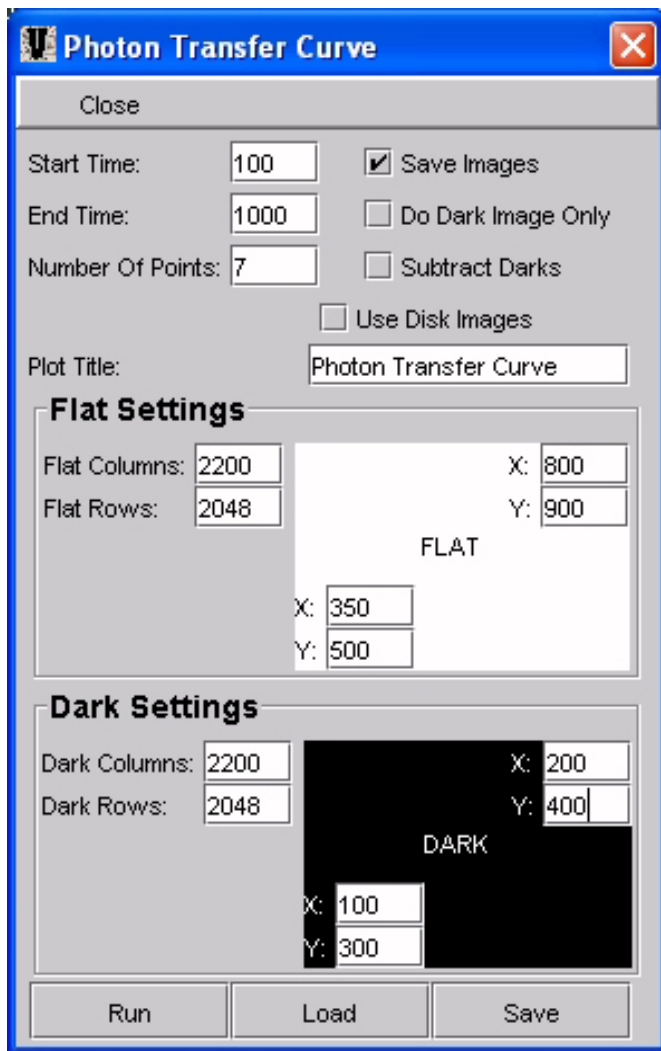
data points (full well turn-over points are not included in the calculation). The full well of the device's pixels is determined from the ADU value where the plot deviates from linear behavior.

PTC Window. The "Start" and "End" times are the exposure times for the first and last flat field images, respectively. The exposure time for the intermediate flat field images is determined by the "Number of Points" to be plotted:

$$\text{Exposure Time} = (\text{End Time} - \text{Start Time}) \div (\text{Number of Points} - 1)$$

The dark image exposure time is set to 0 seconds. The number of flat field exposures taken is just two times the specified "Number of Points" to be plotted. If "Save Images" is checked, then all images will be saved to the local directory. These images can later be used to re-generate the PTC plot by checking the "Use Disk Images" checkbox. If "Do Dark Image Only" is checked, no flat field images will be taken, which is useful if you only want bias statistics. The "Subtract Darks" checkbox causes two dark images to be taken and the average mean calculated (instead of the mean for a single dark image).

The "Flat" and "Dark" settings specify the size of the images and the location on the images where the statistics will be calculated. The "Flat/Dark Columns and Rows" specify the size of the images in pixels. The "X" and "Y" positions mark the lower left and upper right coordinates of the region on the image from where the statistics will be calculated. This region can be anywhere within the image. The "Save" and "Load" buttons allow all values and options to be saved and re-loaded for reuse.

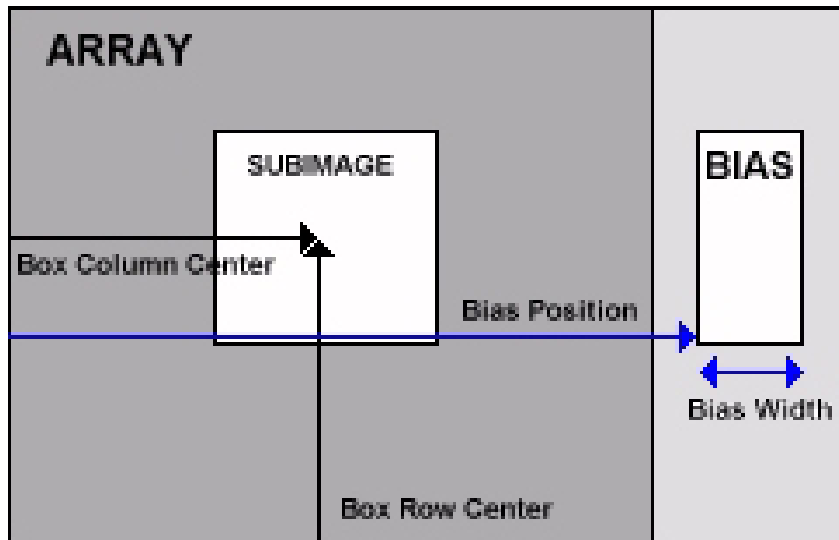


In this figure 7 points will be plotted from 14 images of 2048x2200, with points being generated from the calculations on the 350x400 pixel region located at ((350,500),(800,900)). The situation is similar for the dark image. The flat image exposures will start at 100 ms and end at 1 sec with all intermediate exposures being 212 ms, 424 ms , ...

SUBARRAY

Subarray allows a user to take a sub-image from anywhere on the array. The user specifies the size and position (in pixels) of the sub-image. See the figures below. The box width and height specify the size of the sub-image in pixels. The box column center and the box row center specify the position on the array around which the sub-image will be centered. Bias position sets the offset of the bias region in columns from the left most array column (column 0). Bias width sets the width of the bias region in pixels.

The full array option allows the user to set the image dimensions back to what they were before the subarray dimensions were applied.



Subarray ✕

Close

Full Array
 Subarray

Box Width:
 Bias Position:

Box Height:
 Bias Width:

Box col center:

Box row center:

Apply

Subarray ✕

Close

Full Array
 Subarray

Cols:

Rows:

Apply

FAQ

Below is a list of common user questions.

> What is the maximum number of multiple exposures that can be selected?

None. In terms of the code, the upper limit is about 2 billion.

> What is the maximum delay that can be inputted?

None.

> What are the maximum and minimum exposure times?

Minimum exposure time is 0. Maximum is limited by the controller's 24-bit input in milliseconds (about 4.5 hours).

> What does the de-interlace "Subtract" feature do?

This feature is generally used by IR cameras. But if selected, the final image will be the result of subtracting the first rows/2 pixels from the remaining image buffer pixels. In other words, the upper half of the image buffer is subtracted from the lower half. Generally, for IR cameras the image buffer will consist of two images; this is correlated double sampling for IR cameras.

> In the "Program" menu, what does "Auto Clear Info Window" do?

This option causes Voodoo to clear all the text in the info status window after every 2000 characters. The info status window is the white text pane at the bottom of the main window. If you don't use this option and you run Voodoo for extended periods of time, you will eventually run out of RAM as the text string continues to grow.

> Where is the FITS information, such as date, time, etc., stored. In the header?

Yes, that's correct.

> In Video Offsets, the default readout board setting is 0. What does this affect when changed?

You can have up to 16 video processor boards (32 channels). Board 0 is the default and is the board number in single video processor board systems. The video offsets affect the video signal by subtracting the user specified video offset from the analog video signal. A and B represent the two channels (A/D) per board. See the "CCD Video Processor" manual on our website (www.astro-cam.com) for more info.

> How does "Auto Scale" on Voodoo's image viewer actually function, is it a logarithmic representation, for instance, or some kind of 5-95 cutoff?

It's a linear scale with the minimum and maximum values used for the lower and upper values.

> What does "Select Area" on Voodoo's image viewer do? How do you get rid of the selected area box?

This is primarily meant for use by our observatory for auto-focus. You can remove the box by placing the mouse pointer over it and hitting delete on the keyboard.

> What do the Min/Max slider bars on Voodoo's image viewer do? Are they brightness and contrast, respectively. What do the numbers mean?

No. They represent the upper and lower values for the linear scaling of the image.

ASTROPCI API - C APPLICATION INTERFACE

We offer a C API for users who wish to interface their own application or scripts to the device driver and controller. The API is written in C and contains the basic device driver function calls, which are the building blocks for all controller communications. There are only 5 basic function calls, one to read the PCI DSP command register (HCTR), one to write the PCI DSP command register, one to read the PCI DSP status register (HSTR), one to write a vector command to the PCI board, and one to write all regular ASCII commands. The API also contains functions for reading array temperature, deinterlacing images, storing FITS files, and loading PCI, timing, and utility DSP .lod files.

The API is bundled into the following:

- UNIX – static archive library (.a), dynamic library (.so)
- LINUX – static archive library (.a), dynamic library (.so)
- Windows 2000 – static library (.lib), Dynamic Linked Library (.dll)

The C API is supplied with a sample application called "*apiTest*". It is a fully functioning text based program that exercises the entire API.

```
=====
MAIN MENU
=====
1. Apply Controller Setup
2. Apply Controller Parameters
3. Expose
4. Let It Rip (Do Options 1, 2, 3 With Defaults)
5. Exit

Enter Option: |
```