MityViewer Application Users Manual

Revision 1.3.2 –August 26, 2012

Table of Contents

Preface

The MityViewer application serves a few different purposes. First and foremost, it allows the user to exercise the various calls in the camera DLL (clcamiface3.dll on windows). The intent is to have the program exercise every feature in the DLL. This will allow the user to see how the camera API functions can be used in a real-world scenario. The MityViewer application can also be used to configure and test the camera during development and production. Each application window will be described in its own section of this manual and the DLL calls it uses will be listed.

Revision History

Revision	Date	Notes
-	6 Jul 2009	
1.1.0	16 Jul 2009	Updated for version 1.1.0 of the MityViewer.
1.2.3	8 Jun 2010	Updated for version 2.1.10 MityCCD Release.
1.3.0 - Draft	25 May 2012	Updated for version 2.1.17 MityCAM – sCMOS (beta release)
1.3.1	8 Aug 2012	Updated for Release 1of MityCAM-sCMOS
1.3.2	26 Aug 2012	Various corrections and updates

Getting Started

Installation

Installing the MityViewer application is as simple as running the setup application (MityViewer_setup.exe). This will install the application on your computer and create shortcuts to start it under the standard menus (**Start→Programs→Critical Link→MityViewer** on a windows based computer). During the installation, you will be able to select which plugin elements you wish to install. Some of the plugins are designed for factory and/or development use and will require keys to install.

The installer presents the user with a list of components to install. Please see the sCMOS Application Quick Start Guide for instructions on installing the MityViewer application.

Overview

The MityViewer application is designed as a modular application. The main window provides the basic application shell and core user interface functions. Most of the features of the application are provided by plug-ins. When the application starts, it scans its installation directory for suitable plugins and loads them. These plugins provide various features (camera control, image viewing, data processing, etc). This setup allows the application to be tailored to fit the desired functionality without cluttering it up with features not required by the user (i.e. Internal factory configuration and test plugins are not installed users that just need to use the camera to collect data). The MityViewer application stores its settings in an ini file called “`%APPDATA%\Critical Link LLC\MityViewer.ini`”.

The `%APPDATA%` path is usually `C:\Documents and Settings\<User Name>\Application Data`.

Windows

Main Window

The MityViewer application consists of a main window that serves as a container for the other windows in the program.

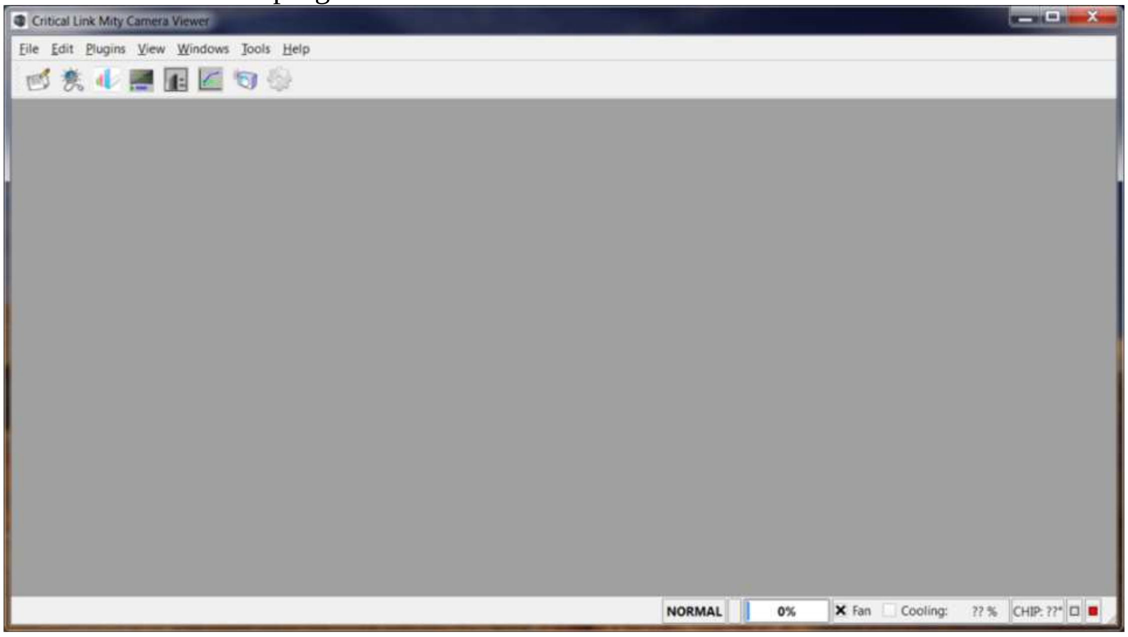

Critical Link Mity Camera Viewer
File Edit Plugins View Windows Tools Help
NORMAL
0%
Fan
Cooling:
?? %
CHIP: ??°

Figure 1 Main Window

It also serves as the main point of control for the camera interface. Many of the main camera callback interfaces are hooked into the main window; including `CLSetErrorCallback()`, `CLSetHeartBeatCallback()`, and `CLSetImageDataCallback()`. A plugin with a user interface will be either a dockable widget or a window. Plugins that are dockable widgets can be arranged by docking them (attaching them) to one of the edges of the main window, or left as floating windows that can be outside of the main window area. Window widgets (like the snapshot and plot view widgets) must remain inside the main application window (although they can be minimized, maximized, tiled, etc). Each loaded plugin will place an icon on the main toolbar (which can also be docked or left floating). The icons on the toolbar are used to show/hide the docking widgets, or create a new window. The main toolbar can be shown/hidden from the “View” menu. Each menu in the application is described below.

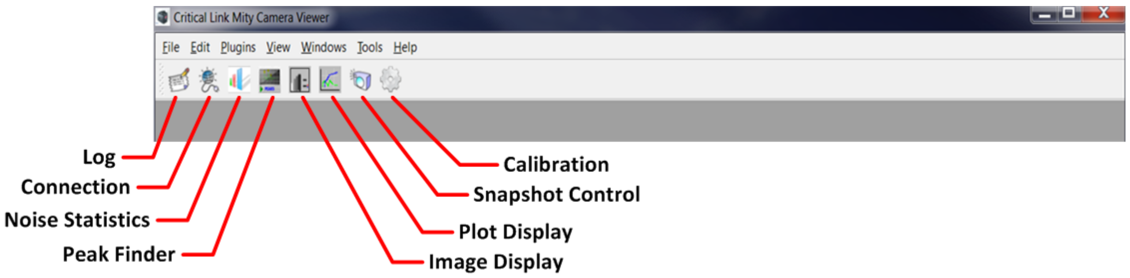


Figure 2 - MityViewer Application Icons

Status Bar

The status bar has a number of controls and indicators that are quite important.

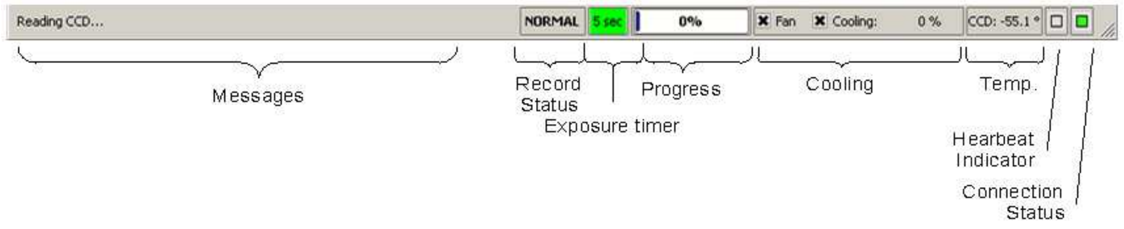
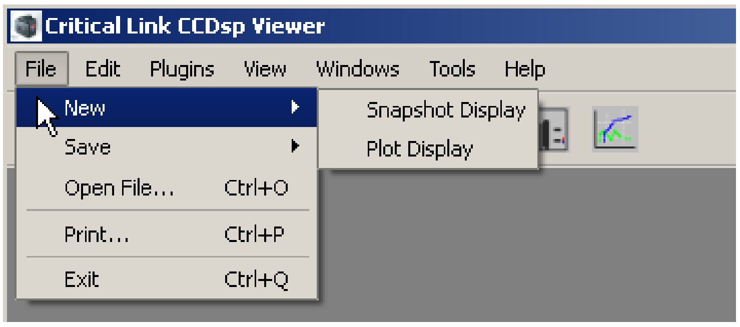


Figure 3 Main Window Status Bar

Application messages are displayed on the left side of the status bar. Important messages are also sent to the log window. The application playback/record state is shown next. This field changes from NORMAL to RECORD or PLAYBACK based on the current state of the application and changes color accordingly. During camera exposure, the remaining exposure time is indicated by the exposure timer label (using information in the camera `tsHeartBeatData` structure). The progress bar is updated upon receipt of every heartbeat message from the camera by calling `CLGetReadProgress()`. The cooling controls allow you to control both the fan (using `CLEnableFan()`) and the TE cooler (using `CLSetCooling()`). There is also a small chart showing the relative amount of cooling being applied (as a percent). If the amount of cooling exceeds 95%, the chart will turn red (versus the normal green color). The actual temperature of the CCD sensor is shown next (in degrees Celsius). The heartbeat indicator blinks every time a heartbeat message is received from the camera, and will turn red if the camera stops sending them (while connected). The connection status indicator will be green, if connected to a camera, or red, if not.

File Menu



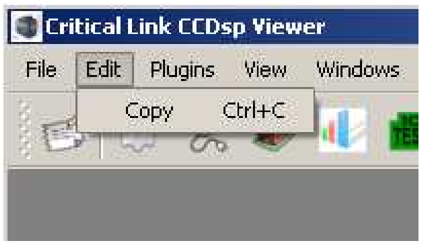
The File menu provides the standard sets of actions you would expect.

Figure 4 File Menu

Table 1 File Menu Actions

Action	Description
New	Provides a sub-menu allowing you to create a new instance of each type of window plugin. These actions are also present on the main toolbar.
Save	If the currently active window is capable of saving its data, this menu entry allows you to do so. It will be disabled if the active window does not support saving data.
Open File…	If the currently active window supports loading data from a file, this entry will do so. Its specific behavior depends on the active window and is documented therein.
Print…	This menu item will print the currently active window or the main window if no window is active.
Exit	Exits the application.

Edit Menu



The Edit menu allows you to copy a window.

Figure 5 Edit Menu

Table 2 Edit Menu Actions

Action	Description
Copy	Copies the active window to the clipboard (or the main window if there is no active window). The default copy behavior is to copy an image of the window. If a specific window behaves differently, it will be documented in its section.

Plugins Menu

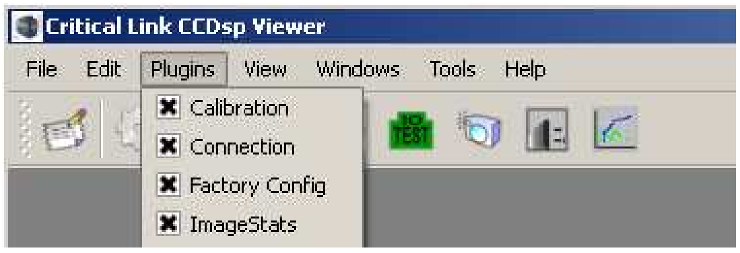


Figure 6 Plugins Menu

The plugins menu shows the currently loaded plugins. If a plugin is checked, it means that it is a dock widget plugin and its corresponding widget has been loaded. Plugins that provide windows do not have a checkbox next to them.

View Menu

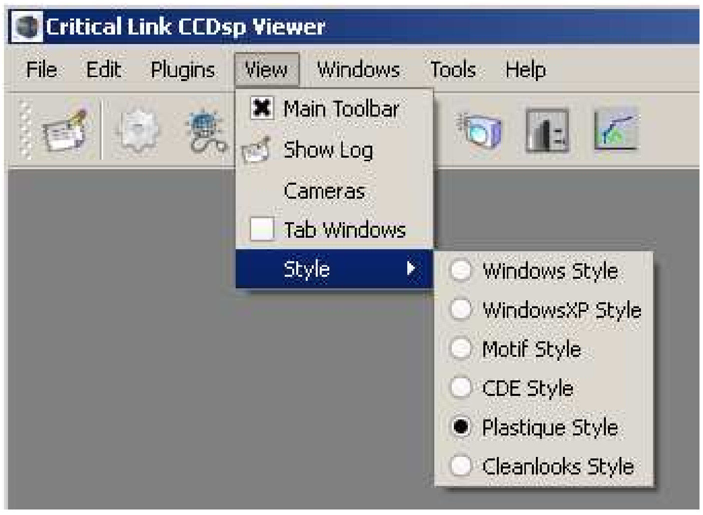


Figure 7 View Menu

The View Menu allows the user to control various visual aspects of the application. Each item on the view menu is described in the table below.

Table 3 View Menu Actions

Action	Description
Main Toolbar	Shows and hides the main toolbar.
Show Log	Shows the application log window.
Cameras	Shows a report of the different types of cameras that have been connected to this application (for troubleshooting).
Playback / Record	Shows the Record/Playback dialog.
Tab Windows	When checked, the application will display the windows as a set of tabs. Uncheck this item to display them as "normal" windows.
Style	Displays a sub-menu allowing you to set the style (visual appearance) of the application. The default style is "Plastique". The selected style will be remembered in the ini file for next time the program is run.

Windows Menu

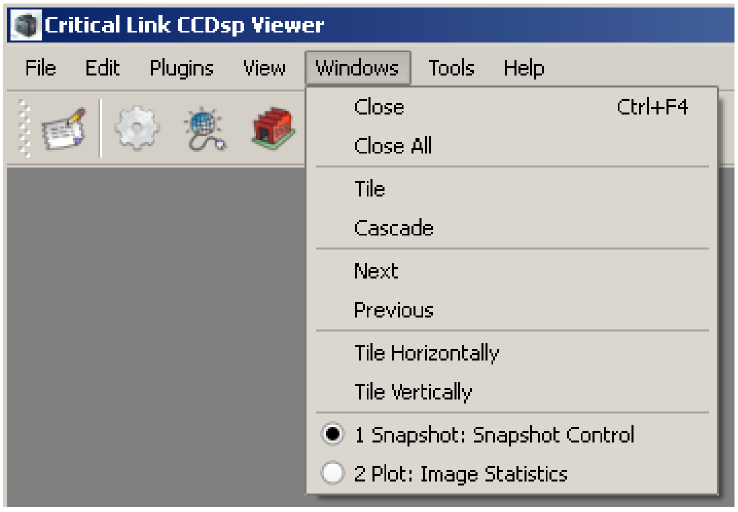


Figure 8 Windows Menu

The windows menu provides the standard set of window manipulation actions you would expect in an application. Each item on the view menu is described in the table below. Note that the windows menu does not affect dock widgets.

Table 4 Windows Menu Actions

Action	Description
Close	Closes the active window.
Close All	Closes all the windows.
Tile	Arranges all the windows in a tile pattern (favoring the active window).
Cascade	Arranges all the windows in a cascade pattern.
Next	Activates the next window. The windows are activated in the order in which they are created.
Previous	Activates the previous window. The windows are activated in the order in which they are created.
Tile Horizontally	Arranges the windows left to right using the full height of the main window area for each window. The windows are arranged in the order in which they are created.
Tile Vertically	Arranges the windows top to bottom using the full width of the main window area for each window. The windows are arranged in the order in which they are created.
Window list	Each window will be listed, allowing the user to directly activate one. The currently active window is indicated.

Tools Menu

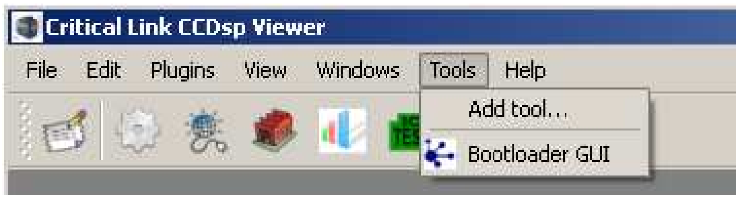


Figure 9 Tools Menu

The tools menu allows you to launch your own tools from within the MityViewer application. Selecting the “Add tool…” item prompts you to select an application or batch file using the standard open file dialog box. If a valid file is selected, you are prompted to give the tool a name for the menu (it defaults to the file name). The program will use the application icon on the menu, if available to help identify the tool to the user. The path to the tool is also shown in the main window status bar as it is highlighted in the menu. To remove a tool from the menu, right-click over its menu entry and select the “Remove Tool” option from the context menu. Information about items in the tool menu is stored in the application ini file.

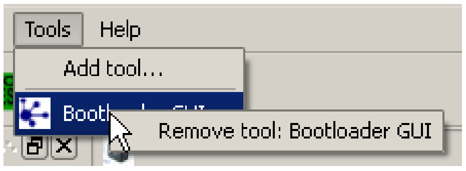


Figure 10 Remove tool menu

Help Menu

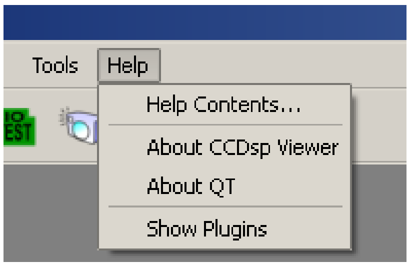


Figure 11 Help Menu

The Help menu provides access to this manual and various application information used when reporting an issue with the program. Each item on the help menu is described in the table below.

Table 5 Help Menu Actions

Action	Description
Help Contents…	Shows this manual.
About MityViewer	Shows the about box for this application, including version information used for bug reporting.
About QT	Shows the QT about box (The MityViewer application is built using the QT application framework.)
Show Plugins	Shows the list of loaded plugins and information about them. Plugins that did not successfully load are also shown in this dialog.

Snapshot Viewer Window

The snapshot viewer window is the primary means of viewing data acquired by the camera. The window displays a spectrogram plot of the acquired data using a user-selectable color map.

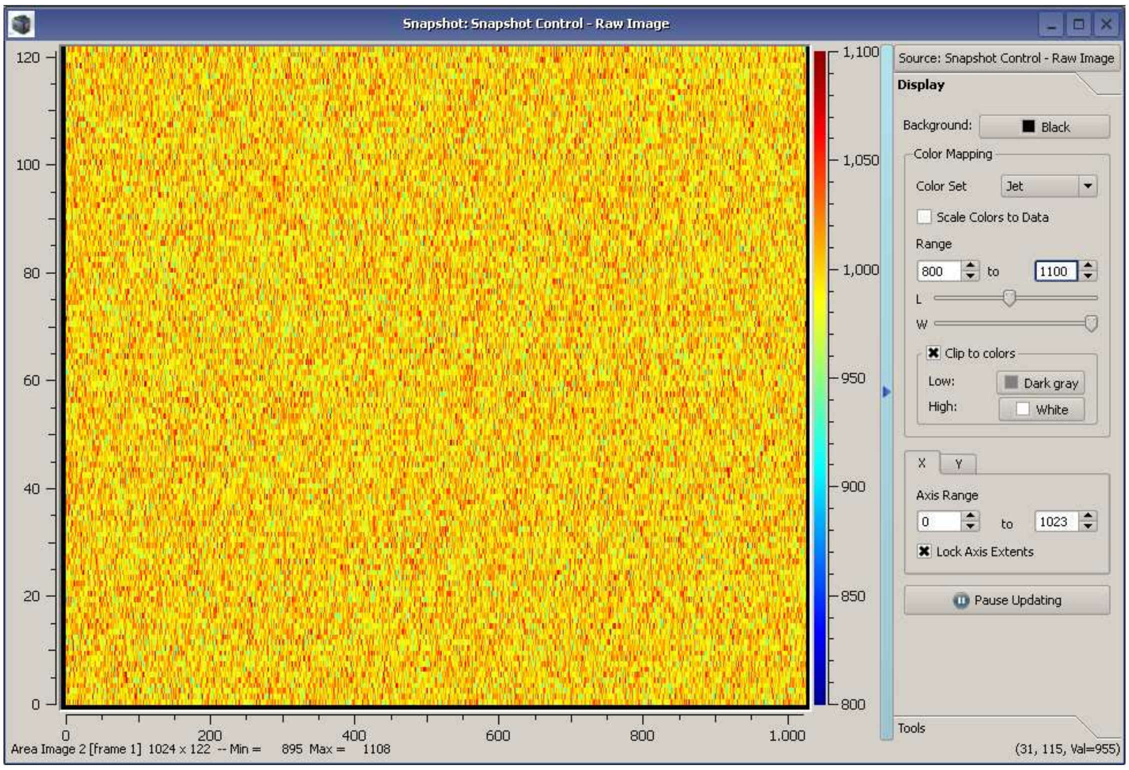


Figure 12 Snapshot Viewer Window

You create new snapshot windows from the **File→New→Snapshot Display** menu option. The blue bar to the right of the image display is used to show and hide the plot settings. At the top of the plot settings area is a menu button that allows you to select the source of the plot data. All plugins that can provide image data are displayed in this menu. Under each plugin, a sub-menu shows the different types of image data the plugin can provide for plotting. The window title will be changed to reflect the current data source and type. The image information (type, size, etc) is displayed in the lower left corner of the window. The position and value of the pixel under the cursor is displayed in the lower right corner of the window.

Display Tab

Just below the data source selection is a toolbox with controls for modifying the data display. You can change the background color and the color mapping of the data. Several color maps are provided and can be chosen using the **Color Set** combo box. If you select the **Scale Colors to Data** checkbox, the colormap will be re-mapped to the data such that the extents of the colormap align with the extents of the data (this obviously uses more CPU resources, so you may want to disable this feature if acquiring images with a high data rate, or large plots). You can also manually set the colormap limits using the **Range** controls (the image min/max values are shown as part of the image description at the bottom-left of the window). Below the **Range** controls are two sliders labeled **L** and **W** (which stands for Level and Window). These controls act like brightness and contrast. The level control adjusts the range center up and down, while the window control adjusts the range in or out. The last set of controls in the **Color Mapping** group is the **Clip to colors** group. When this setting is enabled, pixel values outside of the colormap extents are shown using the specified color. If this group is not enabled, pixel values outside of the colormap range are shown using the last color in the colormap (high or low accordingly).

Below the **Color Mapping** controls are the axis controls. For both the **X** and **Y** axis, you can specify the range of axis. If you deselect the **Lock Axis Extents** (the default is on), the snapshot viewer will scale the axis to the extent of the image every time a new image is loaded. The first time the snapshot viewer window loads an image, the x and y axis ranges are set to the extent of the image (active cols x active rows), unless the user has already set the range manually.

The last control in the **Display** group is the **Pause Updating** button. This is a toggle button and when active, the snapshot display will not update when new images are acquired.

Tools Tab

The **Tools** tab has the **Data Report** button that will create a csv version of the image data and display it in a text viewer. You can then save the text as a file to import into another application.

Plot Viewer Window

The plot viewer window allows you to plot data from any of the plugins that provide it. It can be used to view rows or columns of image data, various image statistics, and sensor data (temperatures, etc).

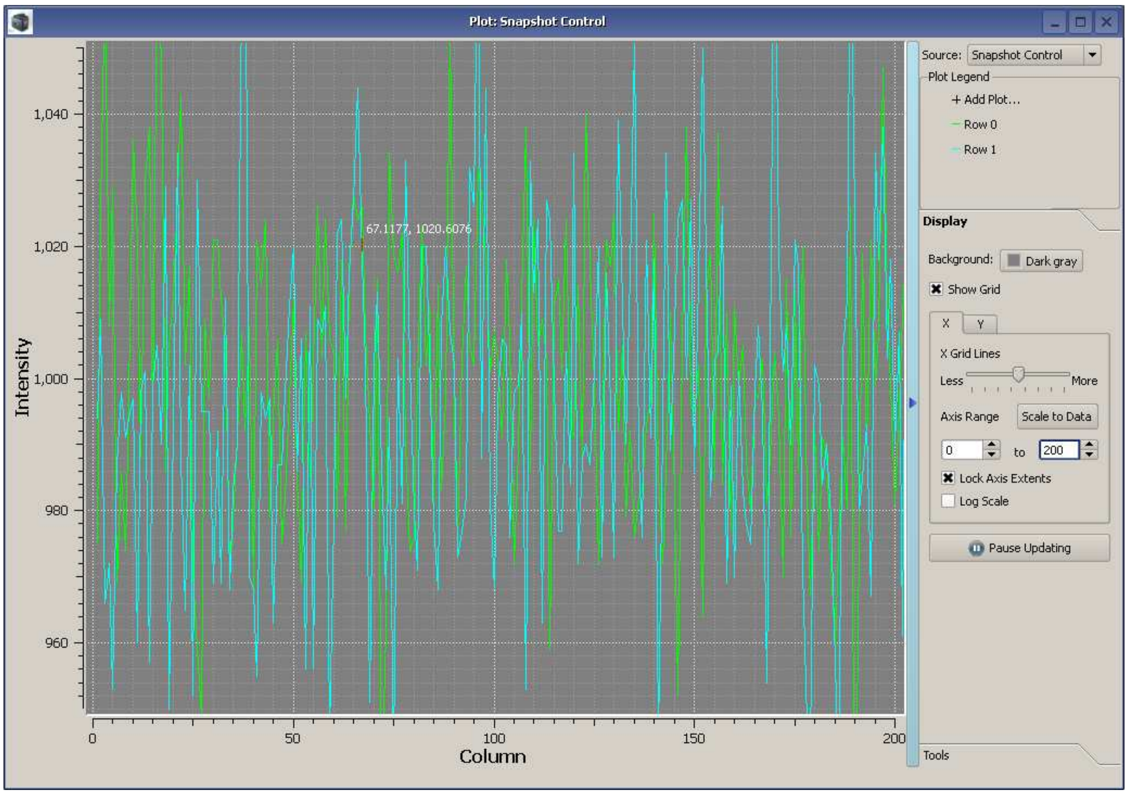
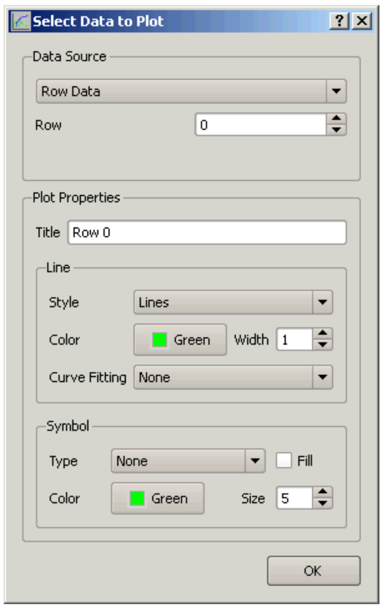


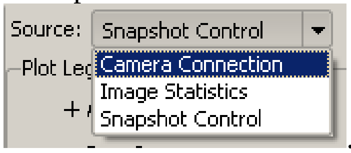
Figure 13 Plot Viewer Window



Row Data
Row: 0
Title: Row 0
Style: Lines
Color: 🟩 Green Width: 1
Curve Fitting: None
Type: None ☐ Fill
Color: 🟩 Green Size: 5

Figure 14 Plot Properties Dialog

Each plot window can be tied to one data source using the **Source** combo box.



.

You then click on + Add Plot... in the **Plot Legend** area to bring up the plot selection dialog. The **Data Source** group varies based on the plugin selected to source the data, but works the same way for all plugins. There is a combo box showing the types of data available from the plugin, and a spin box to select the instance of that data (i.e. row number for image row data). This dialog also has the **Plot Properties** group where you can set the line and symbol styles for the curve. Clicking on the curve in the legend also brings up this dialog to allow you to edit the curve properties. Right clicking on the legend item brings up a menu where you can remove the item from the plot.
Below the **Plot Legend** area is a toolbox for **Display** settings and **Tools**.

Display Tab

The **Display** tab has several controls affecting the data presentation. You can change the background color and the color of the plot, show/hide the grid and control the axis scales. Using the **X** and **Y** axis controls, you can specify the range of axis. If you select the **Lock Axis Extents** (the default is off), the plot viewer will not scale the axis to the extent of the image every time a new image is loaded. Pressing the **Scale to Data** button will perform an auto scaling without affecting the **Lock Axis Extents** setting. The **Log Scale** checkbox changes the given axis scale to a log10 based one.
The last control in the **Display** group is the **Pause Updating** button. This is a toggle button, and when active, the snapshot display will not update when new images are acquired.

Tools Tab

The **Tools** tab has the **Data Report** button that will create a csv version of the image data and display it in a text viewer. You can then save the text as a file to import into another application.
You can control how the mouse interacts with the plot using the Mouse Mode group.

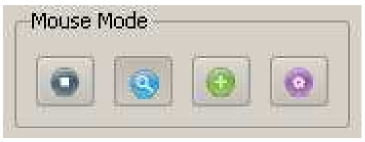


Figure 15 Mouse Mode Controls

■	Disables mouse interaction.
🔍	Uses the mouse to zoom. Left button to drag and define zoom region, right button to zoom back one.
+	Show crosshairs on display when left mouse button is down.
●	Find the nearest point on any curve and show it in the bottom right corner when left button is down.

Dialogs and Dock Widgets

The MityViewer application uses several dock widgets to control various camera attributes. We will go over each of the dialogs individually first. Later in the manual there are some procedures outlined for completing common tasks.

Record/Playback dialog

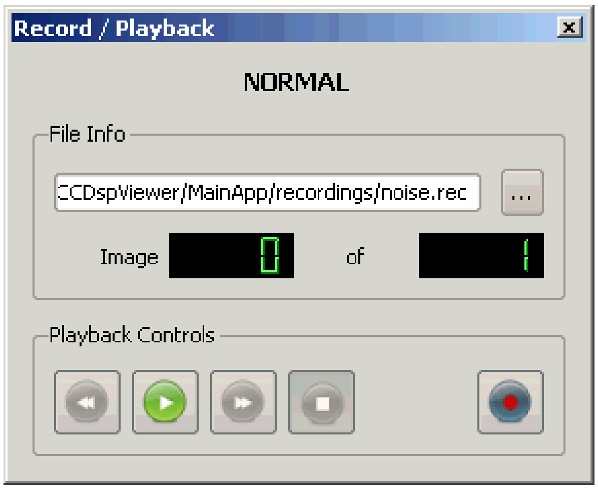


Figure 16 Record/Playback Dialog

The Record/Playback dialog (shown in Figure 16) is used to create and play back record files. To select a record file, click the button and either select an existing file (for playback) or create a new one (for record). The dialog will show how many images are in the record set. To start recording, press the record button (the dialog will prompt you if you will be overwriting an existing record set). Each image acquired is saved to disk before being passed to the rest of the application for processing. Similarly, images played back are injected as if they came from the camera DLL (before any application processing). To stop the recording or playback, push the stop button. Also, during playback, the play button changes to a pause button. The record/playback status is shown at the top of this dialog as well as on the main status bar.

Connect Dialog

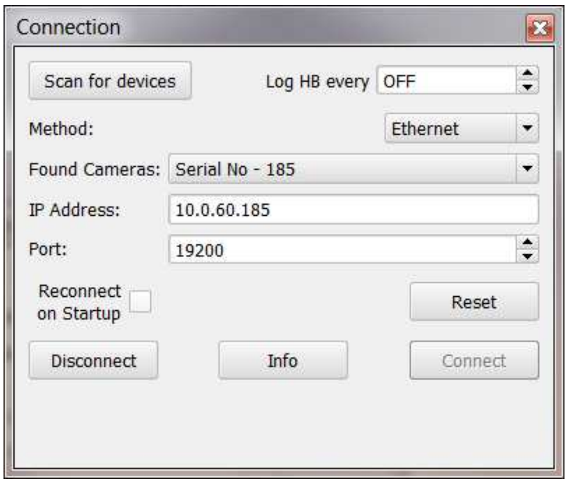


Figure 17 - Connect Dialog

The Connect dialog (*camapp_connect.dll*) is used to specify the camera connection mechanism (USB, Ethernet, etc) and the specific camera you wish to connect to. The Connect dialog is accessed using the "Connect" toolbar item . The MityViewer application polls for both available USB and Ethernet devices [using the `CLLocateHSUSBCameras()` and `CLLocateEnetCameras()` DLL calls]. Pressing the **Disconnect** button closes the camera connection using the `CLCloseCamera()` DLL call. To reconnect to the same camera every time the application is launch, check the **Reconnect on Startup** check box. If the **Log HB every** control is set (not OFF), the application will log the heartbeat message from the camera to a log file based on the camera model and serial number. The files are stored in the *hblogs* directory under the installation directory. This data can be useful for watching trends with camera performance vs temperature, etc. The file can be loaded back in from the Camera Connection plot chooser dialog (if not connected to a camera). The file is stored in a csv format to make it easy to import into other programs.

USB Connection

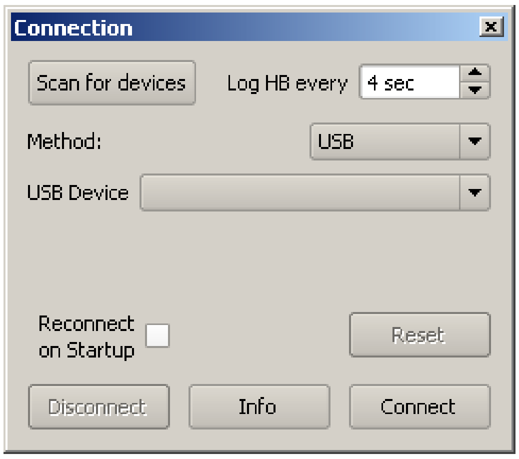
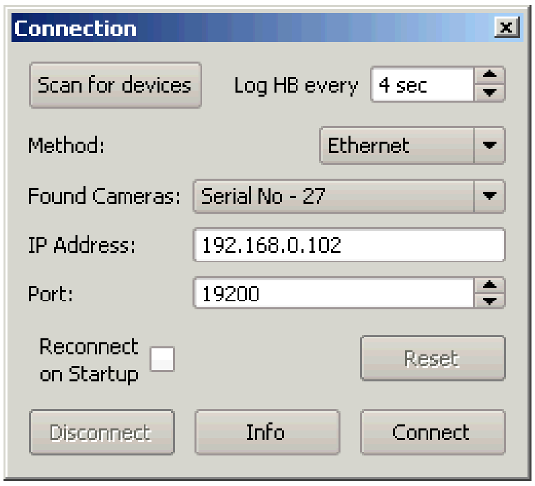


Figure 18 - USB Connection Dialog

Note: *Communication over USB is not supported in this release of the sCMOS development kit.* The USB connection dialog is show in Figure 18. If the camera is connected to the USB port but not communicating (i.e. powered off), it will be shown as a generic MityCCD device like this
0 - CCDSP
. Devices that are functioning will have their model and serial number shown in the **USB Device** combo box. Select the device you want to connect to and press the **Connect** button to initiate a connection to the camera. This connection type utilizes the `CLOpenHSUSBCamera()` DLL call.

Ethernet Connection



Log HB every: 4 sec
Method: Ethernet
Found Cameras: Serial No - 27
IP Address: 192.168.0.102
Port: 19200
Reconnect on Startup: ☐

Figure 19 Ethernet Connection

The Ethernet connection dialog is shown in Figure 19 . Cameras that have been located on the network will be shown in the **Found Cameras** combo box.

Because the camera discovery protocol uses UDP broadcast technology, cameras not on the same local network (subnet) will not be listed. You can still connect to a camera by manually entering the IP address of the desired camera.

Selecting a camera from the Found Cameras combo box will fill out the IP address and Port information on the dialog. The default port is 19200 and should not normally be changed. Select the device you want to connect to and press the **Connect** button to initiate a connection to the camera. This connection type utilizes the `CLOpenENetCamera()` DLL call.

Serial Number Connection

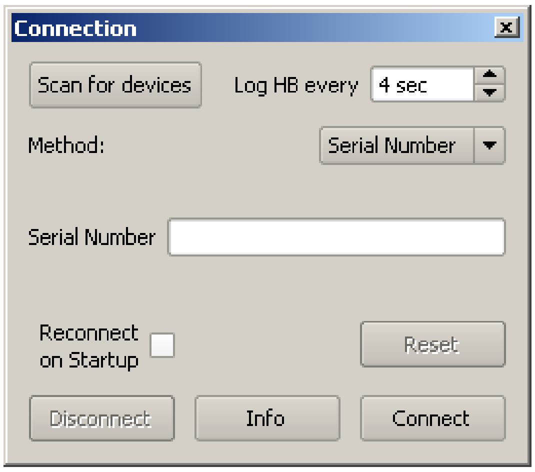


Figure 20 Serial Number Connection

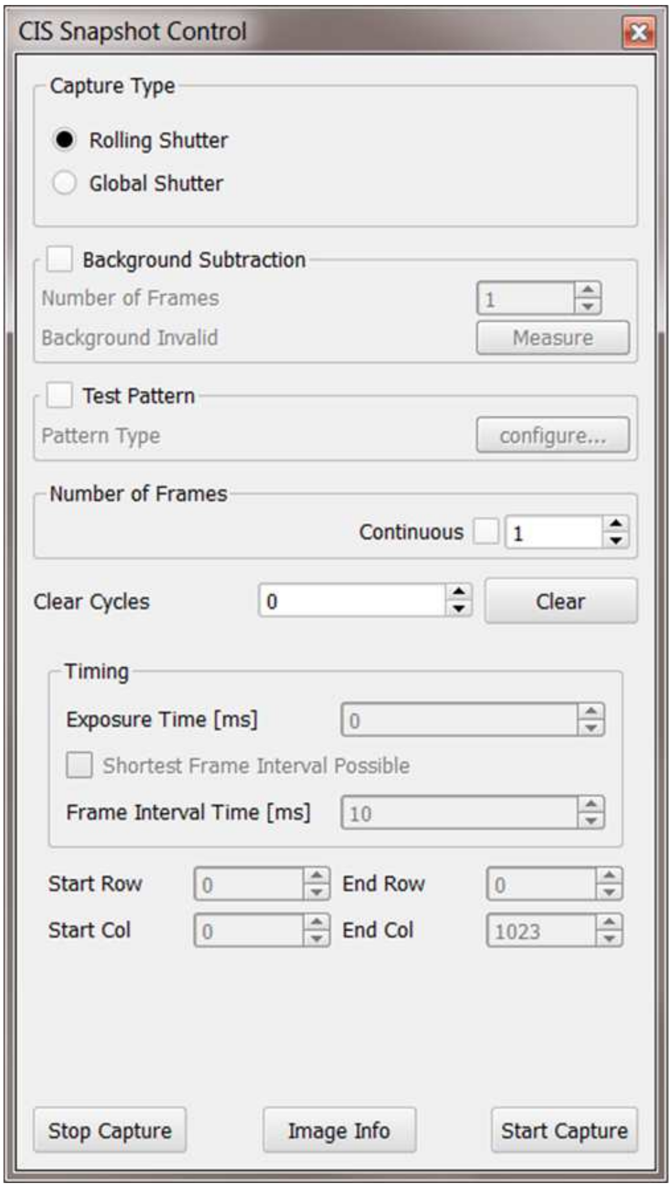
To connect to a camera based on its serial number, select “Serial Number” from the Method combo box. This will show a text field allowing you to enter the serial number of the camera you wish to connect to. When you click the Connect button, the application will use the `CLOpenCameraBySerialNumber()` DLL call to try and establish a connection to the camera. This call tries all available high-speed USB cameras [using the `CLLocateHSUSBCameras()` DLL call] and then polls for Ethernet based cameras [using the `CLLocateEnetCameras()` DLL call] to try and find a camera whose serial number matches the specified value. If the correct camera is found the corresponding DLL call [`CLOpenHSUSBCamera()` or `CLOpenENetCamera()`] is used to open it.

Snapshot Control Dialog

The Snapshot Control dialog (*camapp_snapshotcontrol.dll*) allows you to configure the image acquisition parameters of the camera. There are many sections of this dialog, and they all interact to control the image. Please refer to Figure 21**Error! Reference source not found.** which shows the Snapshot control dialog.

Capture Type

The capture type selects between rolling and global shutter mode. Rolling shutter is the default method of reading out the sensor and consists of horizontally reading out the sensor one row at a time. The image is captured over time; therefore, some blurring may occur for moving objects. In global shutter mode, the image is captured once and then scanned out. Please see application note titled “PM0018_MityCAM-CIS1210_CIS1910 Exposure and Interval Timing” for a discussion on acquisition timings for rolling and global shutter.


CIS Snapshot Control

Capture Type
- ● Rolling Shutter
- ○ Global Shutter

☐ Background Subtraction
- Number of Frames: 1
- Background Invalid — Measure

☐ Test Pattern
- Pattern Type — configure...

Number of Frames
- Continuous ☐ 1

Clear Cycles: 0 — Clear

Timing
- Exposure Time [ms]: 0
- ☐ Shortest Frame Interval Possible
- Frame Interval Time [ms]: 10

Start Row: 0 End Row: 0
Start Col: 0 End Col: 1023

Stop Capture | Image Info | Start Capture

Figure 21 CIS Snapshot Control Dialog

Background Subtraction

To enable background subtraction processing, check the box ☐ Background Subtraction for this group. You can then enter the number of frames to use for this processing and click **Measure** to build the background image. This will first clear the CCD by calling `CLClearCCD()`, and then call either `CLReadCCDArea()` or `CLReadCCDBinned()` depending on the type of image selected to start acquiring frames for the background estimate. The frames collected during the measurement phase are averaged to form the background estimate. This image is then subtracted from acquired images while this feature is enabled. The application checks for the image type (area, bin, etc) and geometry to only operate on like images. It does not check the exposure timing. If you change the exposure parameters, you should re-run the measurement process.

Test Pattern

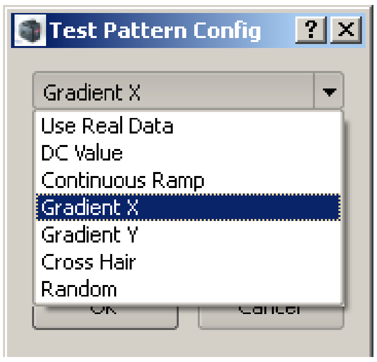
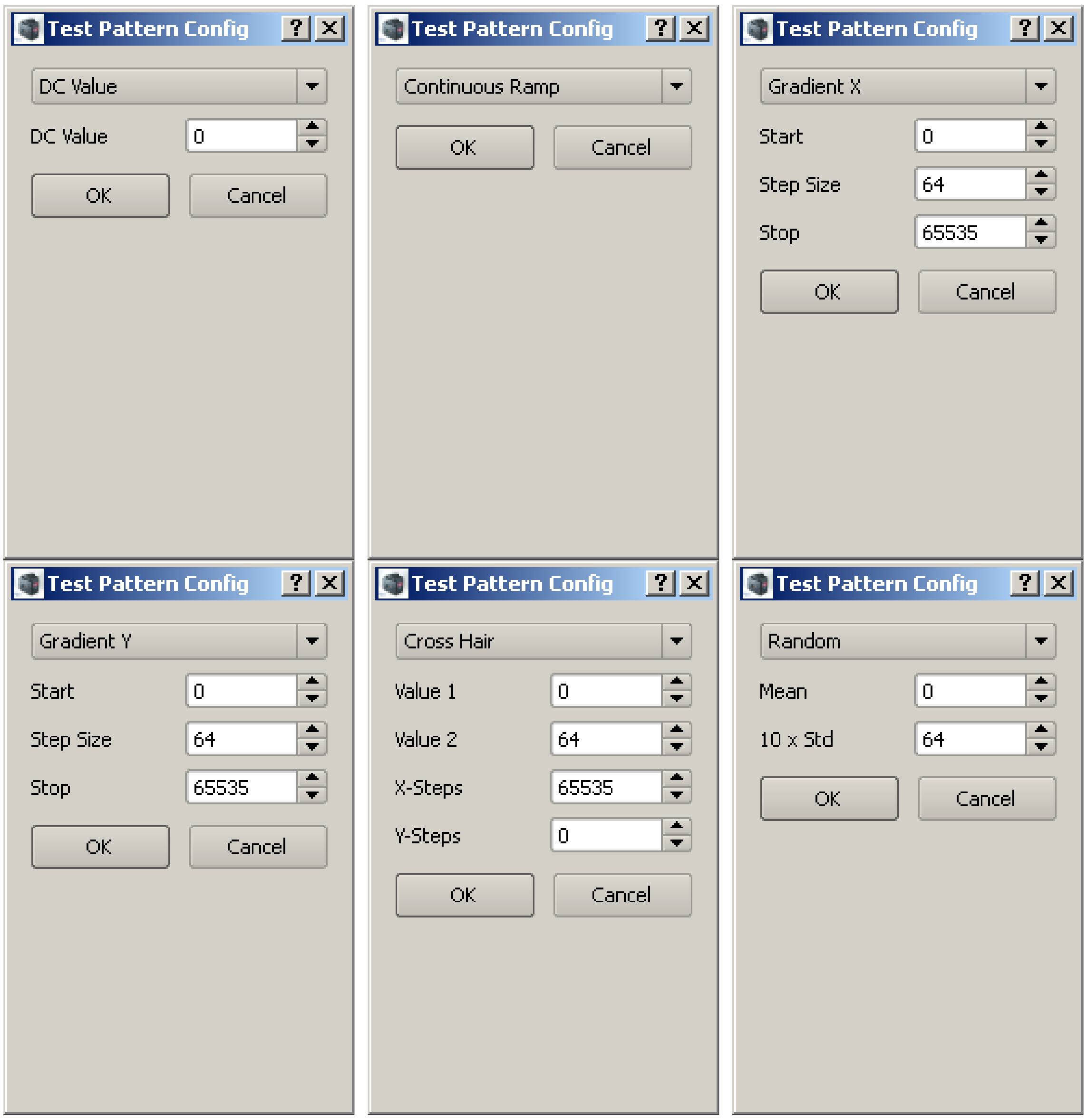


Figure 22 Test Patterns

Note: *Test Pattern function is not supported in this release of the sCMOS development kit.* To substitute a test pattern for the acquired image [using `CLSetTestPattern()`] check this box ☒ Test Pattern. The Configure button will become enabled and will allow you to change the type and parameters of the test pattern.
The different types of test patterns available are shown in the figure to the left. Each type of test pattern is described below.

Table 6 Test Pattern Types

Type from clcamiface.h	Description
`PATTERNTYPE_NONE`	Use real CCD data
`PATTERNTYPE_DC`	Each read pixel from camera is replaced with DC value in mnSimParams[0]
`PATTERNTYPE_RAMP`	Pixels are incremented by one starting at zero
`PATTERNTYPE_GRADIENT_X`	Pixels are scaled from PatternParams[0] to PatternParams[1] by PatternParams[2] in X dimension
`PATTERNTYPE_GRADIENT_Y`	Pixels are scaled from PatternParams[0] to PatternParams[1] by PatternParams[2] in Y dimension
`PATTERNTYPE_CROSSHAIR`	Two values, cross hairs DC value in PatternParams[0], value 2 in PatternParams[1], every [2] in X, [3] in Y
`PATTERNTYPE_RANDOM`	Pixels mean is PatternParams[0] with std of 0.1 * PatternParams[1]



Test Pattern Config: DC Value
DC Value: 0

Test Pattern Config: Continuous Ramp

Test Pattern Config: Gradient X
Start: 0
Step Size: 64
Stop: 65535

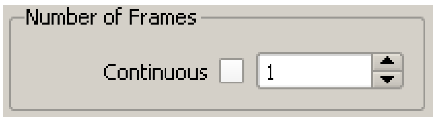
Test Pattern Config: Gradient Y
Start: 0
Step Size: 64
Stop: 65535

Test Pattern Config: Cross Hair
Value 1: 0
Value 2: 64
X-Steps: 65535
Y-Steps: 0

Test Pattern Config: Random
Mean: 0
10 x Std: 64

Figure 23 Test Pattern Configuration

Number of Frames



Number of Frames
Continuous ☐ 1

Figure 24 Number of Frames

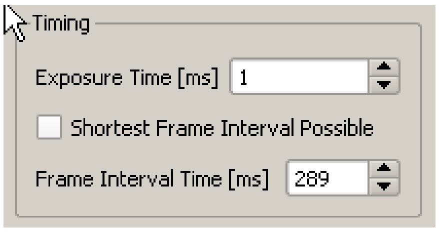
The **Number of Frames** control determines whether the system runs in single-shot or continuous mode. It also controls whether the image data is saved into internal memory or streamed to the PC. The system supports two different operational modes: In Mode 1 the system may acquire a single or continuous image data and send it to the PC for display and possible recording. It can also stream a limited number of frames (up to 35 frames for CIS1011 full ROI) into internal on-board memory (Mode 2) and then transfer data to the PC for display and possible recording. The benefit of the Mode 2 is that the image data may be acquired at a much faster rate as there are no PC transfer delays to deal with. However, Mode 2 is limited to 35 frames or less of image data. As soon as the data is acquired, the system will transfer image data from the system to the PC (without any further user action).

- Acquire a single frame (Mode 1): Uncheck the **Continuous** check box, and set the **Number of Frames** to '1'. Once you select **Start Capture,** then the camera will acquire a single frame and transmit it to the PC.
- Acquire continuous images and stream to PC (Mode 1): Check the **Continuous** check box, and set the **Number of Frames** to '1'. Once you select **Start Capture** the system will continuously acquire image data until the **Stop Capture** button is selected. Each frame is sent to the PC and displayed. Please see the section below titled Timing Control for a description of acquisition timings.
- Acquire continuous images and stream to on-board memory (Mode 2): Uncheck the **Continuous** check box, and set the **Number of Frames** to a number greater than '1'. The system will acquire the specified number of frames and store them into on-board memory. Once acquisition starts, the system will transfer each frame to the PC at a slower rate. This feature is useful for acquiring image data at a fast acquisition rate as it overcomes the relatively slow PC link speed. Note the maximum number of frames is limited to 35 frames (CIS1210 full ROI).

Capture Controls

The clear CCD function is not supported on the sCMOS system.

Timing Control



Exposure Time [ms] 1

☐ Shortest Frame Interval Possible

Frame Interval Time [ms] 289

Figure 25 Timing Controls

Figure 33 below shows the timing diagram and relationship between Exposure Time, Frame Time and Frame Interval Time. The Timing controls allow you to adjust the timing parameters of the capture. Either reading data into on-board RAM or streaming data to the PC can be configured with a precise exposure time between successive image reads or image cycles. The **Exposure Time [ms]** control sets the exposure time using the `CLSetExposure()` API command. The **Frame Interval Time [ms]** control uses the

`CLSetImageInterval()` API call to set the frame interval. The **Shortest Fame Interval Possible** control is not supported on the sCMOS development kit.

The **Exposure Time** is in milliseconds and in rolling shutter mode it can range from a minimum of 1/37.5 MHz to up to 9 seconds for full ROI. The maximum exposure time will depend on the selected ROI (vertical direction). The **Frame Interval Time** specifies the time required to accommodate back to back frames. When streaming data to the PC (Mode 1), this time should be set to 1250 ms or longer to accommodate link delays (to the PC). Setting the **Frame Interval Time** to a time shorter than 1250 ms will cause the image data to be buffered in on-board memory and eventually the system will stall due to memory not being available.

When operating in Mode (2), you may set the **Frame Time Interval** to 30 ms (or larger).

Note, the time parameter specified in the **Exposure Time** should fall within the **Frame Interval Time** parameter. Please see the application note titled Understanding sCMOS Exposure and Interval Timing for details on exposure timings.

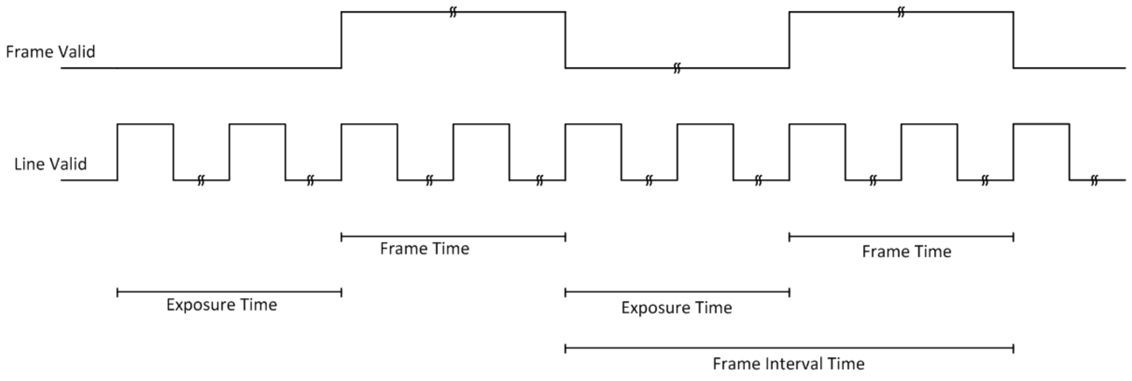
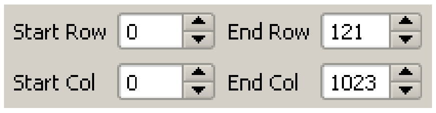


Figure 26 - sCMOS Exposure Timings

Area Control



Start Row 0 End Row 121
Start Col 0 End Col 1023

Figure 27 Area Controls

The Area Control lets the user set the ROI of the acquired image. The values set with these controls are passed as arguments to the `CLSetCCDArea()` API call. There are currently limitations as to the ROI which can be specified. Typically, one operates in full ROI (Start Row=0, End Row=1023; Start Col=0, End Col 1279 for CIS1210 sensor).

One may also operate in ¼ image mode (Start Row=0, End Row=512; Start Col=0, End Col 512). In addition, start / end row may be varied as long as the full extents of the columns are selected.

Calibration Dialog

The Calibration dialog is shown using its icon on the toolbar. The Calibration dialog is used to set up base parameters for the camera such as output channel selection, pixel calibration and temperature setpoint.

Channel Configuration

The sCMOS development kit supports several pixel read-out modes which are described in this section and summarized in the table blow. The sCMOS sensor reads out each pixel in both **High Gain** (x30) and **Low Gain** (x1) form. The sCMOS development kit acquisition electronics (MityDSP FPGA) processes both high and low gain channel data and can transmit either the data as high gain only and low gain only output data. It can also combine pixels form both high and low gain channels into a single pixel (Combined mode). In Combined mode, pixel combining is applied on a pixel by pixel basis – please see application note titled Pixel Combining Design Document for more details.

Channel select drop down list lets the user select between **Low Gain**, **High Gain**, and **Combined Mode**.

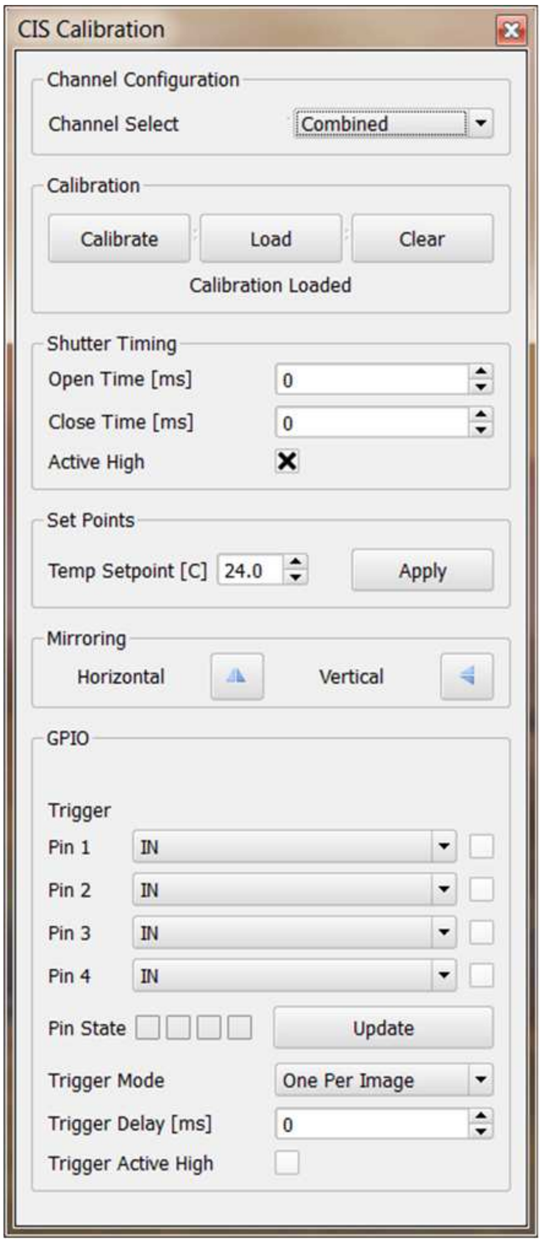
Calibration

The pixel calibration menu allows the user to display the 'raw' pixel value or apply an offset correction value to each pixel.

Please see the application note titled "MityCAM-CIS1210/CIS1910 Pixel Combining Pixel Design Document" for a discussion on offset correction applied. The following buttons manage the applying offset calibration

- **Calibrate:** Saves calibration data into non-volatile memory. Cover the sensor / optics with a dark cloth to before selecting **Calibration** to ensure valid calibration data.
- **Load:** Loads the calibration data from non-volatile memory and applies the correction data to each pixel. Note that when correction is selected, it's applied to the selected output channel (Low Gain, High Gain, or Combined Mode)
- **Clear:** Clears the offset correction data from memory and any subsequent image acquisitions are transmitted as 'raw' pixel data without any correction applied.

Read-out Mode	Raw Data	Offset Correction	Pixel Correction
Low Gain	The output data in this mode is shown without correction	Offset correction is applied	Pixel correction is not currently implemented
High Gain			
Combined			


CIS Calibration

Channel Configuration
Channel Select: Combined

Calibration
Calibrate | Load | Clear
Calibration Loaded

Shutter Timing
Open Time [ms]: 0
Close Time [ms]: 0
Active High: ☒

Set Points
Temp Setpoint [C]: 24.0
Apply

Mirroring
Horizontal | Vertical

GPIO
Trigger
Pin 1: IN
Pin 2: IN
Pin 3: IN
Pin 4: IN
Pin State ☐☐☐☐ Update
Trigger Mode: One Per Image
Trigger Delay [ms]: 0
Trigger Active High: ☐

Shutter Timing

Note: ***Shutter Output*** *function is not supported in this release of the sCMOS development kit.* The **Shutter Timing** group is uses the CLSetShutterParams() API call to set the shutter parameters. The camera will open the shutter (TTL pulse '1' when *ActiveHigh* is non-zero, '0' otherwise). The camera will delay for *OpenMs* milliseconds, expose for the exposure time from `CLSetExposure()`. The camera will close the shutter (TTL pulse '0' when ActiveHigh is non-zero, '1' otherwise), delay CloseMs milliseconds, then the CCD pixel data will be read/shifted from the CCD array.

Set Points

The sCMOS development kit comes with a 1 stage Thermo Electric Cooler (TEC) device and temperature sensor installed on the sensor. Both the temperature sensor and TEC device may be used to maintain the sensor at a set temperature point. The **Temperature Set Point** may be set to within .1deg C and selecting the Apply button downloads the setpoint to the camera. Turn on cooling by selecting the radio button labeled Cooling on the main window status bar

WARNING: The sCMOS sensor and thermoelectric cooler (TEC) are **not** housed in a vacuum chamber and therefore great care must be taken when operating the TEC as condensation may occur. Critical Link recommends maintaining the sensor at 5 deg. C below the sensor ambient temperature. Contact Critical Link if you notice condensation on the senor.

Mirroring

Note: *The **Mirroring** function is not supported in this release of the sCMOS development kit.* This function is used to configure image mirroring for the sCMOS sensor. It is useful for presenting images to an operator in reverse readout order so that an image may be "flipped" along the horizontal or vertical axis. This allows readouts to be more intuitive to a user (e.g., spectrum data aligned from low frequency to high rather than high to low based on optics setup, etc.). The **Mirroring** group allows you to set the horizontal and vertical mirroring (using the `CLSetCCDImageMirroring()` API call). See the API documentation for specifics regarding how mirroring is accomplished in the camera.

GPIO

Note: *The **GPIO** function is not supported in this release of the sCMOS development kit.* The **GPIO** group allows you to configure the 4 GPIO pins available on the sCMOS development kit using the `CLSetGPIO()` API call. Each pin can be configured as an input or output. If a pin is configured as an input, the **Trigger** checkbox is used to set a specific pin as the trigger via the `CLSetTrigger()` API call. The different states you can assign to the GPIO pins are listed in **Error! Reference source not found.**.

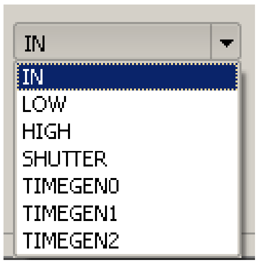


Figure 28 GPIO Pin configuration

Table 7 GPIO States

State	Description
eeINPUT	set the GPIO as in INPUT
eeOUTLOW	drive GPIO low
eeOUTHIGH	drive GPIO high
eeSHUTTER	drive GPIO with shutter signal
eeOUTTIMGEN0	drive GPIO with custom timing strobe 0
eeOUTTIMGEN1	drive GPIO with custom timing strobe 1
eeOUTTIMGEN2	drive GPIO with custom timing strobe 2

Image Statistics Dialog

The Image Noise Statistics dialog is shown using its icon on the toolbar. When the **Enable** box is checked, this dialog will compute image statistics on every acquired image. When acquiring images at high frame rates, you may need to disable this dialog to achieve the desired throughput (depending on your PC's CPU). The dialog computes the image mean pixel value, the noise (std deviation) value, and shows how many pixels fall outside of 1,2,3, and 4 sigma from the noise value as a percentage. The nominal values for these percentages (based on a Gaussian distribution) are shown to the right of the values). You can reduce the size of the considered area by adjusting the **X Margin** and **Y Margin** controls. These values are applied to both edges of the X or Y limits. The noise computation is a multi-pass operation. First it does a sliding window average (using the **Window** setting for the size of the window) across a row. Any pixel who's value is more than **Peak Cutoff** counts above the average is discarded for the computation and the average of its one-away neighbors is used for the computation (the actual pixel data is untouched). Once this peak shearing operation is done, the line mean is computed and used to derive the noise figure. The running average noise and mean values are displayed next to the image mean and noise fields. Pressing the **Reset Averages** button clears the running average.

Plot Statistics

The Image Noise Statistics dialog can compute image statistics suitable for plotting using the plot display window. Clicking the **Generate** button will compute the following data sets:

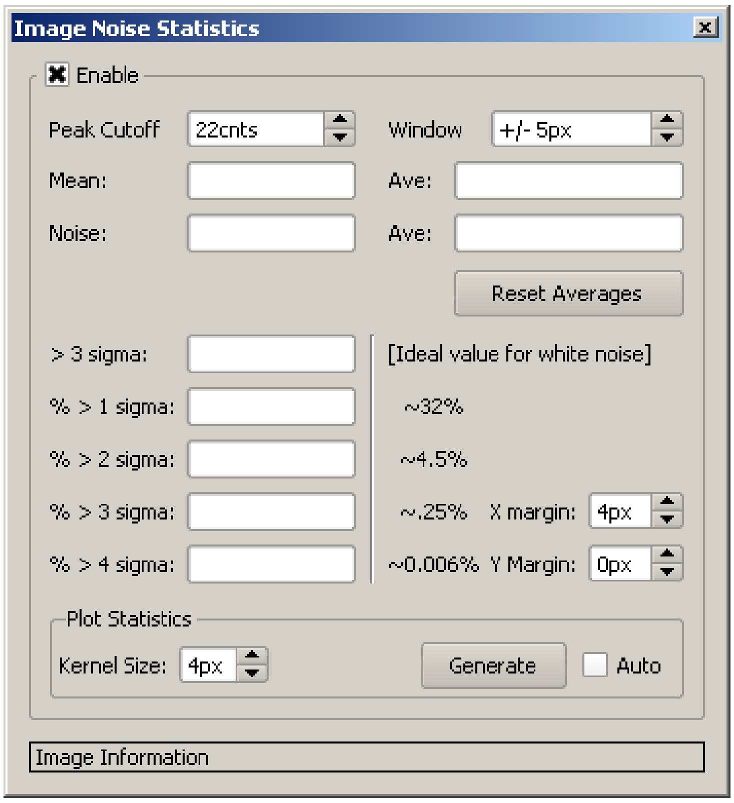


Image Noise Statistics

☒ Enable

Peak Cutoff: 22cnts — Window: +/- 5px

Mean: — Ave:

Noise: — Ave:

Reset Averages

> 3 sigma: — [Ideal value for white noise]

% > 1 sigma: — ~32%

% > 2 sigma: — ~4.5%

% > 3 sigma: — ~.25% — X margin: 4px

% > 4 sigma: — ~0.006% — Y Margin: 0px

Plot Statistics

Kernel Size: 4px — Generate — ☐ Auto

Image Information

Figure 29 Image Noise Statistics Dialog

• StdDev by Row	StdDev for each pixel in a row
• StdDev by Column	StdDev for each pixel in a column
• Mean by Row	Mean value across a row
• Mean by Column	Mean value along a column
• Data by Row	Image data in a given row
• Data by Column	Image data in a given column
• FFT by Row	FFT [1024 bin] of the data in a row
• Histogram	Image histogram.

Checking the **Auto** box will cause the dialog to compute image plot statistics for every image acquired. You may find this is too much for your CPU to handle.

Peak Finder Dialog

The Peak Finder dialog can be used to examine a row in an image to find the peak values.

Click the icon in the toolbar to open this dialog.

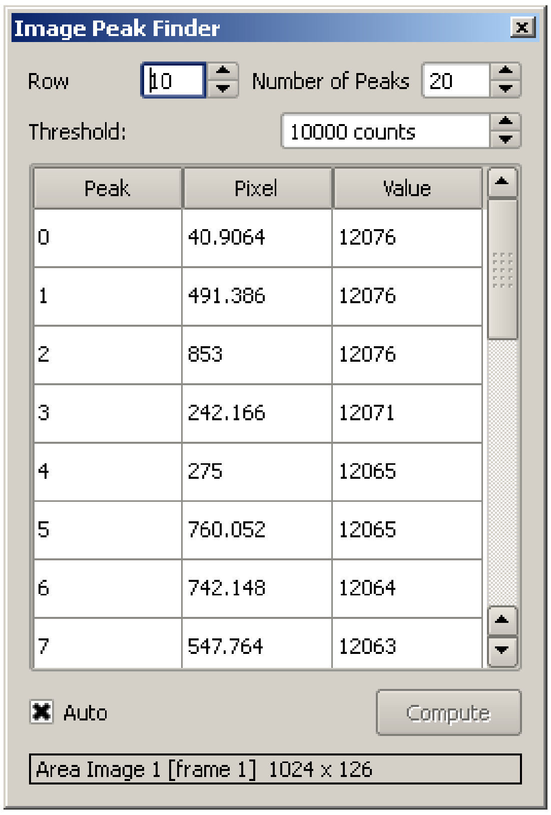


Image Peak Finder

Row 10 Number of Peaks 20

Threshold: 10000 counts

Peak	Pixel	Value
0	40.9064	12076
1	491.386	12076
2	853	12076
3	242.166	12071
4	275	12065
5	760.052	12065
6	742.148	12064
7	547.764	12063

☒ Auto Compute

Area Image 1 [frame 1] 1024 x 126

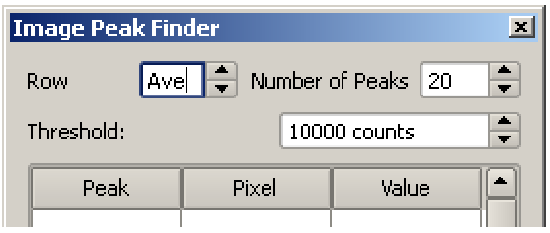


Image Peak Finder

Row Ave Number of Peaks 20

Threshold: 10000 counts

Peak	Pixel	Value

Figure 30 Peak Finder Dialog

Use the **Row** control to select which row to find peaks on (or select "Ave" to average all the rows in the image [its one down from "0"]). The Number of Peaks control limits the number of peaks extracted. Only pixels whose value exceeds the threshold are considered in the peak processing. The Pixel value shown is interpolated using a centroid algorithm and thus is a floating point pixel equivalent. The Value shown is the pixel value of the peak from the image (not interpolated).

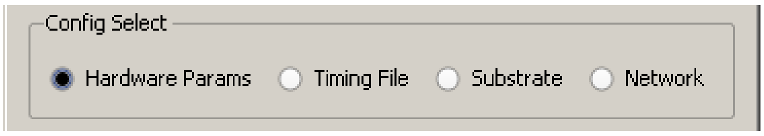
If the **Auto** button is checked the peaks are located for every image acquired, otherwise click the **Compute** button to populate the table.

Link Test Dialog

The link test dialog is used to verify link integrity and measure I/O bandwidth. This dialog is for **factory use only**.

Factory Configuration Dialog

The Factory Configuration dialog is used to do the basic setup of the camera and is intended for **factory use only**. It is used to set up the basic model and serial number information, timing file, substrate configuration, and network settings. The vast majority of this data should not be changed by anyone outside of the Critical Link engineering and/or production departments. There are 4 views on this dialog, selectable from the radio buttons at the top.

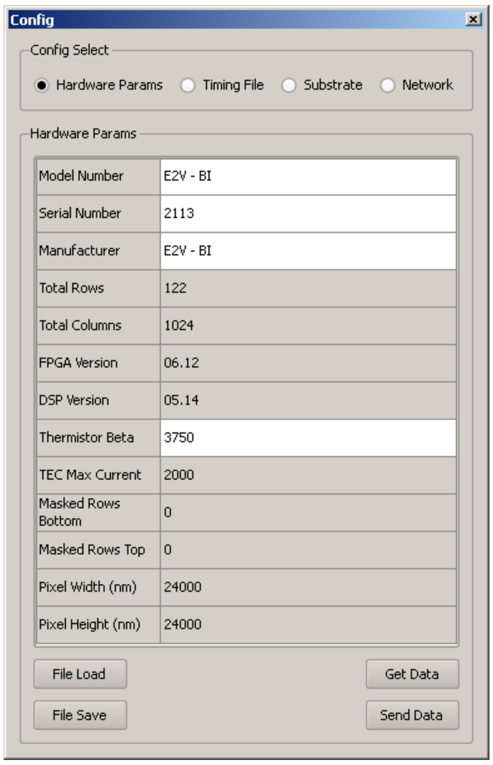


Config Select

[✓ Hardware Params] Timing File Substrate Network

Figure 31 Factory Config Selection

Hardware Params



Config

Config Select

[✓ Hardware Params] Timing File Substrate Network

Hardware Params

Model Number	E2V - BI
Serial Number	2113
Manufacturer	E2V - BI
Total Rows	122
Total Columns	1024
FPGA Version	06.12
DSP Version	05.14
Thermistor Beta	3750
TEC Max Current	2000
Masked Rows Bottom	0
Masked Rows Top	0
Pixel Width (nm)	24000
Pixel Height (nm)	24000

Figure 32 Hardware Parameters Dialog

The Hardware Params view shows (and lets you edit some of) the parameters shown in Table 8. You can save thee current parameters off to a file using the File Save button and subsequently load them back from the file with the File Load button. The parameters are retrieved from the camera automatically, but you can retrieve them manually using the Get Data button. This will make a call to the CLGetCameraHardwareInfo() API call. To update the parameters on the camera, press the Send Data button (this uses internal engineering commands to send the configuration to the camera). End users should normally have to use this feature.

Table 8Hardware Parameters

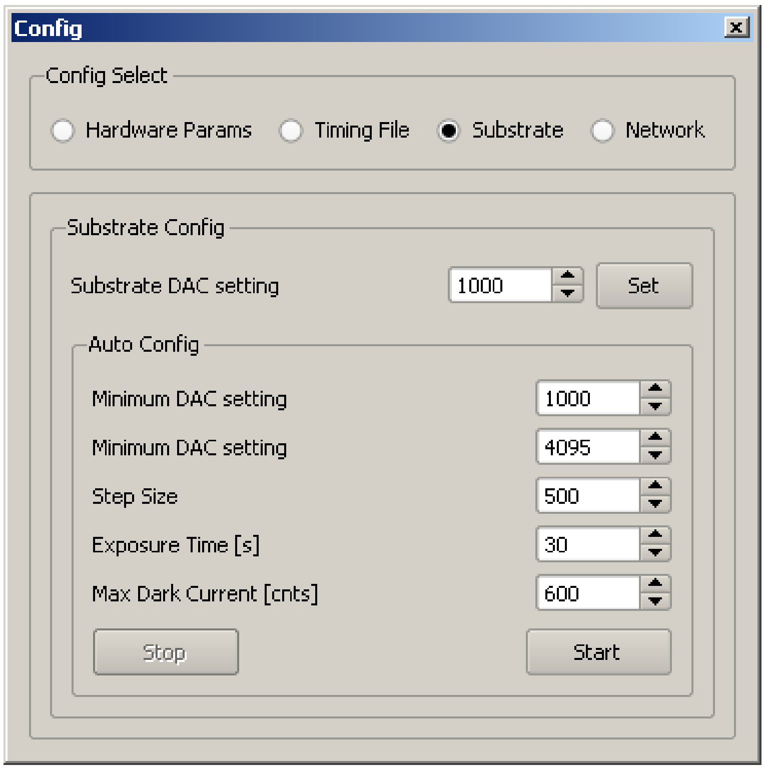
Parameter	**R/W**	**Description**
Model Number	**Y**	Camera model number (as determined by MFG).
Serial Number	**Y**	Camera serial number.
Manufacturer	**Y**	Camera Manufacturer.
Total Rows	N	Total number of rows in the sensor.
Total Columns	N	Total number of columns in the sensor.
FPGA Version	N	FPGA version number.
DSP Version	N	DSP software version number.
Thermistor Beta	**Y**	CCD temperature thermistor beta factor.
TEC Max Current	N	TE cooler current limit.
Masked Rows Bottom	N	Number of rows masked (and discarded) at bottom of sensor.
Masked Rows Top	N	Number of rows masked (and discarded) at top of sensor.
Pixel Width (nm)	N	Pixel width in nanometers.
Pixel Height (nm)	N	Pixel height in nanometers.

Timing File

The Timing File view shows the contents of the timing file for the camera. This file controls the low-level timing of reading the sensor, as well as defining the basic attributes of the sensor. Timing files are generated by Critical Link engineering and are not meant to be modified by end users.

Substrate

The Substrate configuration view is used to configure the substrate DAC setting. You can enter the value manually, or have the application run a calibration loop acquiring dark current images, stepping the DAC setting until **the Max Dark Current** value is met. This loop uses the `CLSetExposure()` and `CLReadCCDArea()` API calls. The actual setting of the DAC value is done via internal engineering functions.



Substrate Config

Substrate DAC setting	1000

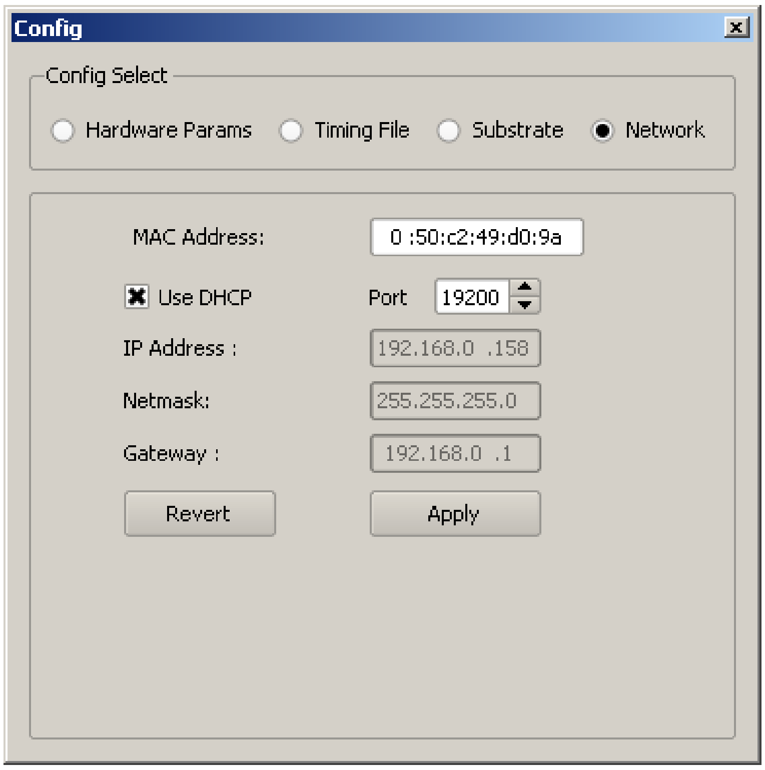
Auto Config

Minimum DAC setting	1000
Minimum DAC setting	4095
Step Size	500
Exposure Time [s]	30
Max Dark Current [cnts]	600

Figure 33 Substrate Configuration Dialog

Network

The **Network** view allows you to reconfigure the MityCAM camera's network configuration. You can select the Use DHCP box to enable DHCP network settings, or enter the configuration manually. The Apply button uses the `CLSetCameraNetworkConfig()` to send the settings to the camera. The **Revert** button will revert the dialog settings to those currently in the camera.



Hardware Params Timing File Substrate [✓ Network]

MAC Address: 0 :50:c2:49:d0:9a

[✓ Use DHCP] Port 19200

IP Address : 192.168.0 .158

Netmask: 255.255.255.0

Gateway : 192.168.0 .1