



Using an AVT camera with Common Vision Blox

VC AVT1394, Delphi CVAVT1394 and VB CVAVT1394 examples for cvAVT1394.vin

Common Vision Blox (CVB) and the example applications delivered on the CVB CD enable you to test your AVT cameras easily. In addition it offers the opportunity to control standard and advanced features in your own application, because the whole source codes are also delivered.

The source code of the Visual C++ application is available as an MS Visual Studio 6 project, the source code of the Delphi application as a Borland Delphi 7 project and the VB source code as an MS Visual Basic 6 project.

Valid for the cvAVT1394.vin driver from CVB 9.0.1 on.

This document can NOT replace the camera manual!



Table of contents

1. VCAVT1394 example	3
1.1 First start of the VC 6 Example Application.....	3
1.2 Main Window	4
1.3 AVT Communication Dialog	5
1.3.1 AVT Basic Communications	5
1.3.1.1 Direct Register Access.....	6
1.3.2 Shading.....	6
1.3.3 LUT	7
1.3.4 ROI Settings	8
1.3.5 IO Controls.....	9
1.3.6 Test Image.....	10
2. VB CVAVT1394 Example	11
2.1 First start of the Visual Basic Example Application	11
2.2 Graphical User Interface	12
2.2.1 Direct Register Access	13
3. Delphi CVAVT1394 Example.....	14
3.1 First start of the Delphi Example Application	14
3.2 Graphical User Interface	15
2.2.1 Direct Register Access	16



► Using an AVT camera with Common Vision Blox



1. VCAVT1394 example

1.1 First start of the VC 6 Example Application

After installing Common Vision Blox (CVB) and the IEEE AVT 1394(CVAVT1394.vin) driver on your system according to the CVB Hardware Quickstart 1394 AVT, you can find the application in:

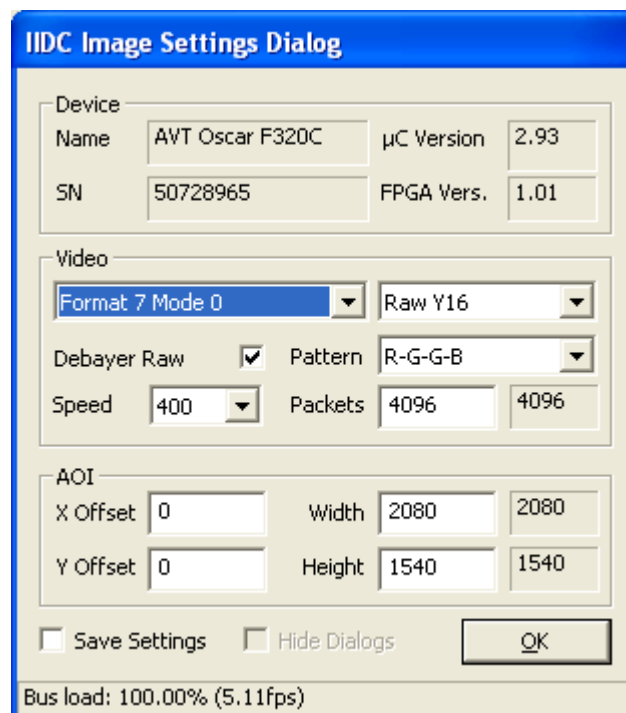
Start → Programs → Common Vision Blox → Hardware → AVT → 1394 → VC AVT1394 Example

If you are interested in the source code you can find it in

Tutorial\Hardware\AVT1394\VC\VCAVT1394

based on the CVB directory (default is C:\Program Files\Common Vision Blox).

As described in the QuickStart a configuration window will appear:



The cvAVT1394.vin supports only format 7 modes.

Format 7 is a scalable format without a fixed resolution, frame-rate and colour mode. The available modes are defined by the camera vendor and can be found in the camera manual. If you are using a colour camera with raw data output please check Debayer Raw and select a pattern. By default CVB uses the whole sensor-resolution and the maximum speed & packet size.

After applying the desired settings the main window of the application is shown:



► Using an AVT camera with Common Vision Blox

1.2 Main Window



Shown above is the main window of the *VC AVT 1394 Example*. On the right-hand side you can see the CVB-Display containing the image and a status bar. The status bar displays information on the current image coordinates the mouse hovers over (no screen coordinates) and the colour information of that pixel. Also when images are grabbed the Frames Per Second (fps) and the Image size are displayed.

The window and the display are resizable so the displayed image can be adjusted according to the screen resolution. Additionally by right clicking you can zoom in (long click) or out (short click).

On the left-hand side are a number of controls:

Control	Description
<i>Save Image</i>	Saves the currently displayed image to disk
<i>Copy to Clipboard</i>	Copies the currently displayed image to clipboard
<i>Grab</i>	Starts/stops image acquisition
<i>Snap</i>	Acquires single image
<i>Board Select</i>	Switches to another board or bus
<i>Camera Select</i>	Switches to another camera
<i>Trigger Mode</i>	Enables/disables asynchronous trigger acquisition
<i>AVT Communication Dialog</i>	Opens the AVT Advanced Features dialog (described later)



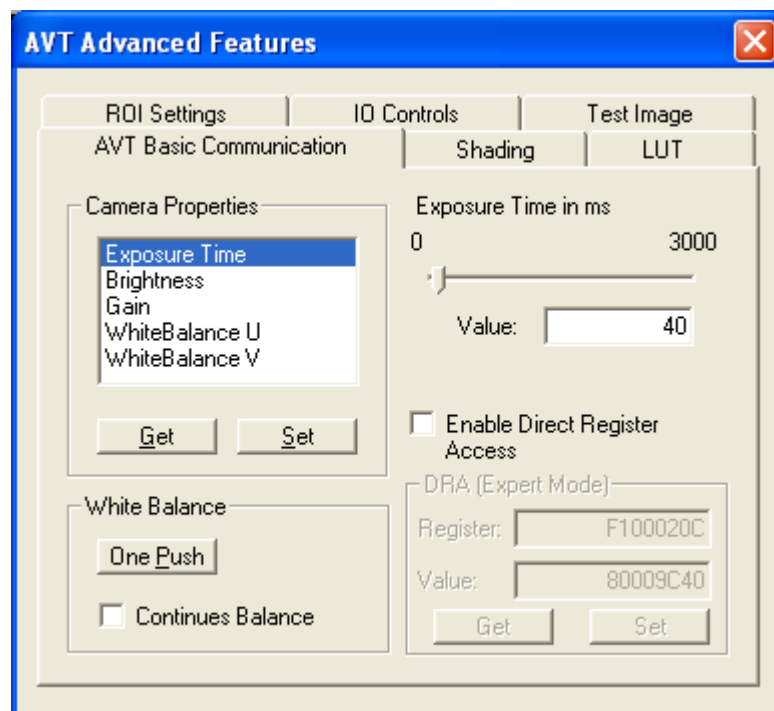
► Using an AVT camera with Common Vision Blox

1.3 AVT Communication Dialog

This dialog grants access to all features that are supported by the camera. Also registers can be written directly. The following frames are available.

1.3.1 AVT Basic Communication

The list box at left side of the frame can be used to select one of the available features (the white balance features are only available for colour cameras):



If selecting a camera feature it's current value will be shown in the edit box and by the slider on the right side of the frame. These controls wrap the settings for the camera registers, which are set by using the CVB IDeviceControl Interface.

The Direct Register Access (the Expert mode) in the right bottom corner allows the writing and reading of camera registers directly, by using the CVB IPort Interface.

Control	Description
<i>Camera Properties</i>	Select the feature for changing (auto read out of the value)
<i>Camera Properties Get</i>	By pushing the Get button the selected feature will be read out and the values will be displayed in the editbox and by using the slider
<i>Camera Properties Set</i>	By pushing the Set button the adjusted value (slider and value-edit box) will be written into the camera register



► Using an AVT camera with Common Vision Blox

<i>Slider</i>	The slider will be used for adjust/display the values
<i>Value-edit box</i>	This edit box will be used to edit/display the values
<i>One Push</i>	One time automatic white balance adjustment by the camera
<i>Continues Balance</i>	Continues automatic white balance adjustment by the camera
<i>Enable Direct Register Access</i>	Enable/Disable the edit boxes and buttons for Direct Register Access (Expert mode)

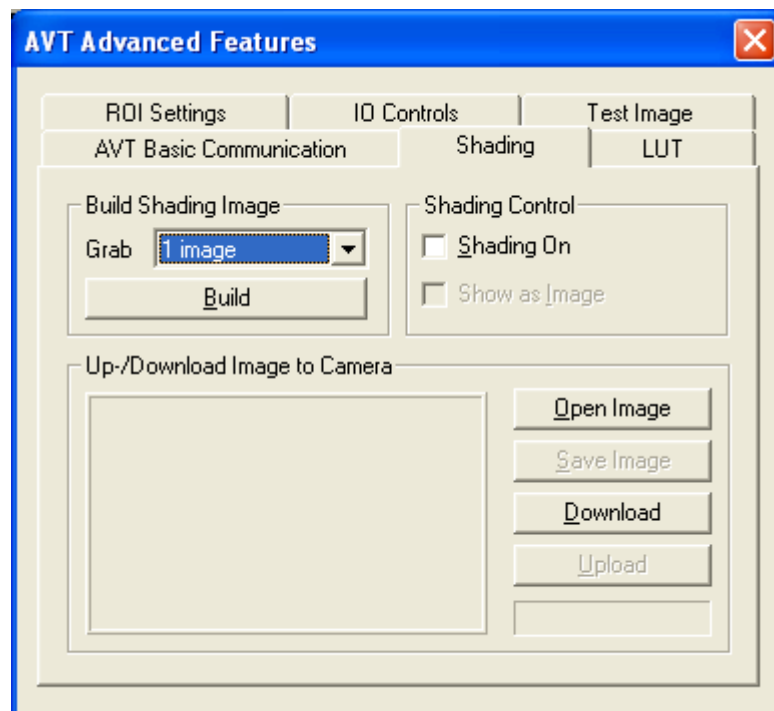
1.3.1.1 Direct Register Access

In the right bottom corner is the *Direct Register Access* frame. In it data can be written directly (*Set* button) to or read (*Get* button) from the current camera by using the CVB IPort Interface.

The address to access can be entered in the *Register* text field. The four-byte-value, which is read or is to be written, is in the *Value* text field as a hexadecimal value.

1.3.2 Shading

A shading image can be used to correct grey level variances due to uneven lighting, lens inhomogeneity, or fixed pattern noise of the sensor.



A shading image is always as large as the sensor and monochrome (containing the Bayer-pattern if a color camera is used).

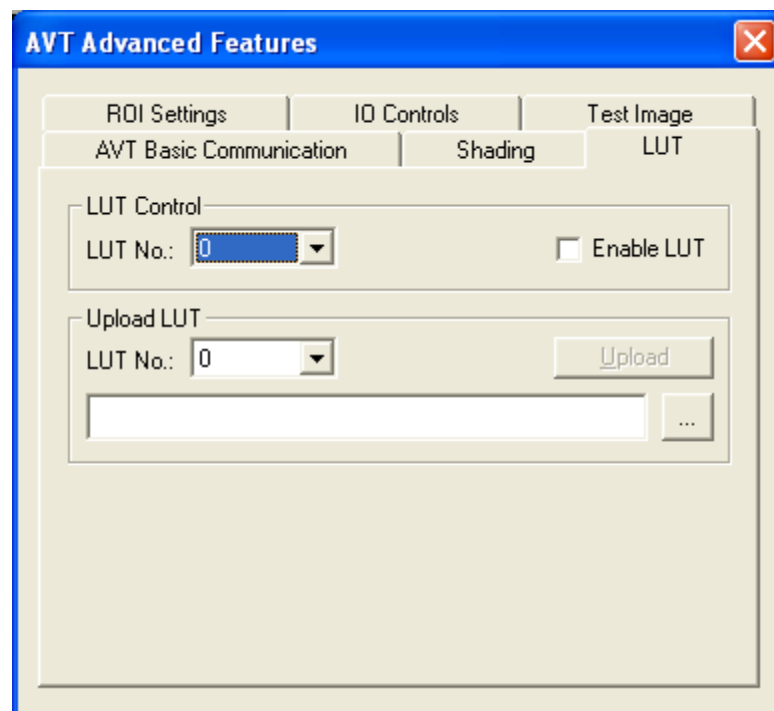


► Using an AVT camera with Common Vision Blox

Frame	Description
<i>Build Shading Image</i>	<p>When the <i>Build</i> button is clicked a shading image is calculated by the camera based on the number of images displayed in the <i>Grab</i> combo box. More than one image can be grabbed for creating the shading image to reduce the effect of pixel-noise.</p> <p>This function mustn't be used while the camera is grabbing!</p>
<i>Shading Control</i>	<p>Here the shading feature can be enabled or disabled by (un)checking the <i>Shading On</i> box. If the <i>Show as Image</i> box is also checked the shading image is displayed instead of the sensor image. This check box is only available when <i>Shading On</i> is checked.</p>
<i>Up-/Download Image</i>	<p>By clicking on <i>Open Image</i> an image can be loaded from disk, which is displayed in the CVB display on the left-hand-side. Also by clicking on download the current shading image saved in the camera is downloaded (progress is shown in the progress bar below).</p> <p>When an image is present in the dialog (displayed), it can be saved to disk by the <i>Save Image</i> button or uploaded by the <i>Upload</i> button (progress is shown in the progress bar below). Only images that have the same size as the sensor can be uploaded.</p> <p>This function mustn't be used while the camera is grabbing!</p>

1.3.3 LUT

Here a Look Up Table (LUT) can be uploaded into the camera.





► Using an AVT camera with Common Vision Blox

A LUT is a grey value function:

$$f(i) = y_i$$

where i is the index of the input grey level from the sensor and y_i is a custom output grey level. With a LUT i.e. the output bit-depth can be reduced or a certain gamma curve can be applied.

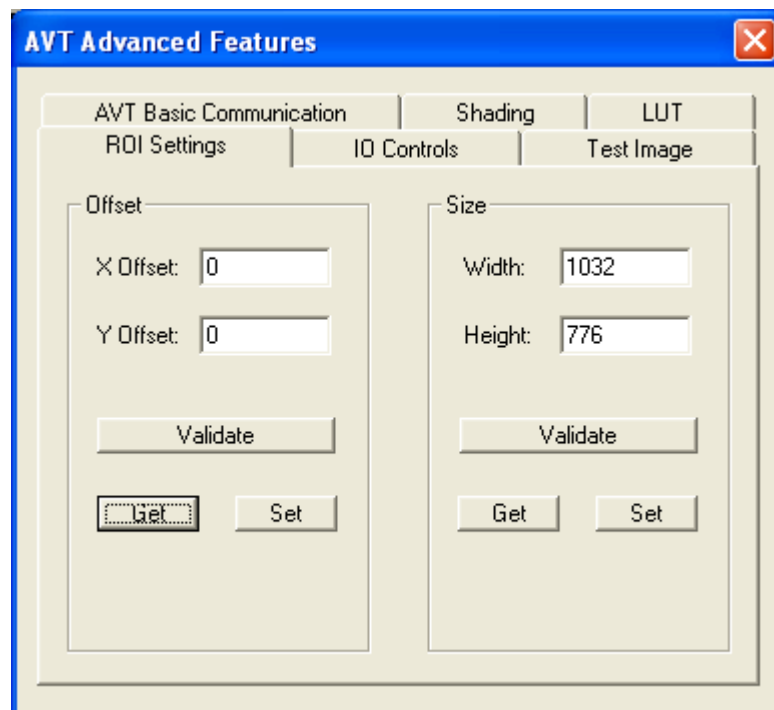
The LUT file that should be uploaded has to be a simple formatted windows text-file: Every line stands for an input grey value. The value written in this line represents the output value. Thus line 255 is the input grey value 255 and if a 128 is written in that line, this is the output value. The length of the LUT-file depends on the input bit-depth: If the input is 10 bit there are 2^{10} entries (or lines) in the LUT file.

The following frames are available:

Frame	Description
<i>LUT Control</i>	Defines which of the available number of LUTs should be enabled or disabled. Only one LUT can be active at one time and either DCAM gamma can be active or the LUT.
<i>Upload LUT</i>	<p>A LUT-file (see above) can be entered in the text box or choosen via a dialog opened by the three-dot-button. When clicking on the <i>Upload</i> button the LUT file entered in the text box will be uploaded to the LUT number specified in the <i>LUT No</i> combo box.</p> <p>If the camera supports it, multiple LUTs can be uploaded and switched via the <i>LUT Control</i> later.</p>

1.3.4 ROI Settings

Here the image size sent by the camera can be changed as well as the image position on the camera sensor.





► Using an AVT camera with Common Vision Blox

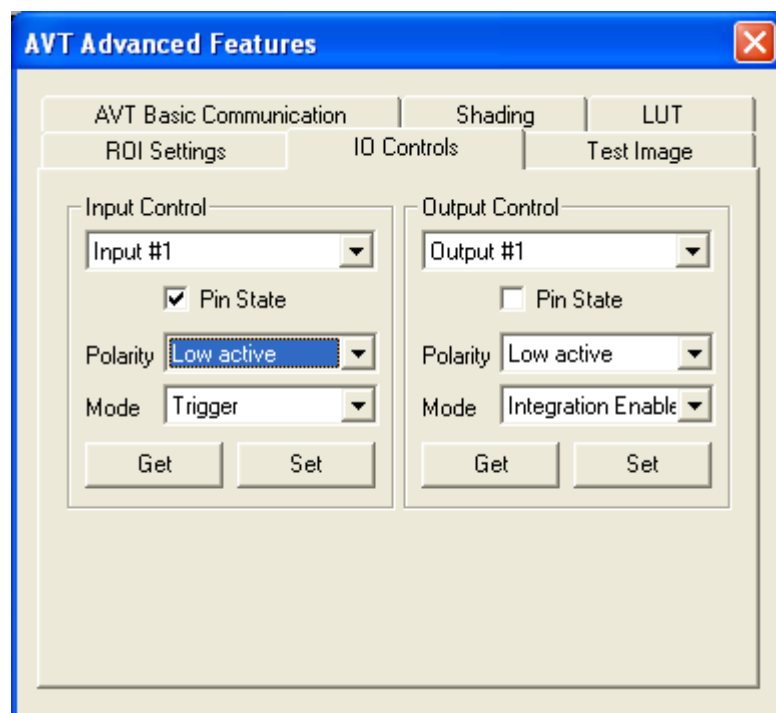
The ROI Settings dialog uses the `ImageRect` interface from CVB (see *Online Documentation*). To change the ROI **grab has to be disabled**. Current settings can be read by clicking on *Get* button, the *Set* button is used to change the values, and the *Validate* button checks whether the entered values are correct.

Frame	Description
<i>Offset</i>	The position of the ROI on the sensor where (0,0) is the top-left corner.
<i>Size</i>	Width and height of the ROI.

All values *Set* to the camera are validated beforehand to prevent false settings in the camera.

1.3.5 IO Controls

This tab contains the Input/Output advanced feature control.



In both frames only the available number of IOs are shown. At the first start-up the current settings of the camera are read. Afterwards the setting for the selected input/output can be saved by clicking the *Set* button or read by clicking the *Get* button.

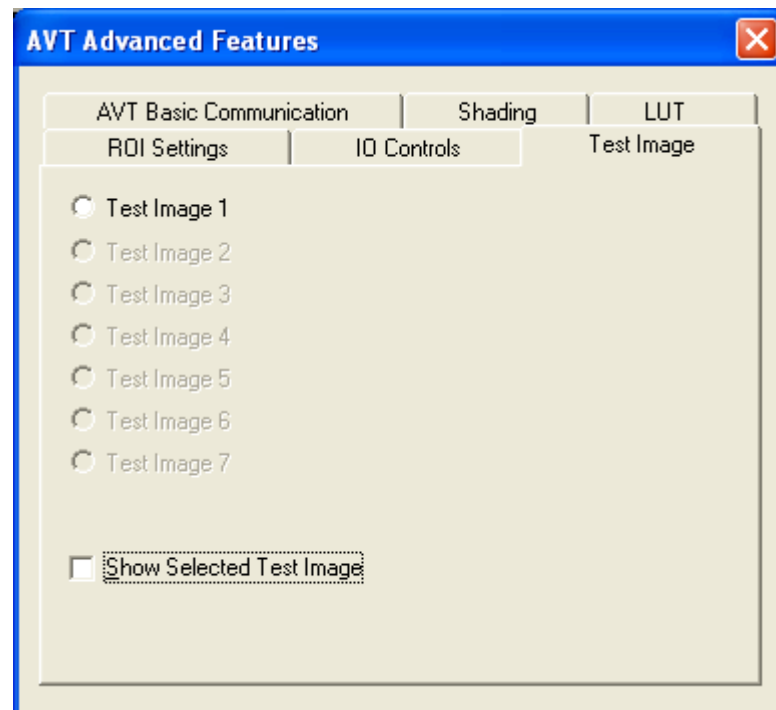


► Using an AVT camera with Common Vision Blox

Frame	Description
<i>Input Control</i>	The topmost combo box is used to select the input number. When changing the number the current setting for it is read. The <i>Pin state</i> checkbox is readonly and signals whether the input is currently high or low (high when open). The <i>Polarity</i> can be toggled between high or low active. The <i>Modes</i> available are those of a Dolphin. Thus check beforehand which features are really available.
<i>Output Control</i>	Is used like the <i>Input Control</i> . If you want to control the output directly you have to switch the <i>Mode</i> to <i>Follow Pin State</i> . Now by checking/unchecking the <i>Pin State</i> box the output can be toggled.

1.3.6 Test Image

In this tab the display of test images can be enabled or disabled.



On the left-hand side the available test images are shown. By clicking on the *Show Selected Test Image* checkbox the chosen test image can be displayed. This feature can be used either when grab is active or not.

The test image is displayed in the CVB-Display in the main window of this application.



► Using an AVT camera with Common Vision Blox

2. VB CVAVT1394 Example

2.1 First start of the Visual Basic Example Application

After installing Common Vision Blox (CVB) and the IEEE AVT 1394(CVAVT1394.vin) driver on your system according to the CVB Hardware Quickstart 1394 AVT, you can find the application in:

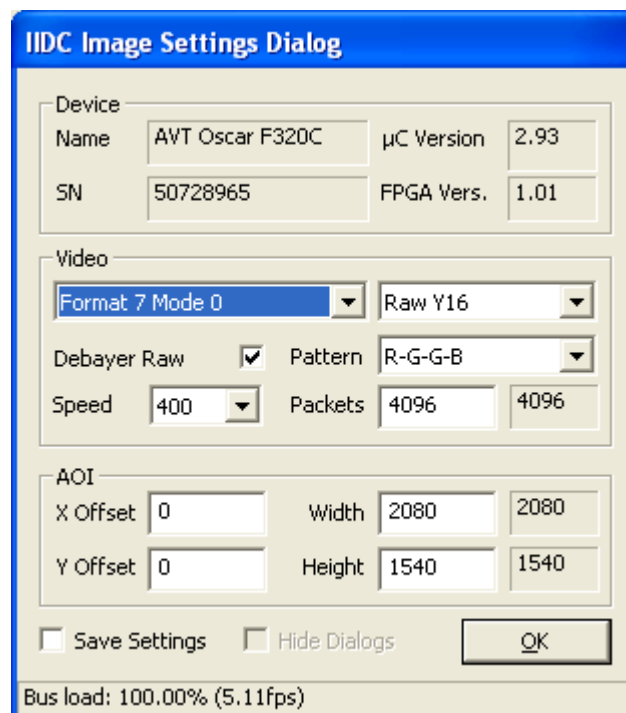
Start → Programs → Common Vision Blox → Hardware → AVT → 1394 → VB AVT1394 Example

If you are interested in the source code you can find it in

Tutorial\Hardware\AVT1394\VB\VB AVT1394

based on the CVB directory (default is C:\Program Files\Common Vision Blox).

As described in the QuickStart a configuration window will appear:



The cvAVT1394.vin support only format 7 modes.

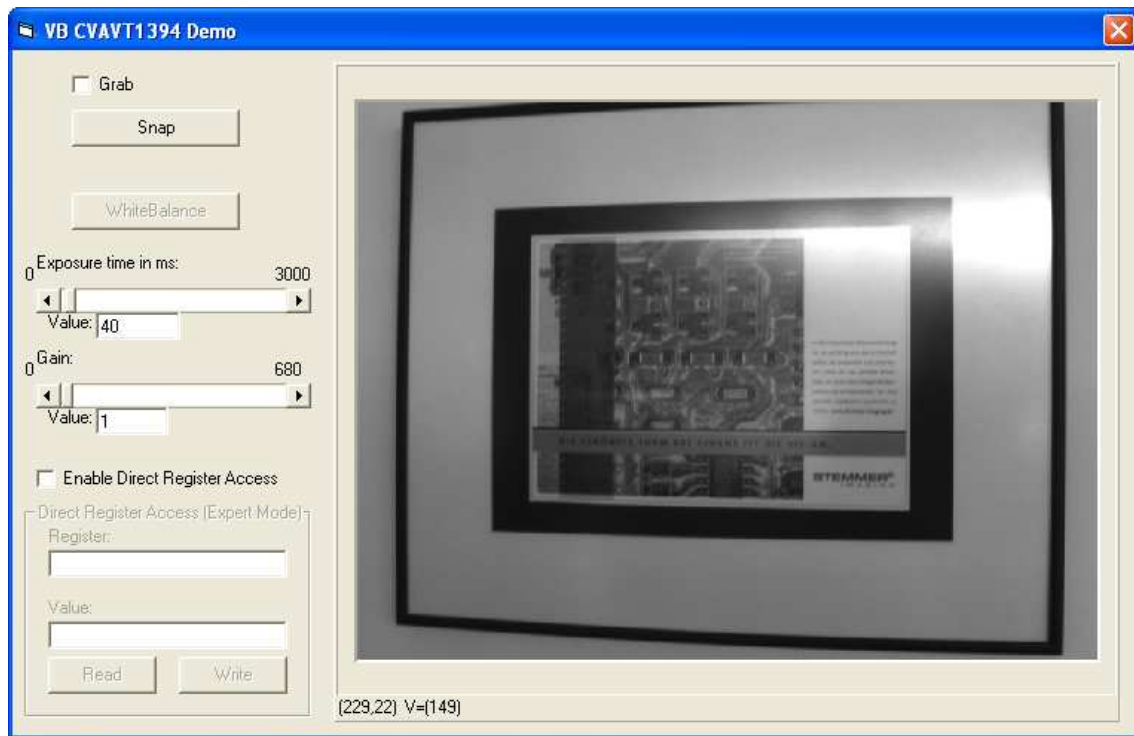
Format 7 is a scalable format without a fixed resolution, frame-rate and colour mode. The modes available are defined by the camera vendor and can be found in the camera manual. If you are using a colour camera with raw data output please check Debayer Raw and select the pattern. By default CVB uses the whole sensor-resolution and the maximum speed & packet size.

After applying the desired settings the main window of the application is shown:



► Using an AVT camera with Common Vision Blox

2.2 Graphical User Interface



Shown above is the window of the *VB AVT 1394 Example*. On the right-hand side you can see the CVB-Display containing the image and a status bar. The status bar displays information on the actual image coordinates the mouse hovers over (no screen coordinates) and the (colour) information of that pixel. Also when images are grabbed the Frames Per Second (fps) are displayed.

On the left-hand side are a number of controls:

Control	Description
<i>Grab</i>	Starts/stops image acquisition
<i>Snap</i>	Acquires single image
<i>WhiteBalance</i>	If the plugged camera is a colour camera, this button can be used for one push automatic white balance adjustment by the camera
<i>Upper Slider</i>	The slider will be used to edit/display the exposure time values (auto read out on start)
<i>Upper Value-edit box</i>	This edit box will be used to edit/display the exposure time values (auto read out on start)
<i>Lower Slider</i>	The slider will be used to edit/display the gain values (auto read out on start)
<i>Lower Value-edit box</i>	This edit box will be used to edit/display the gain values (auto read out on start)
<i>Enable Direct Register Access</i>	Enable/Disable the edit boxes and buttons for Direct Register Access (Expert mode)



► Using an AVT camera with Common Vision Blox



These controls (slider and edit boxes) wrap the settings for the camera registers, which are set by using the CVB IDeviceControl Interface.

2.2.1 Direct Register Access

In the left bottom corner is the *Direct Register Access* frame. In it data can be written directly (*Write* button) to or read (*Read* button) from the current camera by using the CVB IPort Interface.

The address to access can be entered in the *Register* text field. The four-byte-value, which is read or is to be written, is in the *Value* text field as a hexadecimal value.



► Using an AVT camera with Common Vision Blox

3. Delphi CVAVT1394 Example

3.1 First start of the Delphi Example Application

After installing Common Vision Blox (CVB) and the IEEE AVT 1394(CVAVT1394.vin) driver on your system according to the CVB Hardware Quickstart 1394 AVT, you can find the application in:

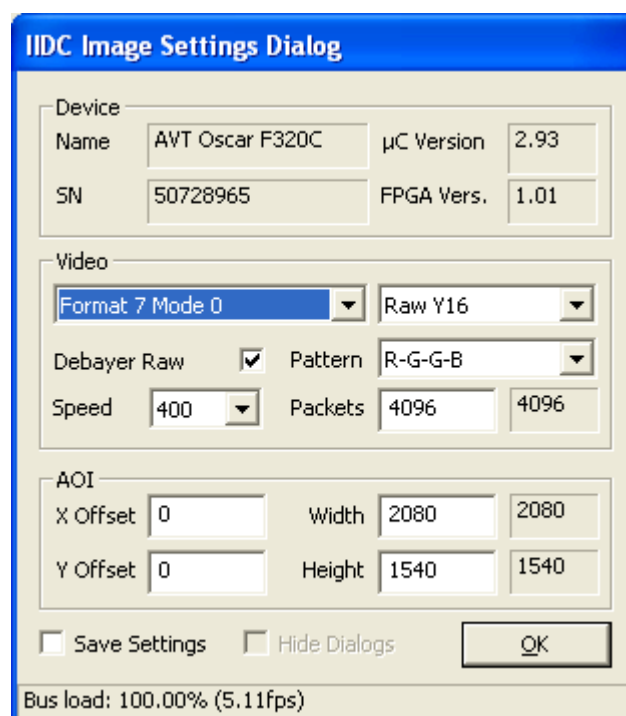
Start → Programs → Common Vision Blox → Hardware → AVT → 1394 → Delphi AVT1394 Example

If you are interested in the source code you can find it in

Tutorial\Hardware\AVT1394\Delphi\DelphiAVT1394

based on the CVB directory (default is C:\Program Files\Common Vision Blox).

As described in the QuickStart a configuration window will appear:



The cvAVT1394.vin support only format 7 modes.

Format 7 is a scalable format without a fixed resolution, frame-rate and colour mode. The modes available are defined by the camera vendor and can be found in the camera manual. If you are using a colour camera with raw data output please check Debayer Raw and select the pattern. By default CVB uses the whole sensor-resolution and the maximum speed & packet size.

After applying the desired settings the main window of the application is shown:



► Using an AVT camera with Common Vision Blox

3.2 Graphical User Interface



Shown above is the window of the *Delphi AVT 1394 Example*. On the right-hand side you can see the CVB-Display containing the image and a status bar. The status bar displays information on the actual image coordinates the mouse hovers over (no screen coordinates) and the (colour) information of that pixel. Also when images are grabbed the Frames Per Second (fps) are displayed.

On the left-hand side are a number of controls:

Control	Description
<i>Grab</i>	Starts/stops image acquisition
<i>Snap</i>	Acquires single image
<i>WhiteBalance</i>	If the plugged camera is a colour camera, this button can be used for one push automatic white balance adjustment by the camera
<i>Upper Slider</i>	The slider will be used to edit/display the exposure time values (auto read out on start)
<i>Upper Value-edit box</i>	This edit box will be used to edit/display the exposure time values (auto read out on start)
<i>Lower Slider</i>	The slider will be used to edit/display the gain values (auto read out on start)
<i>Lower Value-edit box</i>	This edit box will be used to edit/display the gain values (auto read out on start)
<i>Enable Direct Register Access</i>	Enable/Disable the edit boxes and buttons for Direct Register Access (Expert mode)



► Using an AVT camera with Common Vision Blox



These controls (slider and edit boxes) wrap the settings for the camera registers, which are set by using the CVB IDeviceControl Interface.

2.2.1 Direct Register Access

In the left bottom corner is the *Direct Register Access* frame. In it data can be written directly (*Write* button) to or read (*Read* button) from the current camera by using the CVB IPort Interface.

The address to access can be entered in the *Register* text field. The four-byte-value, which is read or is to be written, is in the *Value* text field as a hexadecimal value.