Getting Started with the BitFlow SDK 5.00

Installation, Configuration and Testing

Before Installation

1. You must log onto the computer using an account that has administrative privileges.

2. Uninstall any previous BitFlow SDK releases before installing this release.

3. You should install the SDK first, before installing any BitFlow boards in your computer. The installation process will be simpler if the Windows Hardware Wizard finds a new board after the drivers have been installed.

5. If you are upgrading from a previous release, please be sure to read the "changes.txt" file to see what is new and what has changed in this release.

6. This SDK is compatible with 32-bit versions of Windows XP, Server 2003, Vista and the 64-bit versions of Windows XP, Server 2003 and Vista.

Serial Numbers

If you purchased the SDK, your serial number will be on a label affixed to the CD-ROM case. If you are upgrading from a release older than SDK 3.00, you must obtain a serial number from BitFlow. If you are upgrading from SDK 3.00 or later, you should already have a serial number. In any case, use this serial number when prompted by the installer. If you are upgrading a computer from a previous release of the BitFlow SDK, your serial number will already appear in the installer dialog. If you are using a third party application or just want to run our simple example applications, you can install the binary components of the SDK by using the serial number "0". There is no charge to use or install the SDK using the serial number "0".

Install the SDK

1. If you have a CD-ROM, insert it into your computer. The CD-ROM will auto-run an installation screen, from here you can start the software installer. If you downloaded the SDK, just run the installer directly.

2. Follow the instructions of the installer.

3. When the installer is finished, turn the computer off and install your BitFlow board(s).

4. If you are upgrading and have left your boards in your system, you may be prompted to reboot your computer.

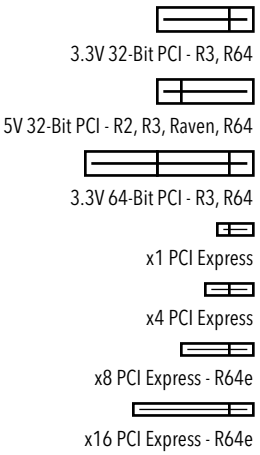
Install your BitFlow hardware

1. Make sure you observe proper static protection procedures when installing any hardware in a PC.

2. Make sure the PC is powered down and is not connected to the AC supply.

3. Open the PC. Locate an unused PCI or PCIe slot of the correct type for you board. See the illustration at right to help identify the correct type of slot.

4. Insert the BitFlow board in this slot. Make sure the board is firmly seated in the PCI or PCIe connector.



5. Secure the board to the PC with a screw on the edge connector.

6. Close the PC.

7. Connect your BitFlow board to your camera using a cable of the correct type.

8. Reconnect the AC power cord to your PC.

9. Power up your camera.

10. Power up and boot your PC.

After Rebooting

1. For plug and play operating systems, the New Hardware Found Wizard will pop up.

2. Follow the Wizard's instruction to install the default driver (recommended).

3. At some point during this procedure, the Wizard will display a message saying "digital signature not found" or "...has not passed Windows Logo Testing". This message is in no way an indicator of insufficient performance or functionality. Press the **Continue** button in this dialog to proceed with the installation.

Testing an R3 Camera Link Model

1. Run SysReg (Start Menu > Programs > BitFlow > SDK 5.00 > SysReg)

2. Make sure the Camera configuration file path is correct. If you installed the SDK in the default directory, the configuration directory should be "C:\BitFlow SDK 5.00\Config".

3. All of the BitFlow boards in your system should appear in the BitFlow boards found list. Highlight a board in this list then click the Board operation: **Configure** button to open the Board Details dialog. If no boards appear in this list or boards are marked "Missing", go to the trouble shooting section.

4. Click the Camera file operation: **Add** button to open the Choose Camera File dialog.

5. Navigate through the tree to the following entry:

+ Make: BitFlow
  + Model: Synthetic
    + Mode: Free Run
      + BFSynth256E1.rcl.

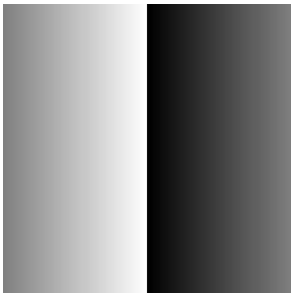
6. Click on "BFSynth256E1.rcl" and then click **OK**.

7. Click **OK** to the Board Details dialog.

8. Click **OK** to the System Configuration dialog.

9. Now run CiView.

10. A display surface will open and you should see a wedge pattern similar to the image on the right. If you see this image, the system is working fine. If you do not see this pattern, go to the Troubleshooting section.



Road Runner and R3 Camera Link model synthetic camera output

Testing an R3 Differential Model

1. Run SysReg (Start Menu > Programs > BitFlow > SDK 5.00 > SysReg)

2. Make sure the Camera configuration file path is correct. If you installed the SDK in the default directory, the configuration directory should be "C:\BitFlow SDK 5.00\Config".

3. All of the BitFlow boards in your system should appear in the BitFlow boards found list. Highlight a board in this list then click the Board operation: **Configure** button to open the Board Details dialog. If no boards appear in this list or boards are marked "Missing", to the trouble shooting section.

4. Click the Camera file operation: **Add** button to open the Choose Camera File dialog.

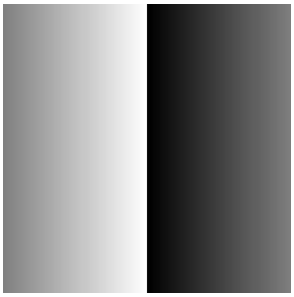
5. Navigate the file open dialog and select "BFSynth256E1.cam". Click **OK**.

6. Click **OK** to the Board Details dialog.

7. Click **OK** to the System Configuration dialog.

8. Now run CiView.

9. A display surface will open and you should see a wedge pattern similar to the image on the right. If you see this image, the system is working fine. If you do not see this pattern, go to the Troubleshooting section.



Road Runner and R3 differential model synthetic camera output

Testing an R64, Karbon or Neon

1. Run SysReg (Start Menu > Programs > BitFlow > SDK 5.00 > SysReg)

2. Make sure the Camera configuration file path is correct. If you installed the SDK in the default directory, the configuration directory should be "C:\BitFlow SDK 5.00\Config".

3. All of the BitFlow boards in your system should appear in the BitFlow boards found list. Highlight a board in this list then click the Board operation: **Configure** button to open the Board Details dialog. If no boards appear in this list or boards are marked "Missing", to to the trouble shooting section.

4. Click the Camera file operation: **Add** button to open the Choose Camera File dialog.

5. Navigate through the tree to the following entry:

+ Make: BitFlow
  + Model: Synthetic 1024 x 1024
    + Mode: Free run
      + BitFlow-Synthetic-1024x1024-E1.r64

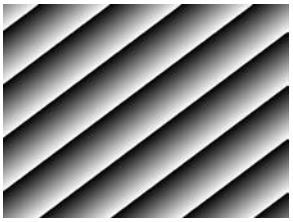
6. Click on "BitFlow-Synthetic-1024x1024-E1.r64" and then click **OK**.

7. Click **OK** to the Board Details dialog.

8. Click **OK** to the System Configuration dialog.

9. Now run CiView.

10. A display surface will open and you should see a wedge pattern similar to the images on the right (the image will appear to continuously shift horizontally). If you see this image, the system is working fine. If you do not see this pattern, go to the Troubleshooting section.



R64, Karbon and Neon synthetic camera output

Testing an Alta

1. Run SysReg (Start Menu > Programs > BitFlow > SDK 5.00 > SysReg)

2. Make sure the Camera configuration file path is correct. If you installed the SDK in the default directory, the configuration directory should be "C:\BitFlow SDK 5.00\Config".

3. All of the BitFlow boards in your system should appear in the BitFlow boards found list. Highlight a board in this list then click the Board operation: **Configure** button to open the Board Details dialog. If no boards appear in this list or boards are marked "Missing", to to the trouble shooting section.

4. Click the Camera file operation: **Add** button to open the Choose Camera File dialog.

5. Navigate through the tree to the following entry:

+ Make: BitFlow
  + Model: Synthetic
    + Mode: B&W, 1024x1024, Free Run
      + BitFlow-Synthetic-1024x1024-E1.anlg

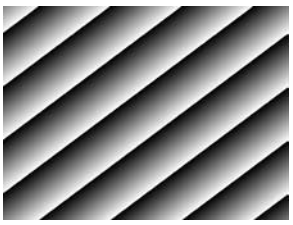
6. Click on "BitFlow-Synthetic-1024x1024-E1." and then click **OK**.

7. Click **OK** to the Board Details dialog.

8. Click **OK** to the System Configuration dialog.

9. Now run CiView.

10. A display surface will open and you should see a wedge pattern similar to the images on the right (the image will appear to continuously shift horizontally). If you see this image, the system is working fine. If you do not see this pattern, go to the Troubleshooting section.



Alta synthetic camera output

Configure your board for your camera

1. Run SysReg.

2. Highlight the board in the BitFlow boards found list, then click the Board operation: **Configure** button to open the Board Details dialog.

3. Highlight the existing configuration file in the Attached camera list and click the **Change** button.

4. Navigate the configuration file tree to get the Make, Model, and Mode of your camera, click on the file name, then click the **OK** button. For differential boards, you will get a simple file open dialog. If you are not sure which file is for your camera, see the document: "BitFlow SDK 5.00\docs\Camera File List.txt". This document lists all camera configuration files and their associated camera model names.

5. The default firmware works for almost all cameras, check the camera configuration file for notes on firmware changes. For the R64/Karbon/Neon/Alta families, the correct firmware is automatically loaded to the board when it is initialized.

6. Click **OK** to the Board Details dialog.

7. Click **OK** to the System Configuration dialog.

8. Make sure your camera is powered up and connected to the board.

9. Run CiView again. You should see live video from your camera. If you do not see an image, go to the Troubleshooting section.

INSTALLED COMPONENTS

The following components are copied to your computer when installing the full SDK.

Read Me - The latest information about the release

SysReg - System configuration utility: associates cameras with installed boards.

BayView - Bayer filter type camera conversion and display application.

BiFlow - Sequence capture application example based on the BufIn API

BiProcess - Simple continuous processing application example based on the BufIn AP.

BFCom - HyperTerminal like application for camera communication through the Camera Link board's serial port.

Circ - Circular buffer processing application example based on the BufIn API.

CamEd - Camera configuration file editor.

PCIWalk - PCI configuration utility.

CamVert - Low Level camera configuration editor.

VerCheck - Displays DLL and driver version information.

DLLs - Dynamic Link Libraries: these contain all of the functionality of the SDK.

Camera Configuration Files - Used to configure the board for your cameras.

Kernel Driver - Provides access to the BitFlow hardware.

Documents - Text files with the detailed information about the release.

Source Code - Complete C and C++ source for all examples and user level DLLs.

Header files - Definitions required for the API.

Libraries - Link libraries for the Microsoft compliers.

Developing your application

Once you are confident that your board is working correctly

The next step is to develop your application. The examples provided should allow you to quickly integrate the BitFlow SDK functionality into your program.

If you need something simple

CiSimple, - This programs show the minimum function calls required to get images into the host memory. It should be very easy to integrate these same function calls into your application. These applications do not provide a GUI. However, they are extremely easy to understand and build upon. Because the example CiSimple uses the Ci API, it works with all product families. It is recommended that you start with this example as it will provide the most flexibility in the future.

If you need to view and save live images

CiView, BayView (for Bayer cameras) - These applications demonstrate real-time display and also allow you to snap single images into separate buffers, which can then be processed and/or saved to disk. It is quite simple to extend these programs by adding your own processing functions or file formats.

If you need real-time processing

BiProcess, BayView, Circ - These applications demonstrate multiple host buffer acquisition. The buffers are used in a circular manner. Each buffer in turn receives DMA data from the board, while the other buffers are available for processing. This allows the CPU to access an entire image without data being overwritten by the board. BiProcess dœs a simple histogram calculation, BayView performs a de-mosaicing operation on the incoming images and Circ simply displays each image in turn. All of these examples are good starting points for your application, you just need to plug in your own processing functions.

If you need to acquire sequences of images

BiFlow - This application acquires sequences of images in real time to host memory using the BufIn API. The resulting sequence can be displayed one at a time on the VGA, played back as a sequence and/or saved to disk. This application captures images in the cameras native format (i.e. pixel bit depth) for internal storage and for saving to disk.

If you need to acquire and process images via a circular set of buffers

Circ - This application demonstrates the circular buffer processing portion of the BufIn API. The concept is that the board is acquiring continuously into a set of buffers. Unsynchronized with acquisition, the host is processing the images in the buffers. The number of buffers required is dependent on the variability of acquisition and processing time. The BufIn API makes this type of an application very easy to write. Circ fully demonstrates the power of this API. You can also click on any buffer to remove it from the list of available buffers. Clicking on this same buffer again, re-inserts into the buffer list. This process demonstrates the new Add/Drop buffer functionality.

If you want to use the buffer management functions (BufIn)

BiSimple, BiSimplePlusPlus, CircClassExample, CircHoldSimple - These project folders contain a wealth of example console program. Because they are console programs the code is simple and easy to understand, unencumbered by messy GUI code. These examples illustrate most of the major components of the BufIn interface, in both C and C++. Each example is a self-contained lesson in an important feature of the BufIn API.

Whatever example you decide to start with, you must build and modify it with a C compiler. We provide source and project files to build all of our examples with Microsoft Visual Studio 2005 or above. However the BitFlow API is fully WIN32 compliant and can be called from any WIN32 programming environment, for example Microsoft Visual Basic. Please refer to the "readme.txt" file to see how to get started with this and other environments.

If you need to understand how your board works

There is a hardware reference manual for each family of boards. These manuals are available on the SDK CD and can also be downloaded from our web site. The following manuals are available:

R3-DIF - Covers Road Runner and R3 differential models

R3-CL - Covers R3 Camera Link models

R64 - Covers R64 and R64e models

Karbon - Covers all Karbon models

Neon - Covers all Neon models

Alta - Covers all the Alta models.

If you need to understand how the software works

There is a comprehensive software manual, The Software Reference Manual, available on the SDK CD or from our web site This manual describes every function in the BitFlow SDK API in detail. Also included are overview chapters which describe how to use the different APIs from a high level.

If you are using the BufIn C++ interface the documentation can be found in HTML form on our web site. As the BufIn C++ class is just a wrappter around the BufIn class, it will be helpful to see the BufIn functions in the The Software Reference Manual

Working with third party software

The functionality of the BitFlow SDK is greatly extended by the addition of third party software. If you are going to use one of these products, then you will need to install a driver that provides an interface between the third party product and the BitFlow SDK. Usually you will have to install the BitFlow SDK before installing the individual 3rd party imaging software product driver. However, some imaging software products install the driver and/or the SDK automatically. Please consult the documentation of the individual imaging software product for details.

When installing the BitFlow SDK for use with third party software, you can enter the serial number "0", which will tell he installer to just install the BItFlow driver and other binary components.

Drivers for third party software packages are available from the download sections of our web site or from their respective vendors. All of our drivers are based on BitFlow's Common Interface library (Ci) and will work with whichever BitFlow board is installed in your system (i.e a single driver for Road Runner, R3, R64, Karbon, Neon and Alta).

Troubleshooting your system

Here are some simple things you can check if your system is not working correctly:

If computer does not boot, or the computer hangs during the boot process:

This problem is usually caused by another device conflicting with the board. Try removing or disabling all other non-critical PCI devices.

Try put the board in another PCI/PCIe slot.

Try reserving the IRQ of the device conflicting with the board, this can be done in some BIOS's.

If SysReg shows an error message and does not run or no boards appears in SysReg, or only some of your boards appear in SysReg, or boards are marked as "Missing":

These symptoms may be seen when upgrading a system which has a previous version of the BitFlow SDK on it. On some system the Plug and Play Wizard gets confused. These problems are easily solved using the device manager as follows.

First try rebooting your computer, this often clears up many problems.

If this does not work try the following procedure:

1. Run the device manager (Start > Control Panel > Administrative Tools > Computer Management).

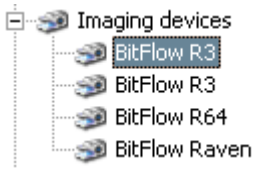
2. Navigate to the BitFlow devices (Computer Management > System Tools > Devices > Imaging Devices OR Computer Management > System Tools > Devices > Other Devices).

3. All of the BitFlow boards should look like the image at top-right.

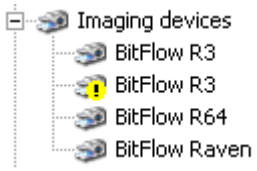
4. If any boards appear with the error icons as shown in the three lower images at the right, then there is a problem with the plug and play configuration. To fix this problem right click on the device in question and select **Update** or **Enable**. Follow the instructions of the New Hardware Wizard (select "Install the software automatically"). You will see a Windows warning dialog, click continue. After the Hardware Wizard is finish, the device should work properly.

5. Repeat step 4 for all devices that have error icons.

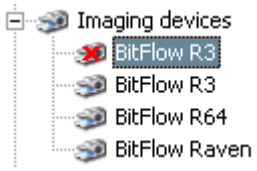
6. You may need to reboot your computer after making these changes.



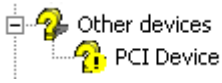
Device is OK



Yellow "!" - Configuration problem



Red "X" - Device disabled



Yellow "!" in other devices - Configuration problem

If the synthetic image does not appear in one of the viewing applications:

Try the board in another PCI/PCIe slot.

Remove all PCI/PCIe cards except the BitFlow board and your display card.

Try the BitFlow board in another PC.

Try updating the drivers for you display card.

If you get time out errors in one of the viewing applications with your camera:

Make sure your camera has power.

Check the cable between the camera and the board to ensure all the connectors are tight.

Make sure you have selected the correct configuration file for your camera.

Make sure your camera is set in the mode for which configuration file was designed.

If you are using an encoder or trigger, make sure that they are sending the correct signals to the board (see the hardware reference manual for interface specifications). Confirm that the camera works in free running mode first.

Check the overall PCI/PCIe DMA throughput using the application CiBench . Your system should be at least 20% faster than your camera's data rate. If you are unsure of your camera's data rate use the following formula:

DataRate (MB/S) = *CameraTaps* × *PixelClockFrequency* × *BytesPerPixel*

Where:

CameraTaps = The number of data streams coming out of the camera (usually correlates with the sensor design).

PixelClockFrequency = The frequency of the data strobe clocking out the pixel data.

BytesPerPixel = 1 (for monochrome or 8-bit cameras)

BytesPerPixel = 2 (for cameras with 9 to 16-bit pixels)

BytesPerPixel = 3 (for packed (RGB) color cameras)

BytesPerPixel = 4 (For unpacked (ARGB) color cameras)

MB/S = Megabytes per second

The formulas define the maximum possible data rate produced by the camera, the average data rate may be somewhat slower.

If the video display is blank and you don't get time out errors in one of the viewing applications:

Check the lens cap and F-Stop settings.

Try pointing the camera directly at a light or window. Some cameras need a very bright light to provide an image.

Check your camera's gain and other adjustments.

If you get video but it is skewed, scrolling or looks out of sync:

Double check that you are using the correct camera file for the camera you have attached to the board and for the mode that the camera is in. If you are not sure which camera file to use, edit the file in CamEd, and check to make sure the comments in the file agree with your camera/mode.

Make sure that you are using the correct firmware for your camera .

If your camera has a binning mode, make sure that your are using a binning camera file.

Getting help

There are many resources, beyond this document, that are available to you if you have trouble getting started:

The manuals There is a wealth of information in both the SDK manual as well as the individual hardware manuals.

Our web site: www.bitflow.com

E-mail: support@bitflow.com

Forum: http://www.bitflow.us/Forum

Telephone 781-932-2900 (9 AM–6 PM EST)

Facsimile 781-933-9965

Please have the following available when seeking support:

Model and serial number (on a small yellow sticker) of your board.

Make and model of camera and mode being used.

Manufacturer and model of camera cable.

Software revision, run VerCheck to find the installed version information.

Computer CPU type, PCI chipset, and bus speed.

Operating system, service pack.

Example code (if applicable).

Text file saved from VerCheck.

The event viewer log from the period of time when the problem occurred.