

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Teledyne Dalsa / Coreco Imaging IC-ASYNC* at our website: *Click HERE*

Coreco Imaging, Inc

IC-ASYNC

User's Manual

405-00002-00

November 2000
Rev 02

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IC-ASYNC User's Manual
Document Number 405-00002-00
Rev 02; November 10, 2000

Copyright © Coreco Imaging, Inc 2001
Copyright © transferred to Coreco Imaging, Inc 2000
Copyright © Imaging Technology Incorporated 2000–1998
All rights reserved.
Printed in the United States of America.

All copyrights in this manual, and the hardware and software described in it, are the exclusive property of Coreco Imaging, Inc and its licensors. Claim of copyright does not imply waiver of Coreco Imaging, Inc's or its licensor's other rights in the work. See the following Notice of Proprietary Rights.

NOTICE OF PROPRIETARY RIGHTS

This manual and the related hardware and software are confidential trade secrets and the property of Coreco Imaging, Inc and its licensors. Use, examination, reproduction, copying, transfer and/or disclosure to others of all or any part of this manual and the related documentation are prohibited except with the express written consent of Coreco Imaging, Inc.

Portions of the software library use the Halo Image File Format Library, which is a copyrighted product of Media Cybernetics.

The information in this document is subject to change without notice. Coreco Imaging, Inc makes no representations or warranties with respect to the contents of this manual and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Coreco Imaging, Inc assumes no responsibility for errors or omissions in this document.

Sherlock, SMART Search, PCVision, and the Coreco Imaging logo are trademarks of Coreco Imaging, Inc. MVTools, SherlockPro, PCVision*plus* and ITEX are Registered Trademarks of Coreco Imaging, Inc.

All other trademarks are property of their respective owners.

Coreco Imaging, Inc
(formerly known as Imaging Technology Incorporated)
55 Middlesex Turnpike
Bedford, MA 01730-1421
+1.781.275.2700
FAX: +1.781.275–9590
info@imaging.com
http://www.imaging.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EMC/EMI Precautions

This product has been tested and found to comply with the FCC limits for a class A device. The intended use of this product is in a commercial, industrial work environment. Operation of this product within a domestic environment may cause radio interference, in which case the user may be required to take adequate measures.

This product has been tested and found to comply with the European Community Directive for EMC, 89/336/EEC. In order to maintain compliance with the EMC Directive this product must be installed in an EEC EMC compliant chassis that has a CE marking. Additionally, all cables used to connect the computer and peripherals must be shielded, grounded and securely fastened.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents

1. Introduction

Welcome to the IC-ASYNC User's Guide 1
Introduction and Overview 1

2. Installation and Setup

Software Installation 2
Environment Changes 3
Hardware Installation 4
Diagnostics Utility 5
Connect a Camera and Acquire an Image 6
ITEX Configurator Utility 7

3. Principles of Operation

Features 11
Video Input 12
Camera Interface 13
Image Memory 14
Acquisition 14
Memory Access 15
External Trigger And Strobe 15
Optical Isolation 16
Parallel I/O 16
Camera Power 17
PCI Interface 17
Bus Master Operation and Scatter Gather 18
Clipping and Padding 18
Video Overlay 19
Destructive Overlay 19
Destructive Mix in VGA Memory 19
Destructive Mix in Host Memory 20
Non-Destructive Overlay 20
Bus Master to the Primary Surface 21
Limitations 21
Bus Master to the Secondary Surface 22

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Tutorials and Program Examples

Camera Configurator 25
Phase-Locked Loop (PLL) or Crystal (XTAL) Mode 26
Triggered Acquisition 29
Triggered Acquisition (With Frame Reset) 30
Variable Scan Mode 32
Interfacing Dual Channel Cameras 33
Tutorial 35
IC-ASYNC AppWizard 39
IC-ASYNC Generated Code 47
Dual-Tap Cameras 54
Four Channel Triggered Acquisition 56
Added Code 60

Appendix A. Video Connections

Connectors 1
15-Pin Connector CAM0 and CAM1 2
26-Pin Connector CAM2, CAM3, and MISC 3
Parallel I/O Connector 4
Break-out Cables 5
Camera Adapter Cables 10
BNC Camera Adaptor 10
Hirose (JIT) 1 11
Hirose (JIT) 2 12
CV-M10E/C 13
CV-M10BX 14
Dual-Tap Cameras 15

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

List of Tables

Number	Title	Page
Table A-1.	15-Pin Connector	2
Table A-2.	26-Pin Connector	3
Table A-3.	26-Pin Dual-Row Header	4
Table A-4.	Cables for IC-ASYNC	5
Table A-5.	BCBL-PCV1 Pin-out	7
Table A-6.	BCBL-PCV2 Pin-out	8
Table A-7.	ACBL-BNC Cable Pin-Out	10
Table A-8.	ACBL-HIR1 Cable Pin-Out	11
Table A-9.	ACBL-HIR2 Cable Pin-Out	12
Table A-10.	CBVM10-6 Cable Pin-Out	13
Table A-11.	ACBL–VSCAN Cable Pin-Out	14
Table A-12.	ACBL–DTAP–15P Cable	15
Table A-13.	ACBL–DTAP–26P Cable	16

List of Figures

Number	Title	Page
Figure 1.	Single Production Line with Multiple Inspections	1
Figure 2.	Setup Type	3
Figure 3.	ITEX Program Group	4
Figure 4.	Install IC-ASYNC in Computer	4
Figure 5.	Diagnostic Program Report	5
Figure 6.	Single Camera Connection	6
Figure 7.	Multiple Camera Connection	6
Figure 8.	Configurator Start Up Screen	7
Figure 9.	Camera Selection	8
Figure 10.	Live Image Acquire from Camera Configurator	8
Figure 11.	IC-ASYNC Block Diagram	10
Figure 12.	Input Gain Stage	13
Figure 13.	Using Programmable Window Generator to Capture an Area-of-Interest	14
Figure 14.	Optical Isolation	16
Figure 15.	Fuse Location	17

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 16. Destructive Mix in VGA Memory 20
Figure 17. Destructive Mix in Host Memory 20
Figure 18. Bus Master to the Primary Surface 21
Figure 19. Bus Master to Secondary Surface 23
Figure 20. Camera Configurator Startup Screen 25
Figure 21. PLL and Crystal Modes 26
Figure 22. Camera Configurator Startup Screen 27
Figure 23. Select Video Timebase for Channel 0 Input 27
Figure 24. System Configuration 28
Figure 25. Saving a File 29
Figure 26. Cable Configuration 29
Figure 27. Select Trigger Source 30
Figure 28. Setting Trigger Parameters 30
Figure 29. CV-M10 and Frame Reset 31
Figure 30. CV-M10 Switches 31
Figure 31. CV-M10 Configuration File 31
Figure 32. Frame Reset Parameters 32
Figure 33. Timing Supplied by Camera 32
Figure 34. Variable Scan Timing Parameters 33
Figure 35. Dual-Channel Configuration 33
Figure 36. Output Fields Toggle Between Video Channels 34
Figure 37. Configure Slave Channel 34
Figure 38. Acquire and Display an Image 35
Figure 39. Create a New Project 39
Figure 40. Select Type of Application 40
Figure 41. Image Transfer Paths 41
Figure 42. Choose Image to Display 42
Figure 43.Connect Interrupts to Application Code 43
Figure 44. Select Overlay Type 44
Figure 45. Check ITEX Include Directory 45
Figure 46. Check ITEX Library Directory 45
Figure 47. IC-ASYNC AppWizard Application 46
Figure 48. VISION1 Classes and Source Files 47
Figure 49. Unlocked versus Locked Image Frames 52
Figure 50. Four Camera Setup 56
Figure 51. Camera Configuration Switches 57
Figure 52. Camera Cable Configuration 58

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 53. Four Channel Acquisition and Display 59
Figure A-1. IC-ASYNC Board and Connectors 1
Figure A-2. Cameras and Cables 6
Figure A-3. BCBL-PCV1 7
Figure A-4. BCBL-PCV2 7
Figure A-5. ACBL-PCVM1 9
Figure A-6. ACBL-PCVM2 9
Figure A-7. ACBL-PCVM3 9
Figure A-8. ACBL-BNC Adaptor Cable 11
Figure A-9. ACBL-HIR1 Adaptor Cable 11
Figure A-10. ACBL-HIR2 Adaptor Cable 12
Figure A-11. CBVM10-6 Adaptor Cable 13
Figure A-12. ACBL–VSCAN Adaptor Cable 14
Figure A-13. Dual-Tap Camera Adapter Cables 15

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Introduction

Welcome to the IC-ASYNC User's Guide

The purpose of this manual is to help you quickly get started using IC-ASYNC in your applications. This guide explores how to use key hardware features and software tools that make IC-ASYNC one of the most powerful, flexible, and highest performing frame grabbers for machine vision applications. After installing your hardware and software, please take time to read the *Software Release Notes* and *readme* files for the latest information.

Introduction and Overview

In the field of high performance machine vision, IC-ASYNC is the first single machine vision board capable of performing the work of four acquisition boards. IC-ASYNC represents breakthrough technology that, for the first time, integrates two or four independent, asynchronous frame grabbers on one full size PCI-bus card. With the ability to acquire up to four camera channels asynchronously, IC-ASYNC can handle high speed applications that traditionally could only be solved using expensive and complicated vision processors. IC-ASYNC is designed for multi-point inspection operations, and is ideal for OEMs building custom, high-speed machinery for consumer packaging, glass/bottle/metal can production, and pharmaceutical and semiconductor/electronics manufacturing.

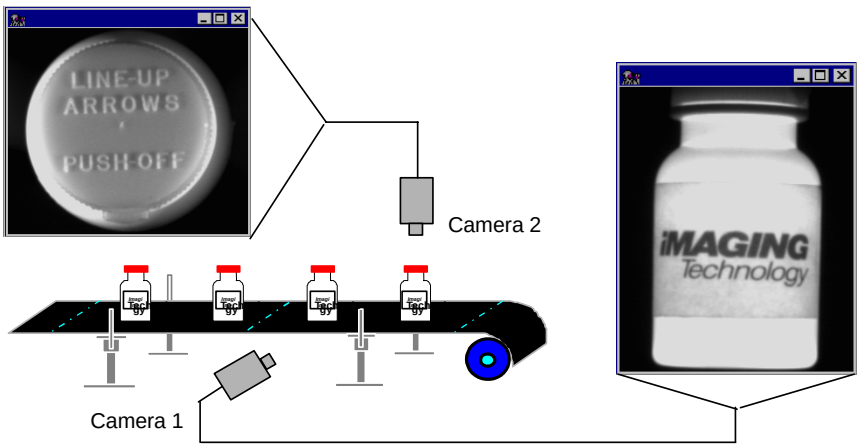


Figure 1. Single Production Line with Multiple Inspections

Multi-point inspection (Figure 1) has always been a challenging application for machine vision. By supporting four asynchronous inputs, all sides of a part can be included using a single acquisition system; an important feature in 100% qualitative inspections. Prior to IC-ASYNC, each camera inspection point required a separate frame grabber, doubling or even tripling the cost of the machine vision system and significantly increasing the programming effort.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IC-ASYNC meets application speed requirements using a high-performance, bus master controller that satisfies demanding bandwidth requirements using the standard PCI-bus. Each of the input channels on IC-ASYNC can acquire images independently and asynchronously. Every channel has its own fully programmable camera interface, digitizer, and 1MB image memory. The two channel version supports one dual-tap camera and the four channel version supports two independent dual-tap cameras.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Installation and Setup

This section describes how to quickly install the IC-ASYNC board and ITEX software, connect a camera and start acquiring images.

NOTE *Please install the ITEX software before installing the IC-ASYNC board.*

System Requirements for using IC-ASYNC

- PCI-bus Computer with Pentium processor or equivalent,
- Windows 95, Windows 98 or Windows NT 4.0 and at least 32MB
- An open full size 32-bit PCI-bus expansion slot

To support hardware overlays, the computer video card should have one of the following chipsets: S3 Virge, Intel 740 or ATI 3D RAGE / RAGE PRO.

Software Installation

IMPORTANT *Please read before installing the ITEX SDK:*

- *Under Windows 95 or 98, the Plug-and-Play Configuration Manager detects new hardware after installing the IC-ASYNC. If you're prompted for a "manufacturer's CD or diskette" to install the ITEX device driver,place the CD-ROM in the appropriate drive and follow the instructions for installing the driver. The driver is located in the root directory on the CD. Windows NT should boot normally.*
- *Uninstall any previous versions of ITEX. To do this, run the Windows "Add/Remove Program Utility" located in the Control Panel folder on your system. The Control Panel folder can be found by selecting:* "Start - Settings - Control Panel" from the Windows task bar.
- *Installing ITEX SDK under Windows NT requires Administrative Privileges.*
- *Make sure that the Microsoft C/C++ or Visual Basic compiler, version 5.0 or later, has already been installed so that ITEX extension help will be installed and enabled properly.*

1. Before running the Setup program, make sure all other Windows applications have been closed. To install the ITEX SDK software, place the CD in the appropriate drive to start the installation automatically. If Auto-Start is not enabled, manually start the install program by selecting "Run" from the Windows Start Menu and type:

   *<CD drive letter>***:\SETUP.EXE**
2. Click-on "Install Software".
3. Select Operating System and ITEX software for development. Click-on "Start Install", then "Yes" to continue.
4. At the Welcome screen, click "Next". Please take a moment to read the Software License Agreement, click "Yes" to continue setup program.
5. Enter User Information, then click "Next" to continue to the Destination screen. Choose a location on the local system drive to put ITEX SDK. Note: This destination is referred to as the *"ITEXroot"* directory. Click "Next" to continue.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6. Select Installation Type. "Typical" is recommended and will install all files. To install software on a Run-time system (loads only the DLLs, Device Driver, Diagnostics and Camera Configurator), select "Compact". Click "Next" to continue.

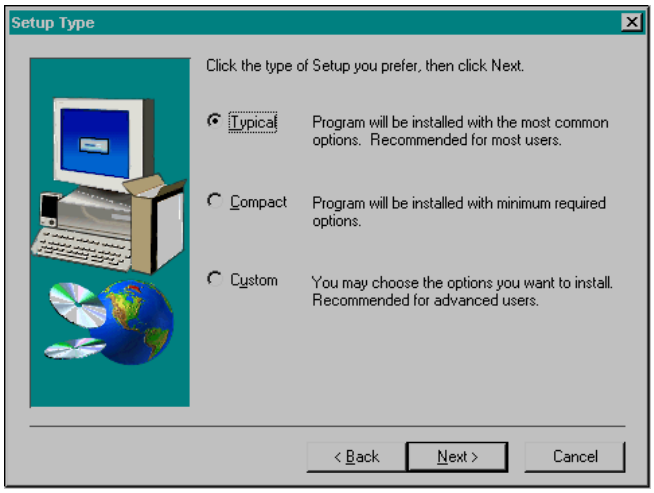


Figure 2. Setup Type

7. Select the product documentation and examples to install, and click "Next".
8. Enter a Program Folder name, and click "Next".
9. You have the option of installing the Acrobat Reader (for viewing and printing documentation supplied in PDF format), choose "Yes" or "No".
   For Windows 95 users, DirectX Media Components are installed at this point. Setup is done. Click "Finish" to restart your computer.

Environment Changes

The following additions have been made to your system:

- Windows 95 / 98 driver location: c:\windows\system\ **itipci.vxd**
- Windows NT driver location: c:\winnt\system32\drivers\ **mvcntp.sys**
- The following ITEX configuration file path has been added to your computer environment: **CNF15040=C:**<*ITEXROOT*>**\config**
  For Windows 95 and 98, the configuration path has been added to the Autoexec.bat file. Under Windows NT, it's been added to System Properties (right-click on "My Computer", select "Properties" then "Environment").
- The following ITEX Program Folder now exists which contains all the example programs, ITEX utilities, documentation, and release notes:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

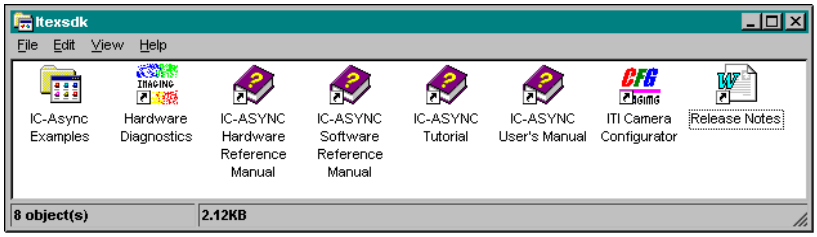


Figure 3. ITEX Program Group

Hardware Installation

Before installing the board, turn off power to the computer and peripherals. Discharge static electricity from your body by touching a metal part of the computer chassis.

1. Remove the cover from the computer and check for an empty, full length PCI slot. Slide the IC-ASYNC board carefully into the slot and ensure the front metal plate slips properly into the slot's filler bracket. Make sure card is nestled securely in the slot and not touching anything else inside the computer. If you have the I/O ribbon cable, attach it to the 26-pin header connector. Place the cover back on the computer, but *do not* connect power.

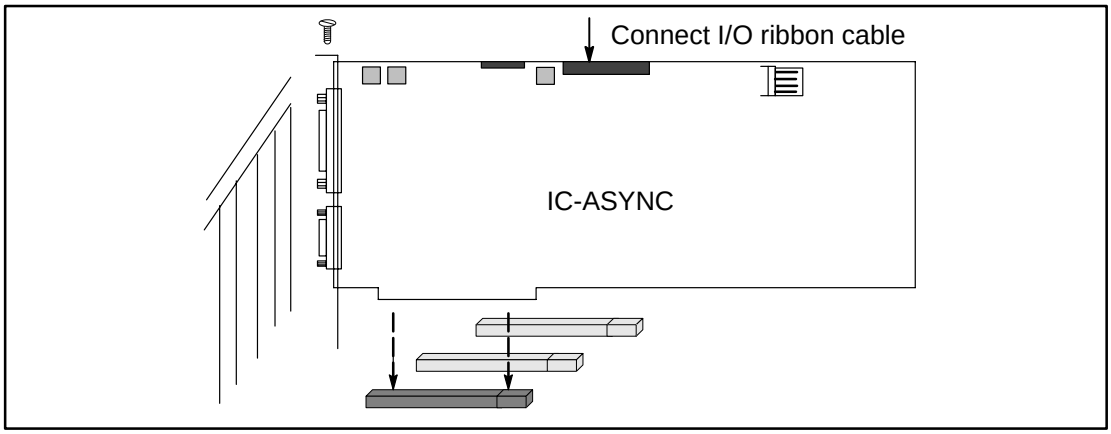


Figure 4. Install IC-ASYNC in Computer

2. Connect all other peripherals to your computer. *Do not connect the camera and cable yet.* Power up and boot your system. Connections to other equipment, sensors, strobes, and other I/O devices are discussed later in this manual.

Diagnostics Utility

After installing the IC-ASYNC board the first time, it's a good idea to run a sanity check on the hardware. This program will auto-detect the number of boards you have plugged into the system, initializes them, then performs some functional and memory tests.

NOTE *Before running this tool, shut down any other Windows applications that are running to prevent "timeouts" from occurring causing the program to generate false failures.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NOTE *Run the diagnostic test with no camera connected. The program tests the internal timebase, and a camera that supplies timing could cause false failures.*
Do not connect or disconnect cameras with the computer power on. Because the connectors provide 12 Volt power, the camera or IC-ASYNC could be damaged by "hot plugging".

1. Double-click on the "Hardware Diagnostics" Icon to start the test.



2. The program will report back any hardware failures it finds for each module, in a ".DLG" file called "DIAG" (located in C:\<*ITEXROOT*>\bin directory). If there is a hardware failure, contact your local field applications engineer to verify the problem, or contact Customer Support at Coreco Imaging, Inc.

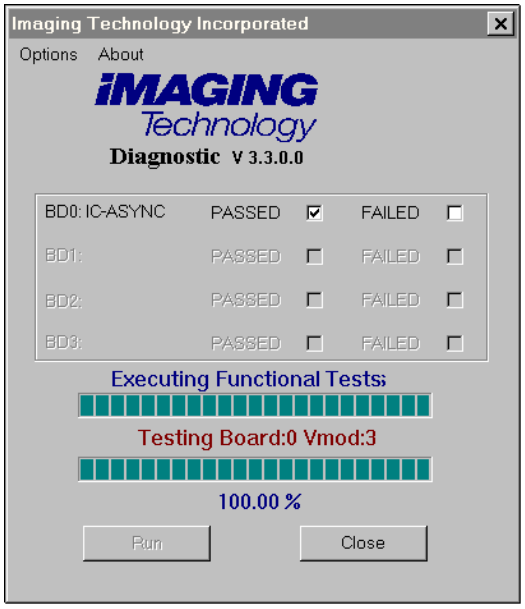


Figure 5. Diagnostic Program Report

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Connect a Camera and Acquire an Image

You are now ready to connect and configure a camera, and acquire an image. First, shut down the computer.

1. Connect your video source to the CAM0 input of the IC-ASYNC. If only connecting a single camera, follow the diagram in Figure 3, showing the camera connected directly to the IC-ASYNC using the adapter cable supplied with the system.

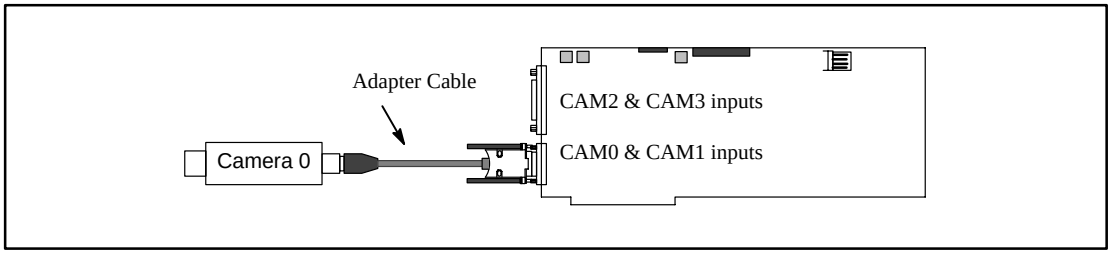


Figure 6. Single Camera Connection

2. If connecting multiple cameras, follow the diagram in Figure 4, showing two cameras connected to the IC-ASYNC using the adapter and breakout cables supplied with the system.

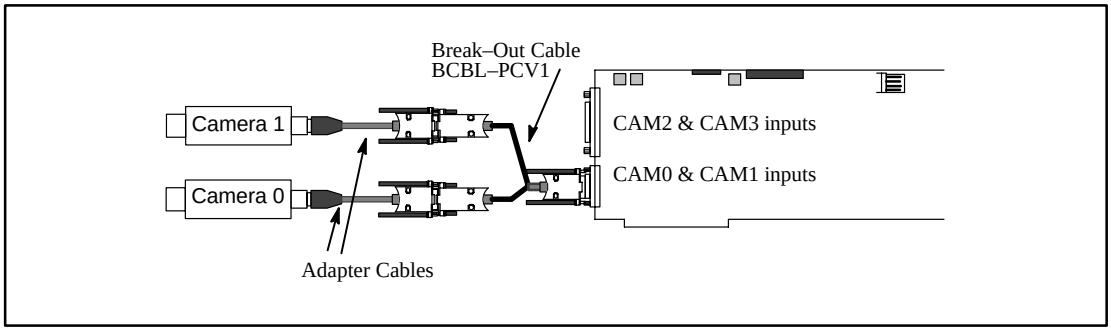


Figure 7. Multiple Camera Connection

NOTE *Coreco Imaging, Inc. provides an assortment of adapter cables depending on your camera requirements. These cables can supply +12VDC @ 0.5 Amps from chassis power. Please refer to the Appendix in the back of this manual for more information regarding these cables and cable configurations.*

3. Power on and boot your system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ITEX Configurator Utility

The Camera Configurator® utility makes it easy to set up and pre-configure each video input port to its associated camera device. Later in this manual there's more discussion on using this tool, but for now we're just using it to re-initialize IC-ASYNC board and acquire an image. At this point, your camera or video source should be connected to the IC-ASYNC frame grabber (refer to hardware install section if not) and ready to acquire a picture.

1. Double-click on the "Configurator" Icon. The utility will take a moment to detect and initialize the hardware.



2. Now you should see the screen shown in Figure 8. Right-click inside the display window and select "edit".

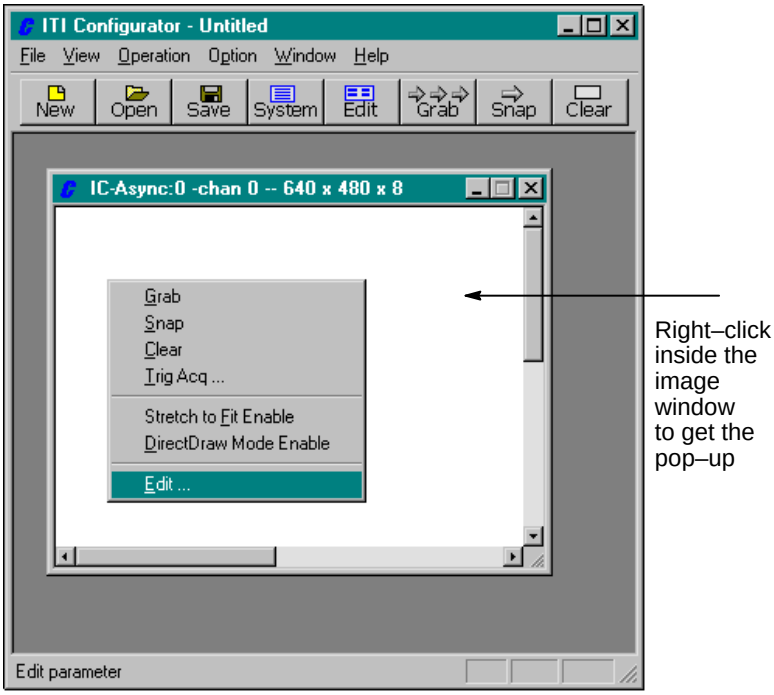


Figure 8. Configurator Start Up Screen

3. From the "Camera Selection" list (Figure 9), select a video timebase that best suits your camera. The default timebase is configured for an RS170 camera in PLL mode with a resolution of 640 H by 480 V. But for example, if you are connecting a CCIR camera and wanted a resolution of 768 H by 574 V, pick "CCIR 768x574XTAL" as shown in Figure 9.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

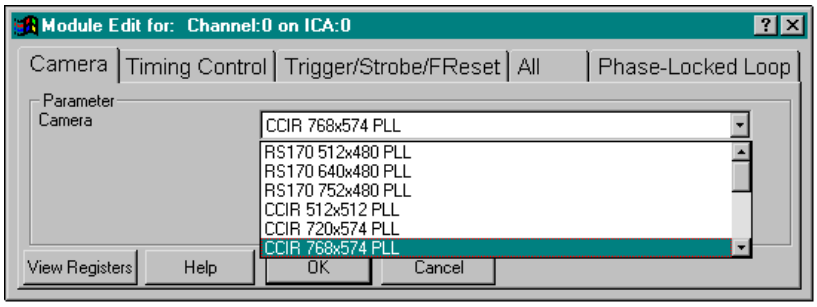


Figure 9. Camera Selection

NOTE *XTAL stands for "Crystal" mode and PLL stands for "Phase-Locked Loop". In Crystal mode, timing for the camera is supplied by IC-ASYNC. In Phase-Locked Loop (PLL) mode, timing is supplied by the camera to IC-ASYNC.*

4. Now to acquire a picture. Make sure the camera is pointed at an object with suitable lighting. In the the Configurator toolbar, click on "Grab". You should see a live image in the display window.

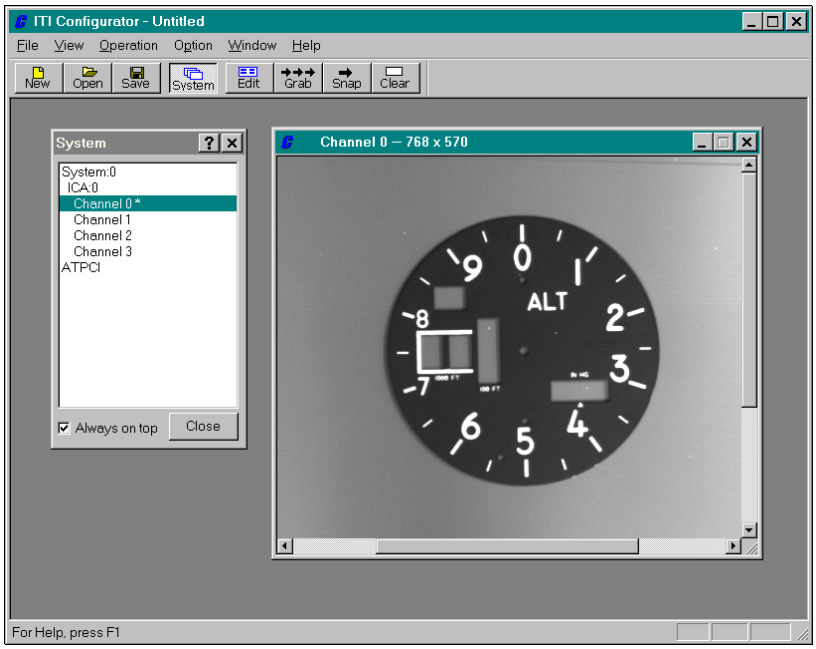


Figure 10. Live Image Acquire from Camera Configurator

Click on the "System" button to show all four channels. If you have cameras connected to these other channels, go ahead and configure them the same way by double-clicking on the specific channel.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Principles of Operation

- IC-ASYNC is an image capture board that acquires monochrome images simultaneously from up to 4 cameras. The cameras do not need to be synchronized (or gen-locked) and do not need to be using the same format. IC-ASYNC interfaces with standard (RS-170 and CCIR, VGA progressive scan) and non-standard "variable-scan" cameras. Variable scan cameras must provide TTL level timing signals for pixel clock, line timing, and frame timing.
- IC-ASYNC provides fully programmable timing and a variety of trigger, strobe, I/O, and asynchronous reset options that are ideal for practical machine vision applications.
- IC-ASYNC architecture resembles 2 or 4 independent frame grabbers ("input channels") connected by a common Bus Master Controller (BMC) that interfaces to the PCI-bus. Each of these input channels has a fully programmable camera interface and 1MB of memory. Figure 11 is a block diagram of IC-ASYNC.
- The PCI-bus interface incorporates a hardware "scatter gather" table for highly efficient, fully automated and simultaneous image transfers from the 2 or 4 image buffers ("image frames").
- IC-ASYNC uses the high data transfer rates of the PCI-bus to eliminate the need for on-board processing or display circuitry. Image display and processing is handled by the host computer resources. The linear format image memory allows acquisition of a variety of image sizes. The 1MB per input image memory behaves as a temporary buffer between the camera interface and the host PCI-bus system.
- IC-ASYNC is capable of bus mastering image data directly to a destination memory within the PCI-bus system, such as system memory or VGA memory. Transfer rates in excess of 100MB/s can be sustained, depending upon the host capabilities. Images can be transferred to host memory in a fraction of the time that they were acquired. By minimizing the PCI-bus transfer time and CPU overhead, more bandwidth is available for processing or other system resources.
- A new display feature (Multi-YCrCb Mode) for IC-ASYNC supports multiple video images in the VGA display board secondary (YCrCb) surface. Not all VGA boards have a hardware secondary surface ("streams processing display"). Those VGA boards that do incorporate hardware for the secondary surface normally support only one video image. The ITEX SDK library incorporates a proprietary display mechanism that supports multiple live video images in the single hardware-implemented secondary surface. Each of these live video images can be used with non-destructive hardware overlay graphics.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

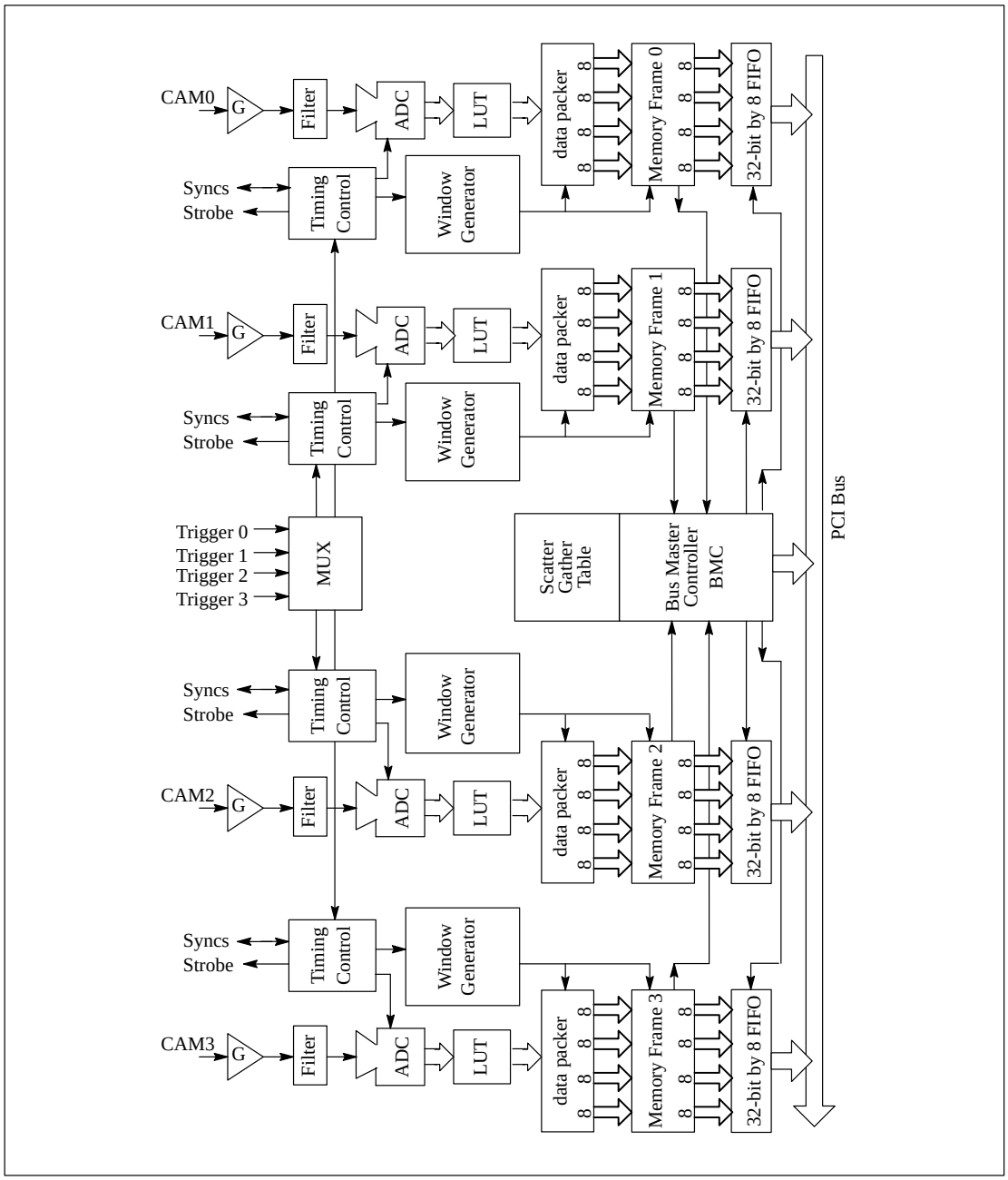


Figure 11. IC-ASYNC Block Diagram

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FEATURES

Sensor Interface

- Two or Four independent analog video inputs, AC coupled and terminated to 75 Ohms. One or two dual-tap asynchronous camera inputs
- Monochrome video sources: "Standard" (RS-170, CCIR), VGA and progressive scan cameras (composite video or driven with separate or external sync signals); Variable Scan ("area scan" cameras providing TTL timing for Line, Frame, and clock)
- Programmable Timebase Generator and pixel clock; programmable resolution up to 1024 by 1024 interlaced or non-interlaced. Outputs horizontal, vertical and frame reset timing to camera
- Programmable Phase-Locked Loop (PLL) for locking to camera timing

Video Digitizer

- 20 MHz Monotonic 8-bit flash ADC; input pixel rates to 20 MHz

Video Memory

- 1MB linear mapped VRAM per input/camera channel

On Board Digital I/O

- 16 bits; 8-bits in 8-bits out
- Input latching capable

Video Signal Conditioning

- Input Gain software selectable 1.0, 1.2, 1.3, 1.5
- Low-Pass Filter (LPF): 6.5 MHz fixed (no bypass)
- Programmable positive ADC reference (full-scale) from 0 to +2 Volts in 64 steps
- Programmable negative ADC reference (zero) from 0 to +1.2 Volts in 64 steps
- DC Restoration clamp pulse, automatic or programmable
- Look-Up Table (LUT) following ADC - 8-bit-in 8-bit-out

External Trigger, Strobe Control and Frame Reset

- Four Trigger Inputs, independently selectable per camera; synchronizes acquisition to external events. Minimum input pulse width 50 ns
- Frame Reset Mode: external trigger initiates camera frame reset, strobe, and image acquisition for immediate capture of moving objects; programmable delay 1 to 511 lines
- Trigger on Frame Mode: strobing and image acquisition are synchronized with the video signal
- Four Strobe Outputs, one per camera; with programmable polarity; 120 ms nominal duration, programmable delay 0 to 63 lines

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Video Window Generator

- Allows selection of video window within video signal
- Horizontal offset programmable 0 to 1023 pixels (one pixel increments)
- Horizontal size programmable 4 to 1024 pixels (four pixel increments)
- Vertical offset programmable 0 to 1023 lines (one line increments)
- Vertical size programmable 1 to 1024 lines (one line increment)

Interrupts

- Host interrupt on occurrence of strobe, trigger, vertical blank, or end of acquisition

Hardware Color Dot Clipping

- Image data can be "clipped" during bus master to a Windows display device to eliminate any conflicts with Windows reserved colors

On Board Decimation

- Image reduction on acquisition by factors of 2, 4, or 8 independently in horizontal and vertical axis

Display - Windows

- Display Resolution as per installed video display card
- Non-destructive hardware enabled overlay support
- DirectX compatible display card (SVGA) required for real time display
- Multiple YCrCb Image Display; proprietary display mechanism supports multiple video images from the VGA board's secondary surface

Automatic Double Buffered Full Frame Acquisition

- Hardware controlled "ping-pong" acquisition into image memory, provides most efficient acquisition and bus master synchronization possible

VIDEO INPUT

IC-ASYNC has 2 or 4 camera input channels. Each channel has its own independent Programmable Timing Generator (PTG), Phase-Lock Loop (PLL), sync inputs and strobe outputs, and 4 selectable trigger inputs. All 4 channels can be active together with all 4 triggers and strobes. The trigger inputs and strobe outputs are on separate connectors from the camera inputs (Appendix A gives information on connector pin-outs and cables).

IC-ASYNC interfaces with broadcast standard RS-170 and CCIR cameras, as well as VGA progressive scan cameras. A phase-locked loop (PLL) provides each channel with the capability to synchronize to RS-170 and CCIR devices that provide composite video or separate TTL sync signals. Alternatively, a programmable timing generator (PTG) can supply the camera or video sensor with horizontal and vertical timing pulses.

"Variable scan" mode synchronizes to cameras that do not use broadcasting or VGA standards, the camera must supply line enable (LEN or HDRIVE), frame enable (FEN or VDRIVE), and pixel clock (VSCLOCK

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

or PCLK). Variable scan timing signals must be positive voltage, single-ended TTL level (active high or active low).

Triggering circuitry allows "part-in-place" and other external events to initiate image acquisition. Software enables or disables the trigger, selects input polarity, and selects external or software trigger. Multiple triggering modes allow flexibility in synchronizing to external events. Each channel can select any one of four trigger inputs. The trigger inputs are available on the miscellaneous connector.

Strobe outputs allow IC-ASYNC to initiate or synchronize external events or equipment. Each channel has its own dedicated strobe output. The position (firing time) and polarity are programmable. The strobe outputs are available on the miscellaneous (MISC) connector.

A Programmable Window Generator (PWG) creates a "valid video window". Only data inside the valid video window gets stored in the image memory. Horizontal size must be a multiple of 8 pixels. The maximum window size (maximum image size) is 1024 H (pixels) by 1024 V (lines).

CAMERA INTERFACE

IC-ASYNC has 2 or 4 single-ended analog camera inputs. Each input channel has its own filtering, timing control, and memory. All inputs are independent and can be acquiring images simultaneously. All 4 channels are identical and the following information applies to all of them.

Each channel can interface with standard RS-170 and CCIR cameras and video sources, or non-interlaced (progressive scan VGA) cameras. Only area scan cameras are supported, not line scan. Timing can come directly from the camera (PLL mode) or from an on-board PTG (XTAL mode). The PTG. supplies composite or horizontal sync, vertical sync and frame reset signals to drive the camera. The same logic provides the capability to synchronize acquisition to external trigger events and generate strobe pulses or interrupts.

Video input (AC coupled) passes through a gain circuit with 4 software selectable gain factors: 1.0 (unity), 1.2, 1.3 and 1.5. To achieve the best performance, the video amplitude should fit the sampling range of the ADC (Analog to Digital Converter) therefore the voltage range of the video signal (black to white voltage) after gain should be between 0.8 and 1.5 volts. Following the gain stage, the video passes through a low-pass filter circuit with a -3dB roll-off frequency of 6.5 MHz. Standard RS-170 and CCIR video has useful frequency content up to approximately 6.5 MHz. Frequencies above this can be eliminated using the low pass filter. The video is then passed to the ADC for digitization. IC-ASYNC uses 20 MHz ADCs with 8-bit resolution.

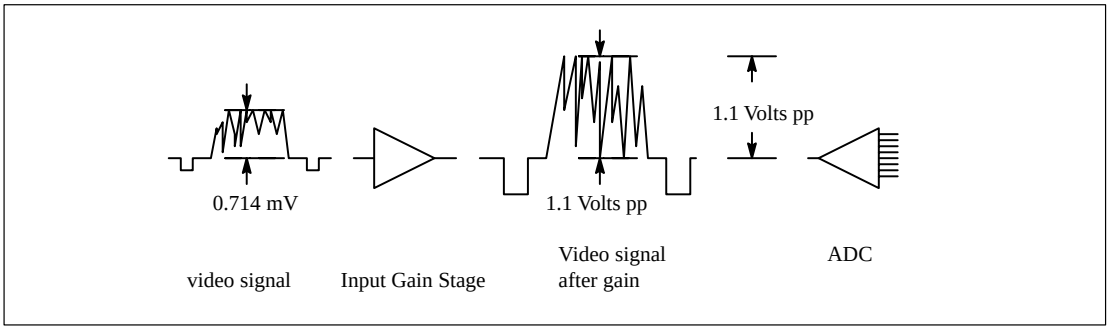


Figure 12. Input Gain Stage

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The digitized 8-bit video passes through a 256 by 8 bit LUT (Look-Up Table) before being loaded into the 1MB linear memory frame. This LUT performs point transformation on the data, without processing delay, before loading into the memory. Writing to the memory is enabled by a programmable window generator (PWG). The PWG provides the capability to crop the incoming image, so that only a specific AOI (Area-Of-Interest) is acquired into image memory. Timing to the PWG can be from the sync stripper (HSYNC and VSYNC stripped from composite video in PLL mode), the PTG (user programmed HSYNC and VSYNC driving the camera) or variable scan timing inputs.

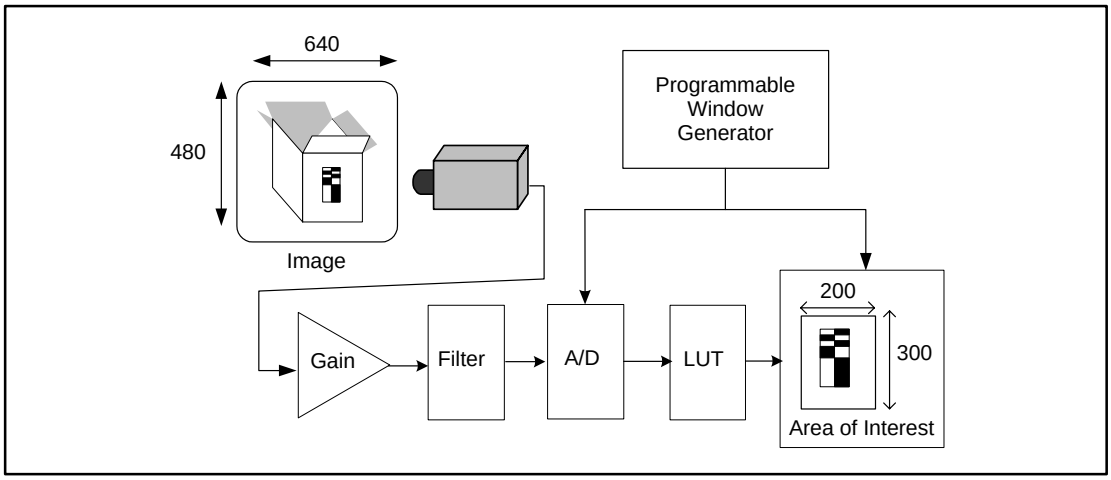


Figure 13. Using Programmable Window Generator to Capture an Area-of-Interest

IMAGE MEMORY

Each video input channel contains a 1MB linear memory frame. Each memory frame has independently programmable acquisition modes. The linear formatted memory frames provide more flexibility for incoming images. Any horizontal and vertical size up to 1024 by 1024 (limited by the PWG) is supported, provided the image is less than 1MB total. As mentioned, the PWG receives timing from the PLL, PTG, or variable scan input.

Each memory frame can be set to acquire multiple images. The hardware has a sequential mode that acquires and stores up to 8 sequential images. Multiple images can be stored, but the hardware does not have restrictions to protect one image from being overwritten by another in the memory. Memory addressing will wrap around at the 1MB boundary if the multiple images are greater than 1MB total.

Every memory frame can perform a 2X image decimation (sub-sampling), The total pixels on a horizontal line must occupy the image memory frame on 8-byte boundaries during acquisition. This restriction also applies to decimated images.

Acquisition

IC-ASYNC supports interlaced or non-interlaced camera data. Images can be any size within the 1MB area of memory and on 8-byte (8 pixel) boundaries. Two types of acquisition are available: Normal and External

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Trigger. Normal acquisitions into image memory are based only on camera frame timing. Externally triggered acquisitions are based on external trigger signals (and camera timing).

Normal acquisitions begin and end on the framing signal, FEN, out of the PWG Frame enable (FEN) is the digital version of the vertical blank signal to or from the camera. A Snap command performs a single frame acquire beginning on the next FEN rising edge. A Grab command performs continuous acquisition beginning on the next FEN rising edge. A Freeze command will stop a grab at the next falling edge of FEN.

When acquiring interlaced images, a Snap lasts for two fields (1 frame). When acquiring non-interlaced images a Snap lasts for one field (1 frame). Interlaced image acquire can be programmed to start on the even, odd, or next field.

External trigger allows image acquisition to be synchronized to external events (for example, a part-in-place sensor). When acquiring an image in external trigger mode, the acquisition does not start until a trigger signal is received. The trigger input remains armed after the acquisition completes, ready for the next trigger input to occur.

IC-ASYNC has the ability to acquire and store up to 8 images sequentially into image memory with the execution of one Snap command. A programmable a frame count (FCNT) sets the desired number of frames (1 to 8) per Snap command. For non-interlaced or interlaced acquisitions, FCNT always refers to full frames. This feature is required for setting up "ping-pong" buffers that allow real time (30 frames/s -RS-170 and 25 frames/s - CCIR) image acquisition and transfer to host memory.

Multiple Frame Acquisition In Normal acquisition mode, the frame count is set for 1 to 8 images. When a Snap command is executed, the programmed number of images is acquired sequentially into a memory frame. In Grab mode, the set number of images are acquired, then the acquire address wraps back to the beginning again and overwrites the previously acquired images. The image size must be taken into account when acquiring multiple images as the frame buffer is only 1 MB and wraps around if the number of incoming pixels fills the buffer. In External Trigger mode (Snap or Grab), each incoming trigger will acquire one frame and advance the frame count.

Memory Access

IC-ASYNC frame memory occupies 8MB in host PC memory address space. During power-up, the PCI-bus host system assigns a base address to this region. Each frame uses dual ported memory, one port supports image acquisition and target access, the second port supports Bus Master access. Both ports can be accessed simultaneously.

Host access to the memory is always on DWORD (32-bit) boundaries. Every access provides 4 bytes, two adjacent 16-bit pixels or four adjacent 8-bit pixels. BYTES or WORDS can only be accessed by reading or writing on 32-bit address boundaries and masking or disregarding the extra bytes within the DWORD.

EXTERNAL TRIGGER AND STROBE

IC-ASYNC can acquire images based on an external trigger input signal and it can output a synchronous strobe signal coincident with the trigger. IC-ASYNC also supports asynchronous frame reset outputs. A software trigger is also provided to simulate a trigger cycle.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The external trigger input is a programmable edge trigger with a minimum pulse width of 50 ns. The trigger signal resets the PTG and PWG to synchronize them to the incoming video. A strobe output (+5 Volts, one horizontal line in duration) occurs at the next vertical blank (SLOW STROBE MODE) or the strobe can be programmed to occur coincident with the trigger input (FAST STROBE MODE). An external FRAME RESET pulse based on an external trigger input can be programmed and driven to the camera. Frame reset immediately forces the camera to take a new picture.

There are four strobe outputs, one on each channel (fixed). Strobe 0 is always associated with channel 0 and strobe 3 with channel 3. There are four external trigger inputs. Each channel can select any one of four inputs. Each camera can have its own trigger input (when applications require multiple cameras to take pictures at different times), or all cameras can use the same trigger (when applications require multiple cameras to take pictures at the same time).

The trigger, once armed, will re-arm itself. This enables the application to operate in "GRAB" mode so that the software can monitor the interrupts for the end-of trigger cycle to know when the image is acquired. When enabled, the selected edge of the trigger input causes an image acquire cycle to start at the next vertical blank following the trigger.

There exists a trigger interrupt that can be monitored to see if there are more triggers than acquisitions. This indicates that missed trigger occurred, and can also measure the precise time from the trigger event to the acquisition completion. The interrupts generated for the trigger are: Start or Trigger (when trigger pulse comes in) and End of Trigger (when triggered acquisition completes).

NOTE *Trigger inputs and strobe outputs are not on the camera connectors. These signals are located on the 26-pin connector and MISC connector on the break-out cable (see Appendix A).*

OPTICAL ISOLATION

When connecting the IC-ASYNC trigger input to a typical industrial sensor, using an optical isolator is recommended since the trigger input is sensitive to noise (Figure 14).

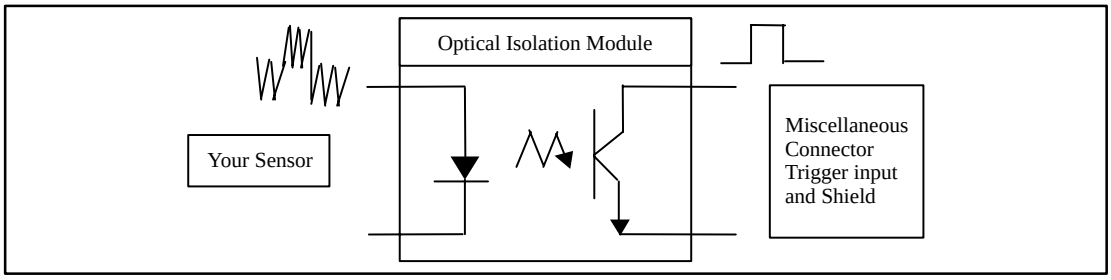


Figure 14. Optical Isolation

PARALLEL I/O

IC-ASYNC provides digital I/O capability for controlling or monitoring external events. The digital input and output lines, which are available on a 26-pin header at the top of the PC card, can be cabled to a 25-pin

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D-SUB connector that occupies an open slot in the PC chassis. The functionality of the I/O port is as follows:

IN(7-0): Eight digital input lines provide capability to read these as raw or latched (by STROBE_I) inputs.

OUT(7-0): Eight digital output lines driven by a programmable register.

STROBE_I: Input strobe signal can be used to latch the 8-bit input data (if this mode is selected). The polarity of STROBE_I is programmable.

STROBE_O: Output strobe is an output line driven by a 1-bit register.

I/O INT: An external input line that can be used to generate an interrupt.

CAMERA POWER

IC-ASYNC can supply +12 Volt power to the camera from the PCI-bus chassis, through the break-out cables and camera adapter cables. The supply is limited to 500 mA per camera, and 2.0 Amperes; 1 Ampere for each connector. This +12 Volt output is available on all four camera connectors. This internal power supply is limited by two resettable fuses shown in Figure 16. Exceeding the 1 Ampere rating on each connector causes one of the resettable fuses to open. Power down the chassis to reset the fuse.

IC-ASYNC can be modified at the factory to support connecting the PC power supply through the resettable fuses to the camera connectors.

IC-ASYNC supplies +5 Volt power to the parallel I/O connector from the PCI-bus chassis. This supply is limited to 500 mA through a resettable fuse.

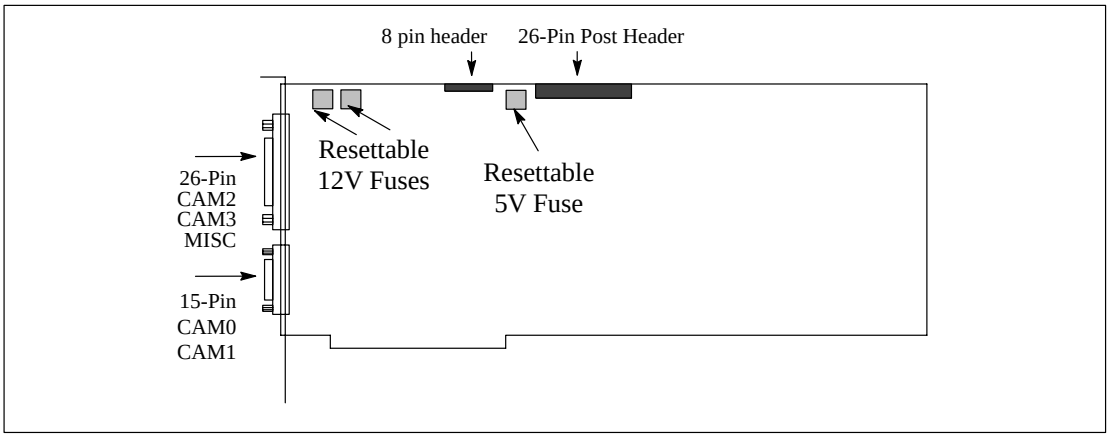


Figure 15. Fuse Location

PCI INTERFACE

The PCI interface allows IC-ASYNC to operate as both PCI-bus master and target (slave). Only DWORD (32-bit) access is supported. IC-ASYNC is capable of bus mastering data from image memory directly to a destination within the system (host CPU memory or another PCI target, such as VGA display memory). The interface supports target access to the registers and image memory. PCI-bus interrupts may be gener-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ated based on events occurring on IC-ASYNC. Configuration registers required by the PCI-specification allow the board to be recognized on power-up for automatic system configuration ("plug-and-play").

Bus Master Operation and Scatter Gather

IC-ASYNC contains a sophisticated bus master controller (BMC) with hardware scatter gather table. The host CPU is free to perform processing and other tasks during bus master transfers rather than controlling the image transfer operation. The BMC has multiple frame ping-pong source and destination capabilities which allow the application to program IC-ASYNC to establish the needed amount of processing time.

When a software application requests a host destination buffer, typically a group of small segmented pages are given rather than a large continuous block of system memory. The "small pages" are where image data is scattered throughout the host memory, like a fragmented file. The host keeps track of these "small pages" to gather the data during processing, much like the file record allows processing fragmented files.

To transfer the source image to a segmented buffer in system memory, multiple "small page" transfers must be executed which normally requires the host to poll throughout the transfer operation. The host sets up each page transfer, polls for completion, and executes the next page transfer until the entire segment is complete. The hardware scatter gather table eliminates the need for polling, and speeds up the transfer process. The scatter gather table is also used to transfer images in an X, Y format directly to VGA memory for display, or in dual-tap mode it can re-interlace separated image fields. Application software can disable the scatter gather table and perform single bus master transfers, which is the mode used on the IC-PCI and PCVision.

All data is stored as non-interlaced images, meaning entire images are sequentially stored in linear frame memory. With "de-interlace" enabled, interlaced video is de-interlaced on input; the two fields are woven back together. With every pixel in a sequential memory location, fast transfer rates are easily achievable. IC-ASYNC transfers complete images to the host CPU memory. All of these operations (bus master transfers, scatter gathering, de-interlacing re-interlacing, etc. are handled internally by the hardware and ITEX, therefore transparent to the application programmer).

Clipping and Padding

IC-ASYNC provides a clipping feature which forces pixels with values between 0x0 and 0xF to the value 0x10 and forces pixels with values between 0xF0 and 0xFF to the value 0xEF. This clipping feature allows data to be bus mastered to a VGA target (primary surface) without seeing "color dots" (MS Windows reserved values). Clipping can be disabled to allow true pixel values to be sent directly to a PCI destination. Each memory frame has its own clipping enable bit. Clipping is a function of the bus master interface, consequently, it does not affect pixels in image memory. As a result, the same image could transferred twice; to VGA (for display) with clipping enabled, and to host CPU memory (for processing) with clipping disabled. Because the bus master transfer speed is 4 to 5 times faster than acquisition speed, this method is not impractical.

IC-ASYNC has the capability to "pad" image data to 16-bit values during the bus master transfer. With pad mode enabled, a pixel value of 0x80 is added to each 8-bit pixel in the bus master FIFO. Each 32-bit bus master transfer contains two 16-bit pixels. The padded values are useful for display from the 16-bit Win-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

dows secondary surface (also referred to as the YCrCb surface, or in ITEX, the YCRCBMONO display sink) on VGA boards that support the secondary surface. Padding is the function of the bus master interface and does not affect pixels in image memory.

Clipping and padding should not be enabled at the same time. Clipping is used for transfer to the primary display surface and padding is used for transfer to the secondary display surface. The two features should never be used together.

VIDEO OVERLAY

Overlay is graphical data, possibly alphanumeric or other symbology, that appears on top of video data on a video display. There are several different techniques for achieving an overlay on video data acquired on Coreco Imaging, Inc's image capture products. All techniques described are for "video in a window" on a single monitor driven by the system VGA, as the image capture products have no on-board display. The techniques, up until recently, have provided a destructive overlay, which has disadvantages. These techniques will be discussed for completeness. However, there are now techniques which can provide a non-destructive overlay, which is preferable.

Destructive Overlay

There are two techniques for creating destructive overlays. "Destructive" means that video information and the overlay information reside in the same area of memory. The overlay information is written directly on the video information "destroying" it. The drawback with destructive overlay is, if you want to view live video at 30 fps (frames per second), 25 fps, or any speed other than static, the video information is also overwriting the overlay information and destroying it.

In the two destructive overlay techniques discussed, there are no inherent limitations on Acquisition Modules that can be used on the image capture products.

Destructive Mix in VGA Memory

The first technique for implementing destructive overlay is to use Direct Draw to transfer video information to the VGA card. The host then writes graphical data on the video data on the VGA card to achieve the overlay. The graphical data has to be constantly rewritten by the host because new video data is being written over it, usually at 25 or 30 fps. This requires a certain amount of host involvement which could otherwise be used for more useful purposes. Another disadvantage is that the overlay flickers on the display. The video is being written at 30 Hz, the host is writing the overlay however often it has the time to, and the VGA is running at its own speed, probably 72 Hz. This inherent lack of synchronization causes flicker.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

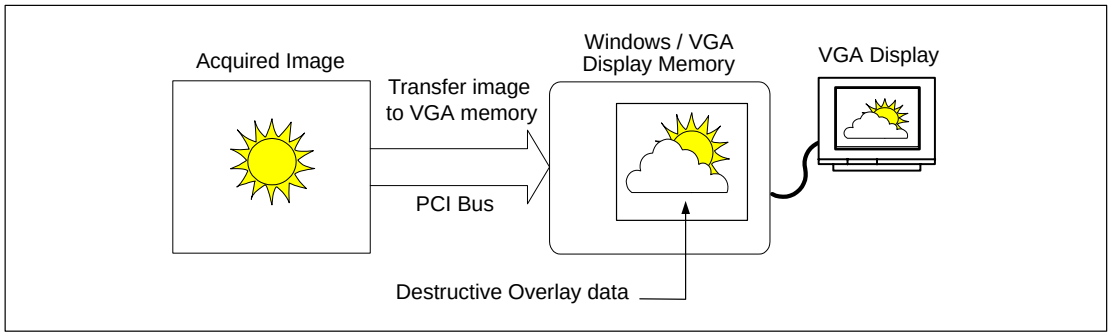


Figure 16. Destructive Mix in VGA Memory

Destructive Mix in Host Memory

The second destructive overlay technique is to use DIB (Device Independent Bit-maps) to transfer video information to the VGA card. A DIB transfer starts by performing a bus master transfer to host memory. The host then writes graphical overlay data on the video data, and transfers the entire frame to the VGA card. This method removes flicker, but requires much more host involvement in transferring and formatting the video data.

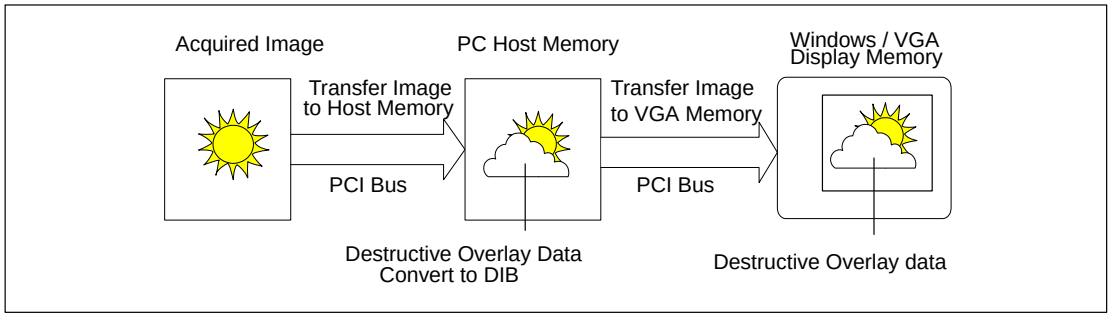


Figure 17. Destructive Mix in Host Memory

Non-Destructive Overlay

The non-destructive techniques use a feature of certain newer VGA chips called "streams processing." This feature allows a VGA chip to parse the memory on the VGA card into two parts called the "primary surface" and the "secondary surface." These surfaces are individually addressable. These two surfaces can be mixed together by the VGA chip for display.

The secondary surface can only be defined as a 16 bit surface. There are several different 16-bit formats to choose from, but 16 bit is the only choice. We use two different 16-bit formats in the two non-destructive overlay techniques. The first is RGB (5,5,5,1); there are five bits each of red, green, and blue information with an extra bit which serves as a kind of mask. If that bit is one, then that pixel will be mixed in for overlay. If it is zero, it will not be. This format is inherently poor at displaying monochrome data since the effective bit depth of an image is 5 bits or 64 gray-scale values. It is, however, fine for displaying color as you have a palette of 215 colors. The second format is YCrCb (4:2:2). In this format "Y" is the luminance (mono-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

chrome) component and the "Cr" and "Cb" are color components. The term 4:2:2 is a representation of how much of each kind of data (Y, Cr, and Cb respectively) is present in this format. In each 16-bit pixel there are 8 bits of luminance data and 8 bits of color data. An unsophisticated compression is achieved by having adjacent pixels share the same chroma information. Therefore the data effectively looks like: Y0Cr0, Y1Cb0, Y2Cr2, Y3Cb2. All of the luminance data is retained while some of the color data is averaged or shared. This loss of color data actually has little effect on the image as viewed, and therefore is a good format for displaying data. This is a very popular format in graphics and multimedia.

Bus Master to the Primary Surface

The first non-destructive technique is to use Direct Draw to provide video data to the primary surface and put the overlay in the secondary surface. In this method, the secondary surface must be a 16 bit RGB surface. The Direct Draw of the video data to the primary surface is a familiar technique to achieve a live display. Now, however, it is possible to have overlay data present in the secondary surface and have the two surfaces mixed together by the VGA hardware for display. There is no speed penalty for performing this hardware mix. The overlay data in the secondary surface is never overwritten, therefore it does not have to be constantly updated by the host.

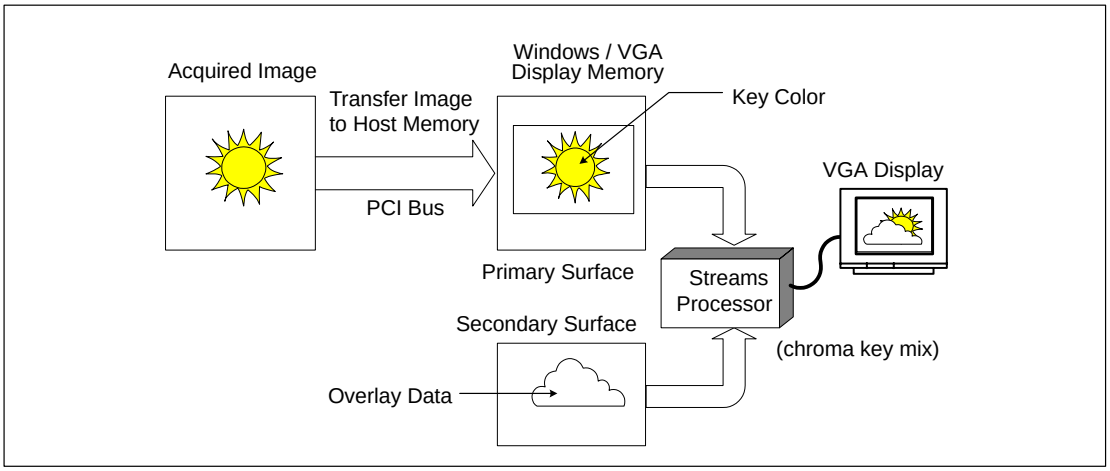


Figure 18. Bus Master to the Primary Surface

Limitations

There are some limitations with this technique. First, there are the normal restrictions incurred with Direct Draw of video data to the VGA card: VGA bit-depth matching and "color dots." As always, we have to match the pixel bit-depth of the output of the frame grabber with the VGA display. If 8-bit monochrome data is being acquired, the VGA card primary surface must be set at eight bits to bus master to it. This equates to a computer display of only 256 colors to be shared by Windows and all of the grayscale values of your video (which ideally would be 256 just by themselves). Windows will demand a minimum of 20 colors (usually numbers 0-9 and 246-255 in the palette). If video data is bus mastered to the VGA memory with those values reserved for Windows colors, the pixels with those values will take on these colors instead of the grayscale values that were intended. We call these "color dots" and they can be distracting or unaccept-

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

able. The technique used to remove color dots is to clip the data with a Look-Up Table before it is transferred. This leaves only 236 colors left for grayscale video which reduces the contrast of the video somewhat. It also may be said that limiting Windows to its minimum 20 colors makes for an unattractive Windows display. If 24-bit color data is being acquired, then the VGA primary surface must be set at 24 bits. This presents fewer "color dot" problems. The color video data can be clipped by a look-up-table to leave room for the 20 Windows colors (and more if desired) with negligible degradation to the video quality (20 colors out of 224). There is a restriction on the overlay as well. The overlay is in the secondary surface of the VGA card and is not being managed by Windows. Windows keeps track of all of the objects in the primary surface so that they all remain "good Windows neighbors." Pull down menus, dialog boxes, tool bars, main windows, video in a window (bus mastered data) etc., are all managed so that the proper components are on top, in the background, in the foreground, and displaying properly. Windows does not keep track of the secondary surface where the overlay data resides, a problem arises if any Windows graphics such as a dialog box intrudes on the same area of the screen as the overlay. The overlay will be turned off by the VGA card if the graphics encroaches on its area, even if just by a few pixels. This can cause problems with the application's GUI.

Bus Master to the Secondary Surface

The second non-destructive technique is to bus master video data to the secondary surface and put the overlay in the primary surface. In this method the secondary surface must be a 16-bit YCrCb 4:2:2 surface. The frame grabber must have 16-bit YCrCb data to bus master to this surface. Coreco Imaging, Inc's image capture products can provide data in this format. This 16-bit data can be either color data from an NTSC or PAL camera, or monochrome data from an RS-170 or CCIR camera.

The overlay is accomplished by mixing in video data from the secondary surface into the video window of the application GUI in the primary surface using a chroma key. A specific chroma value is designated as the hardware chroma key and wherever this value appears in the primary surface, the video data from the secondary surface is substituted for it in the display. The application GUI, overlay, and all windows graphics reside in the primary surface and the video is mixed in wherever it is needed by placing the chroma value. There are no problems with "color dots" using this technique. There are also no problems with the overlay being "turned off" as in the previous technique since Windows is managing all of the graphics while video is just being quietly mixed in by the chroma key. The primary surface can be run at whatever bit-depth is best for the application.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

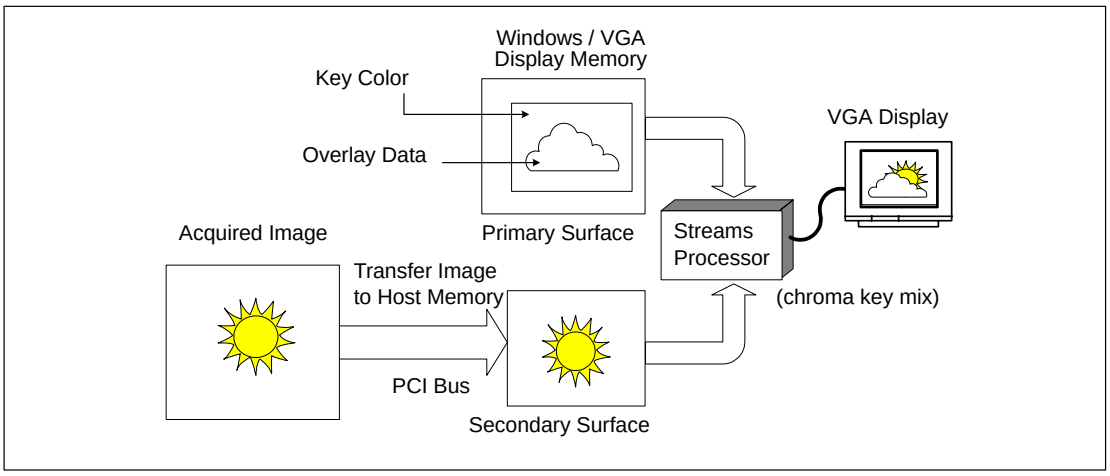


Figure 19. Bus Master to Secondary Surface

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Tutorials and Program Examples

In this section, each program example is explained in detail. Code segments are extracted to specifically highlight important points. The discussion here is limited to code pertaining only to ITEX, however, the user should open the Workspace (.DSW) file in Microsoft DevStudio to examine the complete example. Below is a listing of each example, what each one illustrates and what program source files contain ITEX code. All these examples are provided with the release and are located in: C:\<*ITEXROOT*>\examples

NOTE *Discussion in this section is limited to programming IC-ASYNC. Only source files which contain ITEX code are listed. It is assumed that the user has a working knowledge and familiarity with Windows NT or Windows 98, C/C++ using MFC, and Visual Basic.*

Example	*Description*	*Source Files*
1 Camera Configurator	ITEX utility for interfacing cameras.	ITICFG32.EXE
2 Tutorial	A simple Windows console application to capture and display a live image. This example explains how to properly initialize and shutdown an ITEX application	TUTORIAL.C INIT.C BUFFERS.C DISPLAY.C ACQUIRE.C
3 IC-ASYNC AppWizard	Creating an ITEX application template using the IC-ASYNC AppWizard within the Microsoft Development Studio (C++ Version 5.0) environment.	VISION1.CPP VISION1VIEW.CPP ITEXINTR.CPP PROCTHRD.C
4 QuadVis Application	4 Camera asynchronous triggered acquisition using frame reset.	QUADVIS.CPP QUADVIS.H QUADVISVIEW.CPP PROCTHRD.CPP ICA4CHTEST.CNF

The PDF file of this manual uses highlighting and callout boxes to comment the code examples.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CAMERA CONFIGURATOR

Description

The Camera Configurator is a Windows-based utility that dramatically reduces the time and effort required to set up a video camera and acquire images. The Camera Configurator first auto-senses what Coreco Imaging, Inc. image capture boards (IC-ASYNC, PCVision, IC-PCI) are installed in the system.

This tool provides the user a selection of camera models (standard and non-standard) from a pull-down menu. The Camera Configurator establishes all of the manufacturer's recommended parameters for that camera model as default. Each video channel has it's own independent configuration and setup.

The Camera Configurator creates both a binary file (name.cnf) and a text file (name.txt). The binary file is required by the ITEX library software. The ITEX function itx_load_cnf loads binary configuration files at the beginning of an application program.

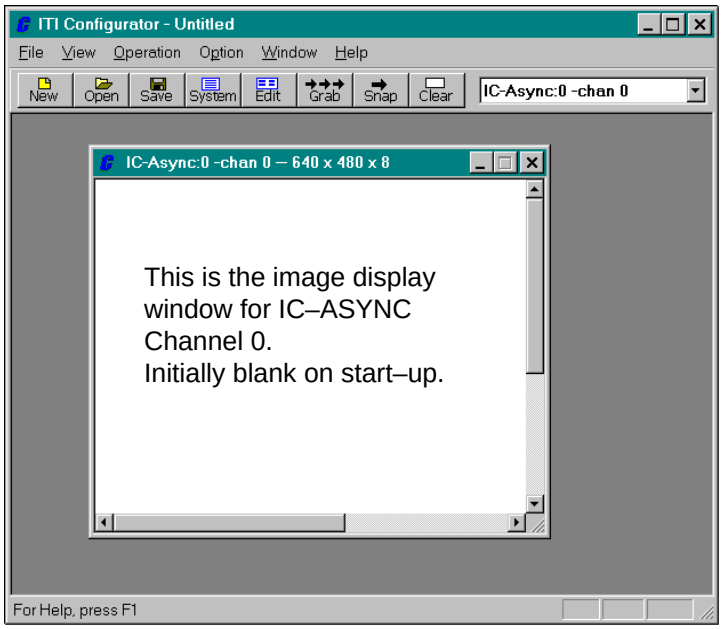


Figure 20. Camera Configurator Startup Screen

Discussion

This tutorial discusses how to use the Camera Configurator to create configuration files for standard and non-standard cameras. By standard, we mean RS-170 or CCIR (PLL or XTAL) and by non-standard we mean "variable scan" or camera which outputs it's own pixel clock, horizontal and vertical sync. Also, we'll show how to interface to a Dual-Tap camera which outputs 2 separate channels of video. At this point, if you haven't connected your camera(s), refer back to the Hardware Installation Section for directions.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Phase-Locked Loop (PLL) or Crystal (XTAL) Mode

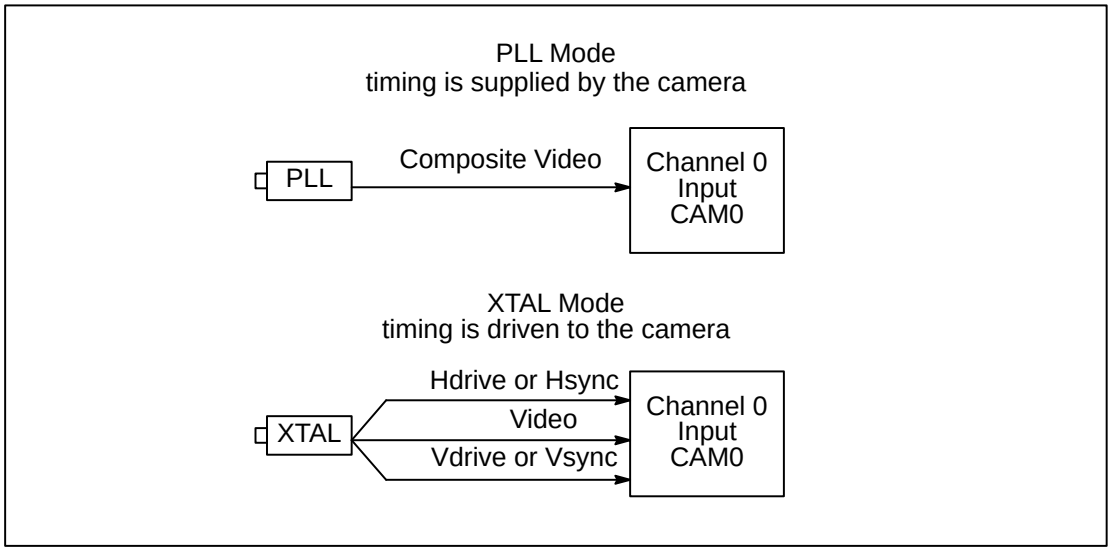


Figure 21. PLL and Crystal Modes

1. Run the Camera Configurator located in your ITEX program folder. This program is also found in the C: \ <ITEX root> \ bin directory.



2. The Configurator will take a moment to detect and initialize the hardware in your system. When it's finished initializing, it appears as the startup screen shown in Figure 22. Right-click in the image display window and select Edit.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

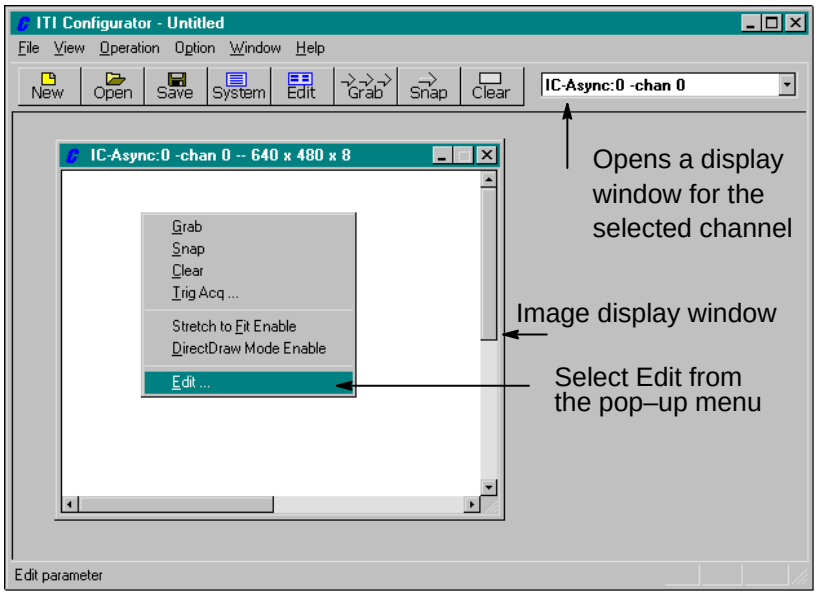


Figure 22. Camera Configurator Startup Screen

3. From the "Camera Selection" list (Figure 23), select a video timebase that best suits your camera. The default timebase is configured for an RS-170 camera in PLL mode with a resolution of 640 H x 480 V. But for example, if you were connecting a CCIR camera and wanted a resolution of 768 H x 574 V, pick **"**CCIR 768x574 PLL".

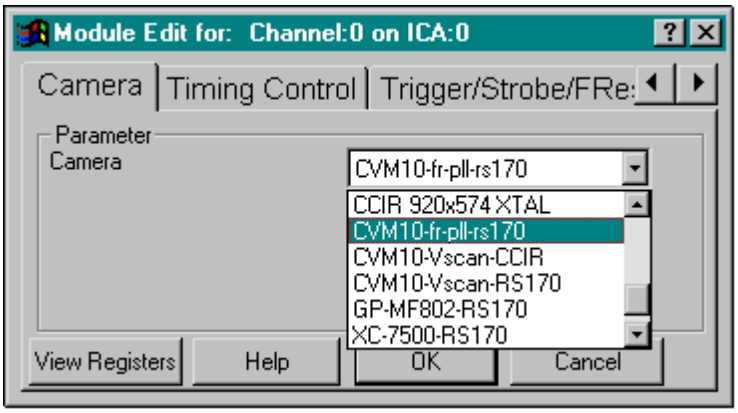
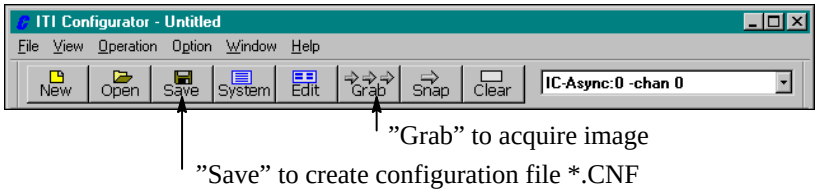


Figure 23. Select Video Timebase for Channel 0 Input

4. Now to acquire a picture, click on "Grab" at the top of the Configurator toolbar. You should now see a live picture displayed in the image window.

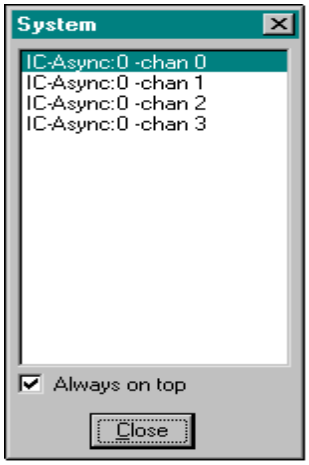
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



To save this configuration, click-on "Save". A dialog box will open displaying the default location to place camera configuration file:
C:\<ITEXROOT>\config

The Configurator creates both a binary file (name.cnf) and a text file (name.txt). As mentioned earlier, this file is very important and will be used later in all your applications. The binary file initializes the system at the beginning of every program.

5. Click-on "System" in the toolbar. This box shows the complete system configuration.



Double–click on a channel to configure the video input. Each channel is independent and has its own configuration.

Figure 24. System Configuration

Configure all your camera ports from here by "Double-clicking" on the channel. If you have other IC-ASYNC boards in the system, they will appear as well.

6. To save an image, "Snap" a picture, then select "Save TIFF file" from the Operation menu.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

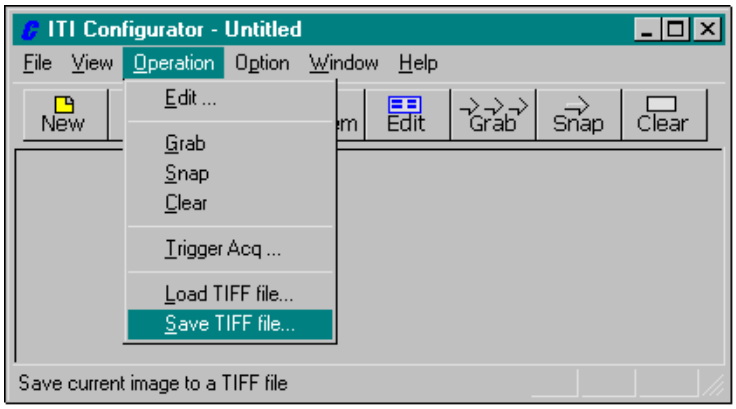


Figure 25. Saving a File

Triggered Acquisition

In the previous section, when "GRAB" was clicked, image were acquired continuously. Many applications require some event to initiate or 'trigger" image acquisition. These events can be hardware (external) or software generated (internal).

For externally triggered acquisitions, a trigger signal (5 Volt, TTL level, positive or negative edge) must be connected to one of four trigger inputs (0-3) or I/O inputs (0-1). These signals are located on the 26-pin MISC/ Video input connector. For software triggered applications, a register on IC-ASYNC must be written to.

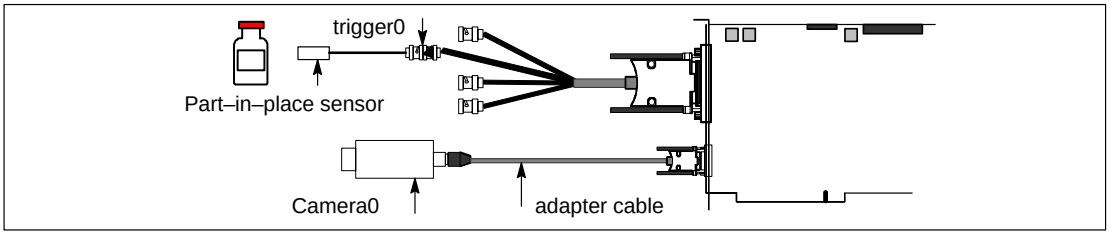


Figure 26. Cable Configuration

1. Connect a trigger signal to Trigger 0 as shown in Figure 26 *(refer to Appendix A: Video Connections for other cabling options).*
2. From the Operation menu, select "Trigger Acq". Select "External" trigger source and click-on "Arm + poll". Input a trigger pulse to snap a picture. The trigger input remains armed, waiting for the next trigger pulse to start another snap.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

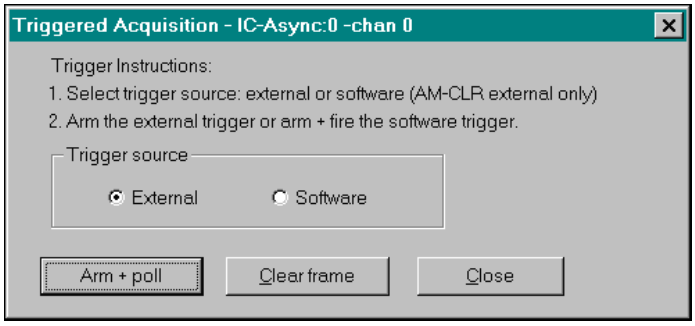


Figure 27. Select Trigger Source

3. To save your configuration permanently, click-on the "Edit" button in the toolbar, then select the "Trigger/Strobe/Freset" tab. Make sure Trigger source, input, and polarity are set properly.

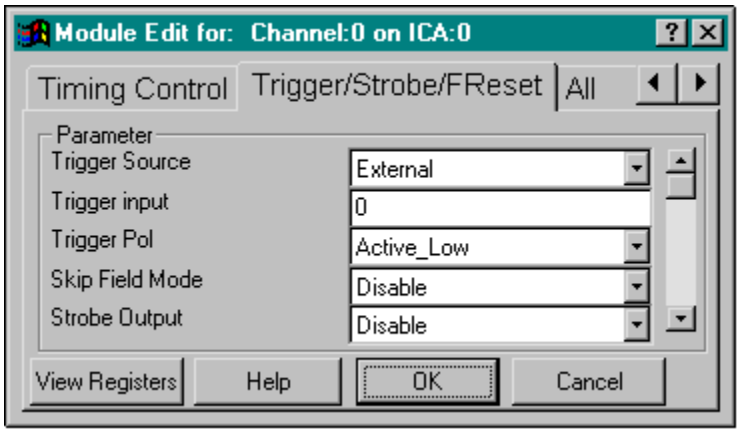


Figure 28. Setting Trigger Parameters

Triggered Acquisition (With Frame Reset)

Many cameras have a frame reset or trigger input. A frame reset signal will force the camera to immediately take a picture. When an external or software trigger occurs, IC-ASYNC immediately sends a frame reset pulse (if enabled) to the camera. Each channel has its own frame reset signal. Figure 29 shows an example using the CV-M10 camera (PLL mode). Figure 30 shows the CV-M10 switch settings for this application.

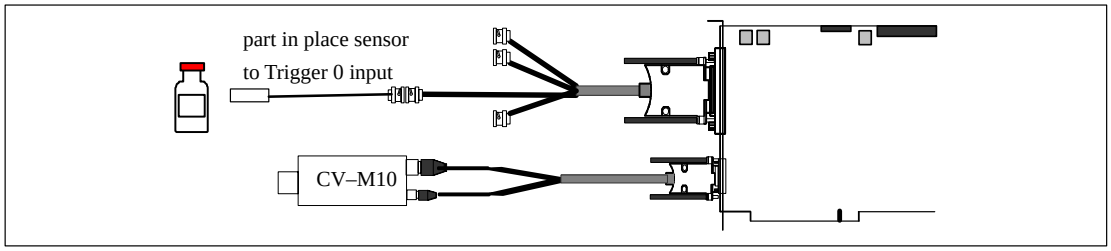


Figure 29. CV-M10 and Frame Reset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

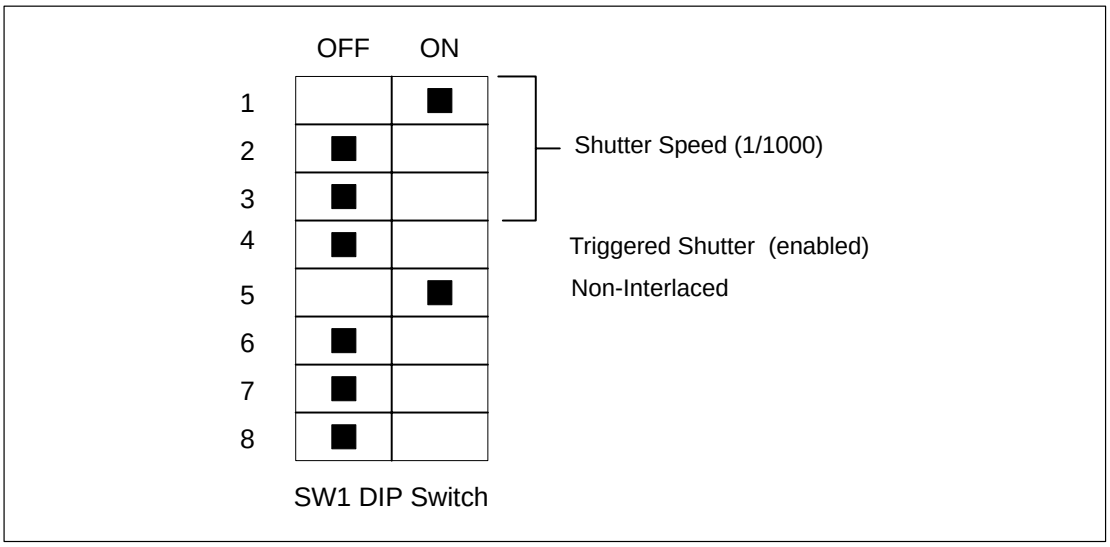


Figure 30. CV-M10 Switches

1. From the camera list, select CVM10-fr-pll-rs170.

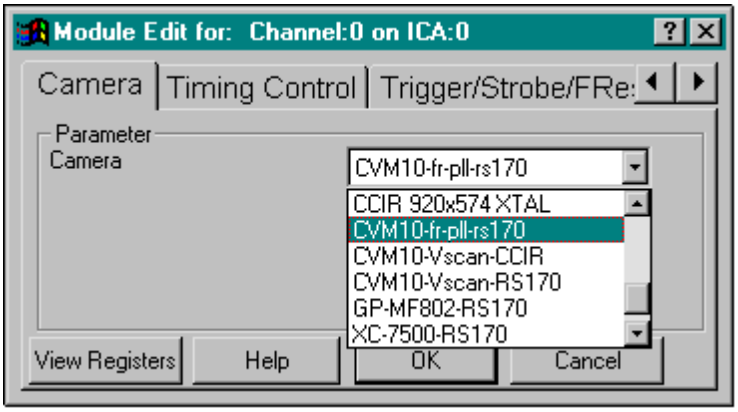


Figure 31. CV-M10 Configuration File

2. Click-on the Trigger / Strobe / Freset tab to view the frame reset parameters. Frame reset is enabled and is set for active low. From the Operation menu, select "Trigger Acq" and select "External" trigger source as before. Click-on "Arm+poll" and input a trigger pulse. Also, make sure the camera is configured for external trigger mode (*For more information on Strobe and Frame Reset mode, refer to the IC-ASYNC Hardware Reference Manual).*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

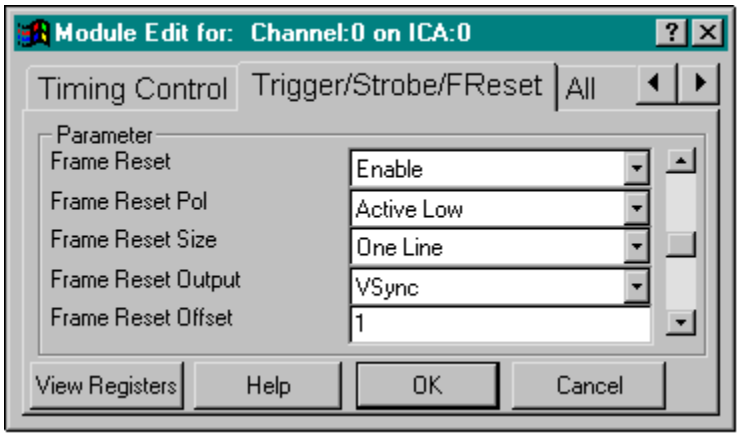


Figure 32. Frame Reset Parameters

Variable Scan Mode

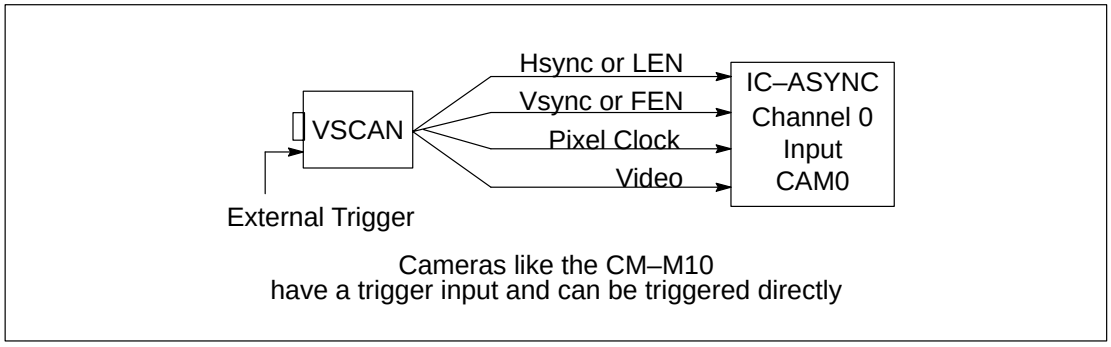


Figure 33. Timing Supplied by Camera

In "variable scan" mode, the camera supplies it's own pixel clock, horizontal and vertical sync. The Configurator supplies some pre-generated configuration files for these type of cameras, for example, the CV-M10 (JAI/Costar). The IC-ASYNC timing generator is disabled.

1. Click-on "Edit", from the camera list select one of the following configuration files: CVM10-Vscan-CCIR or CVM10-Vscan-RS170.
2. Click-on the "Timing Control" tab. Most of these timing parameters are not used in VSCAN, however, they are used to program the PTG. in PLL or XTAL mode (*For more information on programming the P.T.G., refer to the IC-ASYNC Hardware Reference Manual).*

   Highlighted in Figure 34 are the important parameters to configure properly in VSCAN mode. Not shown, but part of the list are LEN, FEN, and Pixel Clock polarity.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

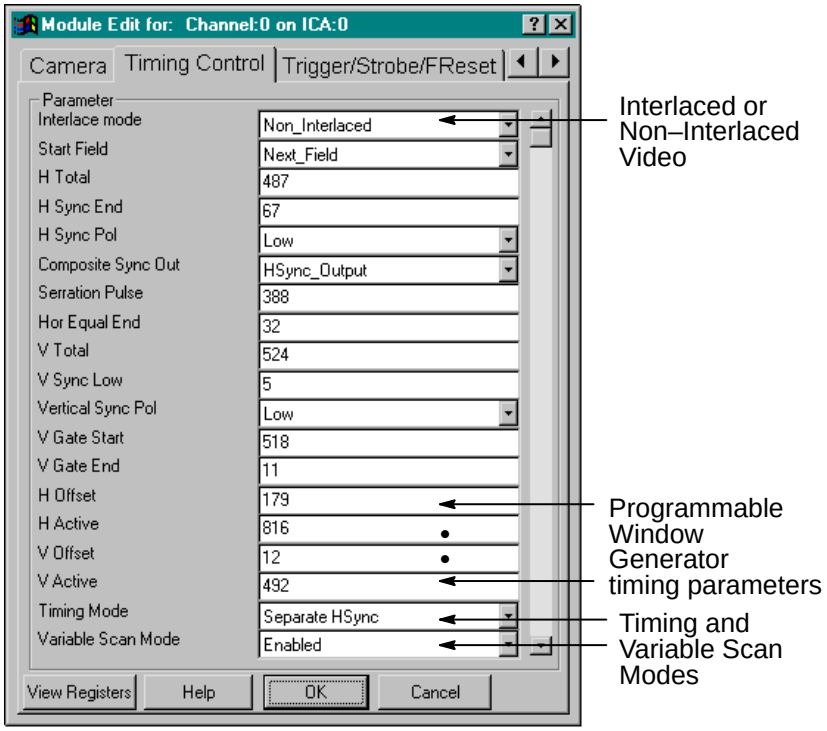


Figure 34. Variable Scan Timing Parameters

Interfacing Dual Channel Cameras

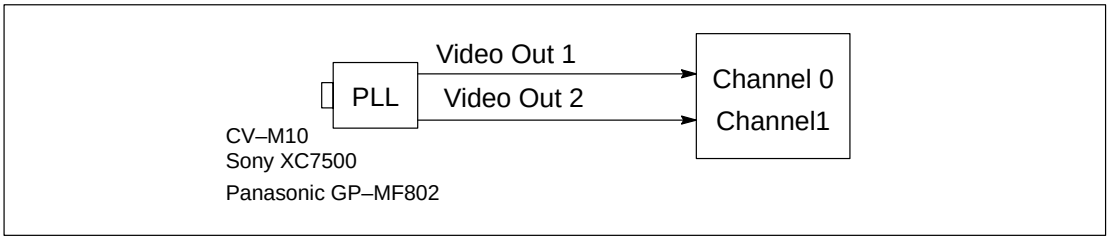


Figure 35. Dual-Channel Configuration

Dual-channel (or tap) cameras output two video signals (ODD /EVEN fields) in parallel, doubling the speed of a normal RS170 or CCIR camera. Typically, during the first field cycle (1/60 sec. RS170 and 1/50 sec. CCIR), one channel outputs the EVEN field and the other the ODD field. These fields toggle every cycle as shown in Figure 36. When grabbing images continuously, the frame grabber needs to reconstruct the two fields in memory before processing or displaying them. This reconstruction is handled automatically by the system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

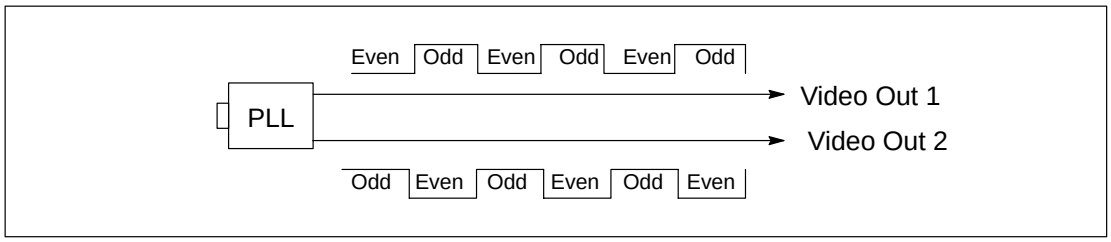


Figure 36. Output Fields Toggle Between Video Channels

To acquire video properly from these cameras, each stream must connect to a separate IC-ASYNC input channel. One channel must be configured as the 'master' and the other a 'slave'. Only channels 0 and 2 can be master inputs. Channels 1 and 3 are configured as slaves. The slave channels are synchronized to the master's timing. The IC-ASYNC pairs channels 0/1 and channels 2/3 together.

1. From the camera list, select a PLL video timebase (RS170 or CCIR) for the master channel (either channel 0 or 2). Then for timing mode, configure the slave channel (channel 1 or 3) to "Dual Tap(Slave)".
2. Click on "Grab" in the Toolbar to acquire continuously.

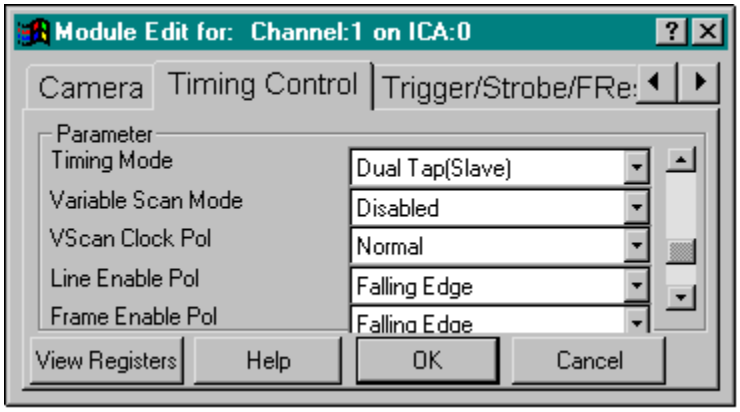


Figure 37. Configure Slave Channel

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TUTORIAL

Description

A simple Windows console program to capture and display an image. This example explains how to properly initialize and shutdown an ITEX application.

Source Files

TUTORIAL.C	Main program.
TUTORIAL.H	Header file containing ITEX include files and program declarations.
ACQUIRE.C	Routine to start and stop image acquisition.
INIT.C	Contains all the setup routines for initializing IC-ASYNC.
BUFFERS.C	Creates and allocates memory space to store the images.
DISPLAY.C	Creates the display channel between host memory and VGA.

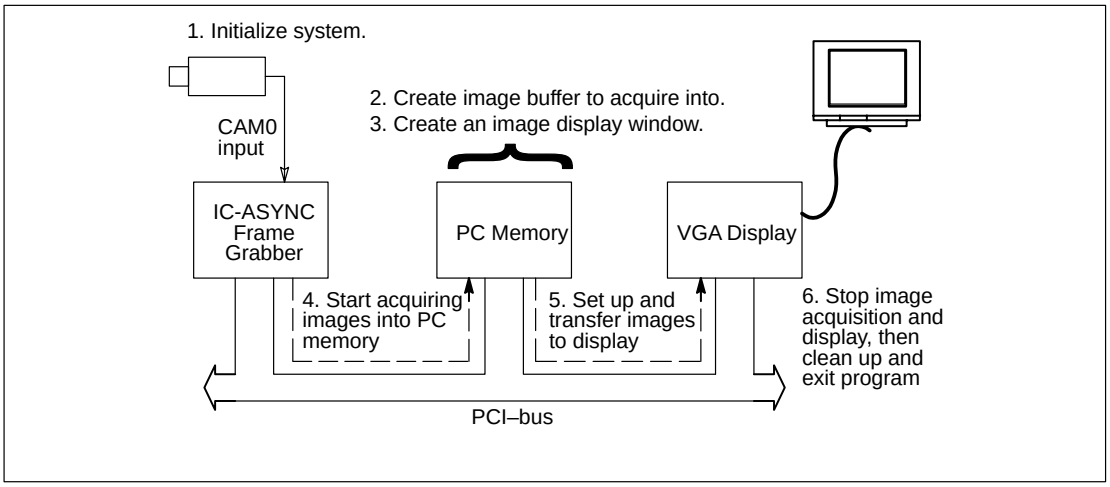


Figure 38. Acquire and Display an Image

Discussion

Let's look at the example first: **TUTORIAL**. This program acquires an image into IC-ASYNC frame grabber memory, transfers it over the PCI-Bus to PC host memory and finally from PC host memory to VGA display. By default, the input video port chosen is CAM0.

NOTE *In all the examples in this section, the term "host" refers to your PC computer memory.*

This example looks for an ITEX configuration file named: **ICATEST.CNF**. This is the default configuration file specified by the PATH environment variable located in your AUTOEXEC.BAT: CNF15040= C:\ITEXROOT\config. This file configures all 4 video ports by default to RS-170. If the video format you're using is not RS170, create a new ITEX Configuration file that matches your camera format by running the ITEX Configurator. If you save the this file to a different name besides ICATEST.CNF, make sure you change the name in the call itx_load_cnf() in INIT.C to reflect this new filename.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TUTORIAL.C: The is the main program file, but it does not contain ITEX specific code. However, it calls all the functions located in the other 4 files which contain ITEX routines to run the application, including the console I/O.

INIT.C: This file contains the ITEX initialization routines which must be called before you begin programming your application. Calls that load the ITEX Configuration file (.CNF), initialize both the hardware and software, get a handle to the IC-ASYNC (this handle is used in every ITEX call that manipulates the board) and retrieve the camera image dimensions.

ITEX Source Code for INIT.C

```
#include <itxcore.h>
```

All ITEX application programs must include this file which contains all core function declaration, etc.

```
BOOL Init()
{
/* perform ITEX initialization here */
/* Set the error level to display all errors */
itx_err_level(WARNING);
/* Auto Configure with defaults */
if (itx_load_cnf("icatest.cnf") != ITX_NO_ERROR) {
```

Use this function to load ITEX Configuration file (.cnf) created by the Configurator. Specifying "AUTO" automatically detects the hardware if no configuration file can be found.*

```
    if (itx_load_cnf("AUTO") != ITX_NO_ERROR){
    MessageBox(NULL,  "Can't Load Config File.", "Initialize",
    MB_OK);
    return FALSE;
    }
}
/* initialize system 0 boards and their daughter boards */
if (itx_init_sys(0) != ITX_NO_ERROR){
```

Initializes the hardware and software, must be called after loading the configuration file but before any other ITEX function calls.

```
    MessageBox(NULL,  "Can't Init System.", "Initialize",
    MB_OK );
    return FALSE;
}
/* Get handle to Async board #0 */
if(!(pBoard = itx_get_modcnf(0,"ICA",SEQ0))){
```

Each board installed in a system requires a handle which is used in subsequent ITEX calls to that board.

```
    MessageBox(NULL,  "Can't Find IC-Async Board.",
    "Initialize", MB_OK );
    return FALSE;
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/* Get handle to Channel 0 */
if(!(pVirtualModule0 = itx_get_vmodcnf(pBoard,0))){
```

Each IC-ASYNC video input requires a separate handle to manipulate the camera channels independently. Call this routine 4 times in a loop if using all 4 channels.

```
    MessageBox(NULL,  "Can't Find Virtual Module 0.",
    "Initialize", MB_OK );
    return FALSE;
}
/* Get image size*/
itx_get_acq_dim(pVirtualModule0,&wWidth,&wHeight);
```

Returns the camera dimensions (image size) for the active camera port. These dimensions are the same as H Active and V Active from your camera config file and can be used to create the display frame.

```
return TRUE;
}
BOOL UnInit()
{
    if(pBuf)
    free(pBuf);
    if(pBoard)
     itx_remove_sys(0);
```

Frees all ITEX memory allocation (device handles, internal structures, etc.) It's the opposite of itx_init_sys. This should be the last ITEX call to your application. Any memory allocated for storing images is the applications responsibility to manage.

```
    return TRUE;
}
```

BUFFERS.C: Allocates enough host memory to store a sequence of image frames. The total amount of memory allocated is equal to: MAXIMUM NUM. OF FRAMES X WIDTH X HEIGHT (of image).

DISPLAY.C: Creates a window and path for the image to be displayed. The path is formed by first creating a "source" and "sink" (can be thought of as source and destination). The "source" in this case is the host buffer in PC memory. The "destination" will be the handle created earlier to our display window. A final call is made to connect the two by the function: itx_create_img_conn().

An End-of-Frame (EOF) interrupt generated by IC-ASYNC determines when and how often an image is transferred between the host memory and window display. Select a hardware interrupt to connect to by calling the function itx_intr_connect(). The handle returned by this function is passed along to the image server created by itx_create_img_server(). The Windows source code in this file is used to create and terminate the display window. The function CleanDisplayHndles() frees all data structures used for creating frames, connections, and sinks.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ITEX Source Code for DISPLAY.C

```
BOOL LiveDisplayFromHostDIB()
{
   CleanDisplayHandles();
   hImageSource = itx_create_host_img_src(NULL,pBuf,ITX_PIX8,
        wWidth,wHeight,0,0,wWidth,wHeight);
```

Makes the image acquired into host memory the "source" image for display.

```
    hImageSink = itx_create_win_img_sink(hWnd,
        ITX_PIX8,0,0,wWidth,wHeight,WIN_DRAWDIB);
```

Makes "hWnd" the "sink" for displaying the image.

```
    hImageConnection =
     itx_create_img_conn(hImageSource,hImageSink);
```

Connects the source to the destination.

```
    // connect interrupt
    if((itx_intr_connect(pVirtualModule0,
    ICA_INTR_EOF, &nIntr)) != ITX_NO_ERROR){
    CleanDisplayHandles(); return FALSE;
    }
```

The image transfer between host memory and display is connected to an End-of-Frame (EOF) interrupt generated by IC-ASYNC.

```
    // create image server
    if((hImageServer= itx_create_img_server(
     hImageConnection, nIntr, 1)) == NULL){
      CleanDisplayHandles(); return FALSE;
```

Spawns an image server thread that displays the image connection made by itx_create_img_conn(). This function displays the image every nth interrupt, in this case, after every frame acquisition.

```
    }
}
```

ACQUIRE.C: The last step is to start the acquisition between IC-ASYNC and host memory which is handled by the routine: itx_host_start_acq(). And to halt the acquisition: itx_host_grab_stop().

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IC-ASYNC AppWizard

Description

This tutorial explains how to create an ITEX application template using the IC-SYNC AppWizard within the Microsoft Development Studio (C++ Version 5.0) environment. This template provides the programmer a baseline framework for developing and inserting application code.

Source Files

VISION1.CPP	Main ITEX Application
VISION1View.CPP	Implementation of the VISIONView class
ITEXINTR.CPP	ITEX Interrupt Implementation
PROOCTHRD.CPP	Image Processing Thread

Discussion

Make sure Microsoft Development Studio has been installed prior to the IC-ASYNC software, otherwise the ITEX AppWizard and Extension Help Utility will not be setup correctly. ITEX Extension Help gives the programmer quick access to on-line help for all ITEX functions by highlighting a function then pressing the 'F1' hotkey. Check under the Help Menu in DevStudio to see if **"Use Extension Help"** has been enabled (checkmark indicates enabled).

Open the Microsoft Development Studio and select: **File - New**. Click on the Projects tab, select **IC-ASYNC AppWizard**, and enter a Project name and Location to place your application, as shown in Figure 39.

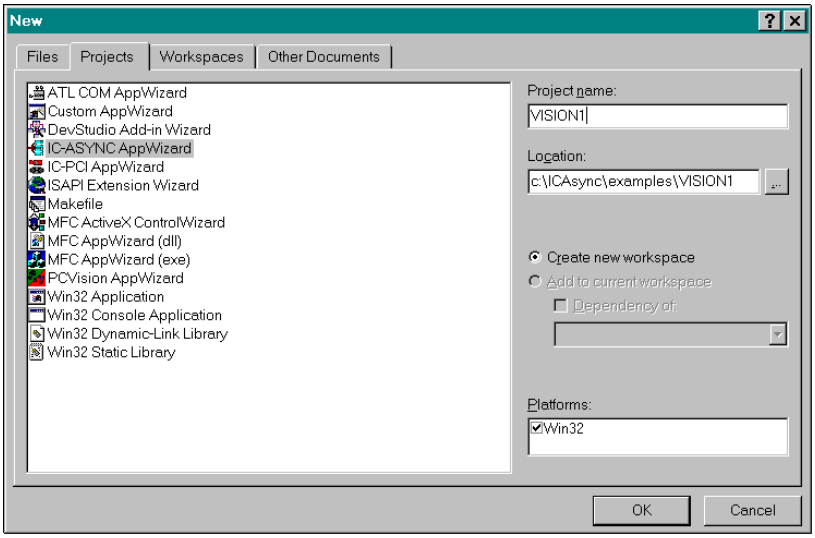


Figure 39. Create a New Project

Step 1. Select which type of application to create, Single Document, Multiple Document", or "Dialog Based" (no display). In the real application, you will need to determine if a display window is necessary or not. Such questions as: do you want a single display window or do you want a separate display window for

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

each camera, or no display at all? Typically for development, display windows are very useful (in most cases necessary), but in the actual application the display window can be unnecessary overhead added to the host CPU and is not required.

If your application requires one image display window, use "Single document"; two or more, use "Multiple document"; and if no image display is necessary select "Dialog based". For this example, we'll create an application that uses two display windows. Select "multiple document" as shown in Figure 40.

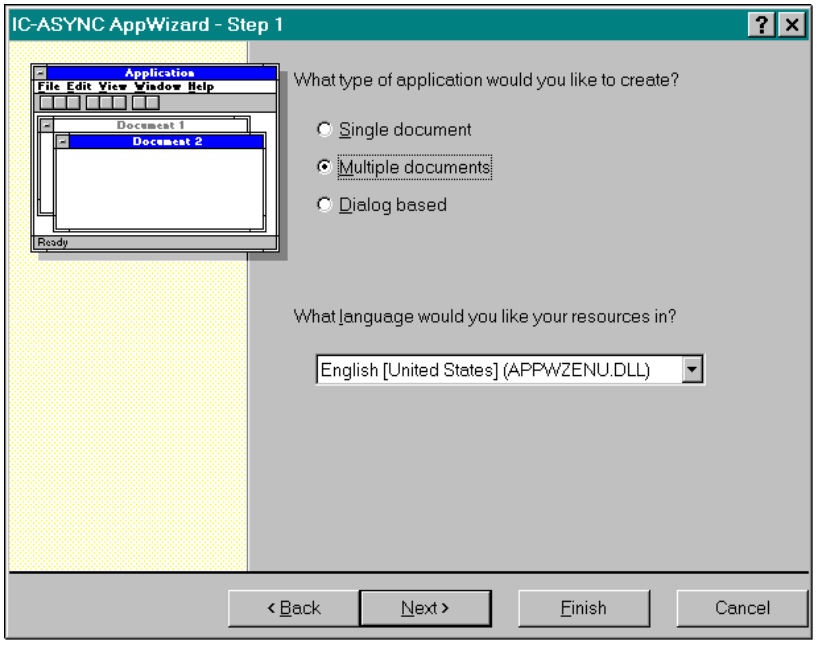


Figure 40. Select Type of Application

Steps 2-6. For this example, select the defaults pre-chosen. All these screens affect how Windows code is generated, but not ITEX.

Step 7. This remaining AppWizard screen selections do affect how ITEX code is generated starting with the image display. In Step 7, chose the image source to be displayed: **Camera** (unprocessed), **Host Buffer** (processed) or **None** (no display at all). Select **"Host buffer"** which enables us to display a processed image from the host buffer. In Figure 41, observe how the image is transferred in each case.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

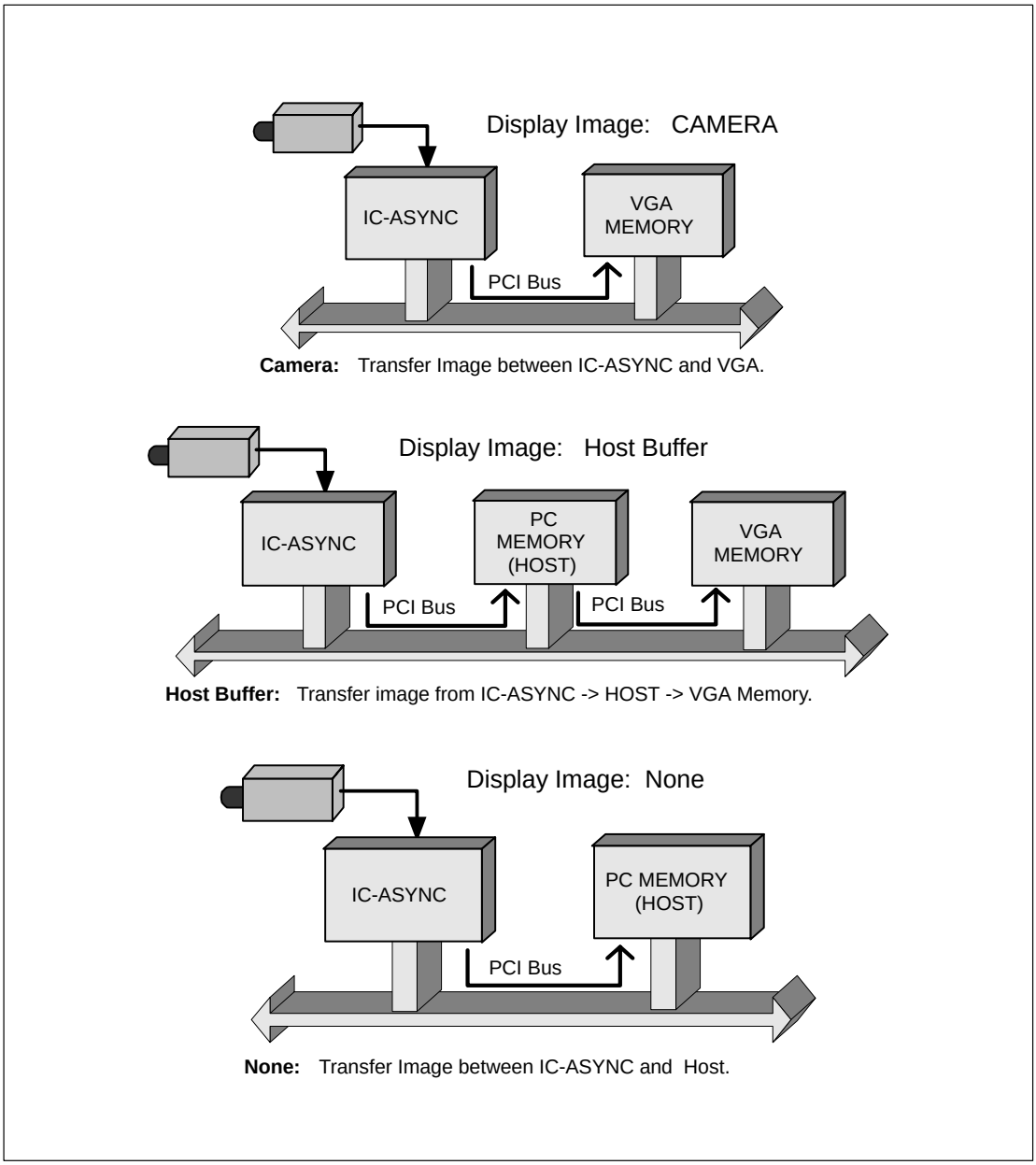


Figure 41. Image Transfer Paths

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Next, select how frequent the display is updated (in units of "frame" time). In this case, select "Every Frame" to display in "real-time". Later on, we'll take a closer look at how these image frames are stored in memory, including other ways to change the display frequency.

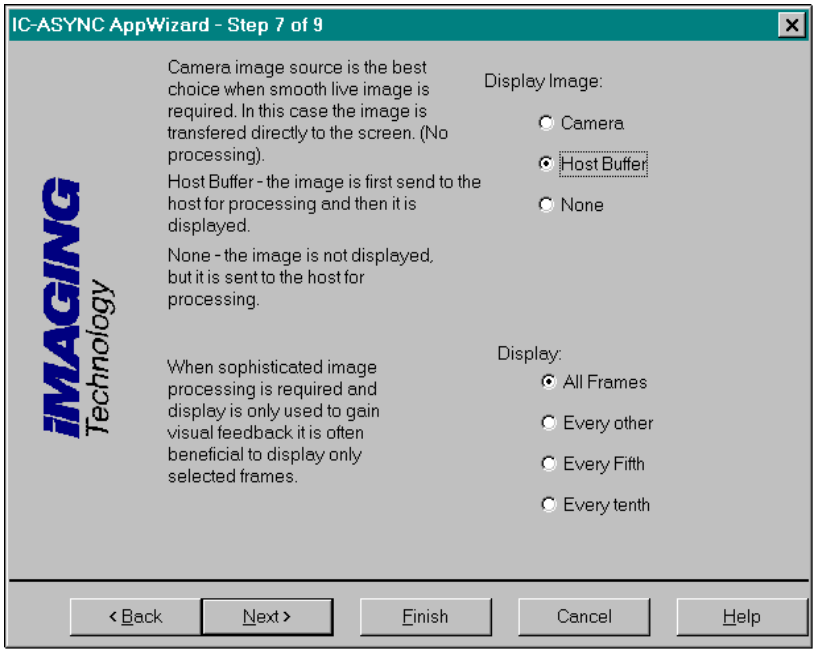


Figure 42. Choose Image to Display

Step 8. Now select which hardware interrupt(s) to connect and enable. IC-ASYNC can generate interrupts based on the 9 events listed in Figure 43. For each interrupt selected, the AppWizard creates a separate thread (user routine). The programmer can easily insert application code inside these threads which get executed whenever the interrupt occurs.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

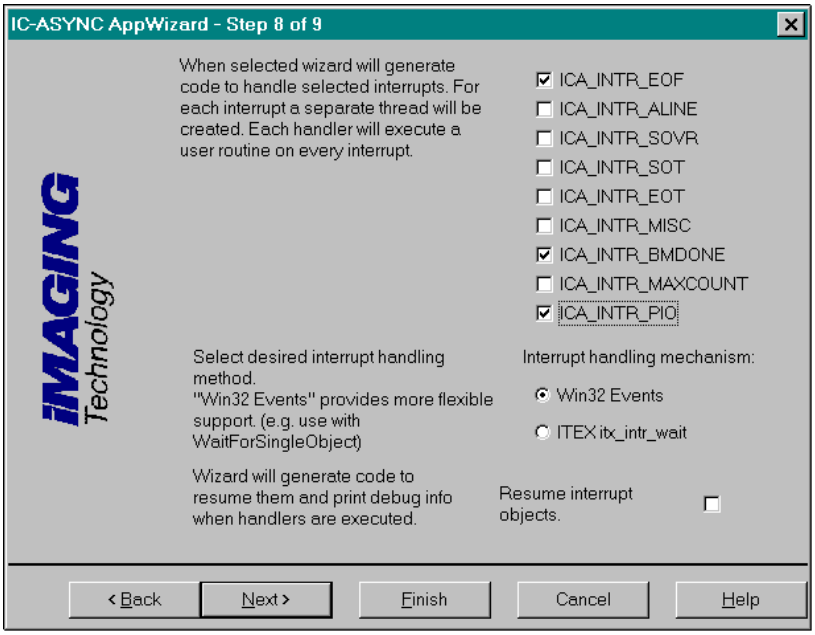


Figure 43.Connect Interrupts to Application Code

ICA_INTR_EOF - Interrupt on End Of Frame

ICA_INTR_ALINE - Interrupt on acquisition address

ICA_INTR_SOVR - Interrupt on acquisition overrun

ICA_INTR_SOT - Interrupt on Start Of Trigger

ICA_INTR_EOT - Interrupt on End Of Trigger

ICA_INTR_MISC - Interrupt on MISC connector external input

ICA_INTR_BMDONE - Interrupt when bus mastering completed

ICA_INTR_MAXCOUNT - Interrupt on bus master frame skip

ICA_INTR_PIO - Interrupt on I/O connector interrupt input

Select these three events: INTR_EOF (End of Frame), INTR_BMDONE (Bus Master Done), and INTR_PIO (Digital port Interrupt input). For End of Frame, an interrupt gets generated after the image is acquired into IC-ASYNC memory. For Bus Master Done, an interrupt gets generated after the transfer is complete across the PCI bus between IC-ASYNC and host or VGA memory. And for Digital port I/O, an interrupt gets generated after sending or receiving a trigger signal via the digital port (for example, this signal could from a part-in-place sensor that's wired to the digital I/O).

Below the interrupt list, there's a choice for handling interrupts. **Select the default, Win32 Events** (which uses standard Win32 synchronization methods for handling interrupts, also "time-out" granularity is programmable in milliseconds). Enabling "Resume interrupt objects" causes threads to respond to interrupts immediately upon creation, otherwise, a user call is required to activate the interrupt procedures.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Step 9. Select overlay type either **"No Overlay" or "Print Overlay with Sample Image"**. Note "YCrCb Surface" requires a VGA card that supports hardware mixing (see Video Overlay section of User's Manual). Also refer to the Overlay Example in this section to see how software and hardware overlay graphics are implemented. Now click **"Finish"** to build skeleton project. When the project build completes..

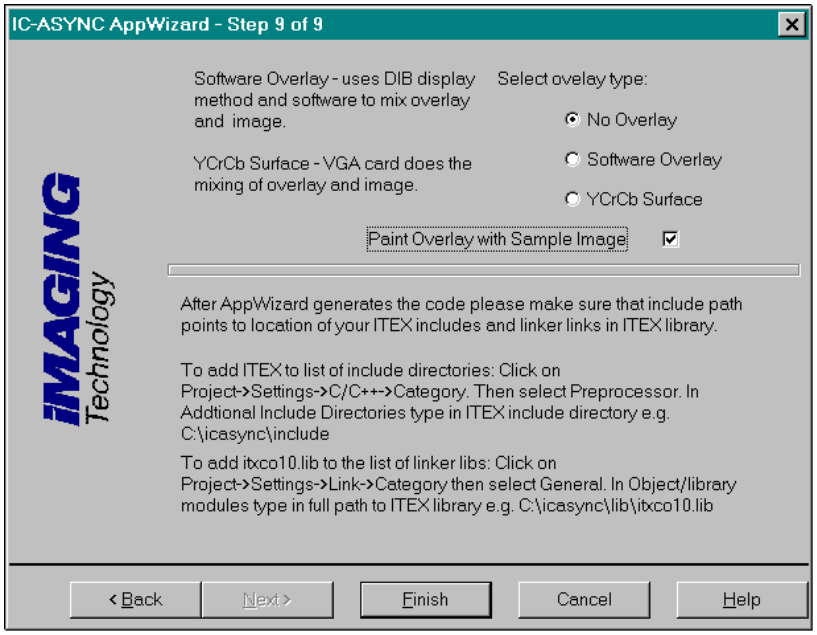


Figure 44. Select Overlay Type

Step 10. Now try building and running the application, select **"Re-build All"** from the Build menu in DevStudio. Note, should you get any compilation or unresolved link errors, make sure the ITEX include and library paths have been set correctly (these paths are automatically set during ITEX installation).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

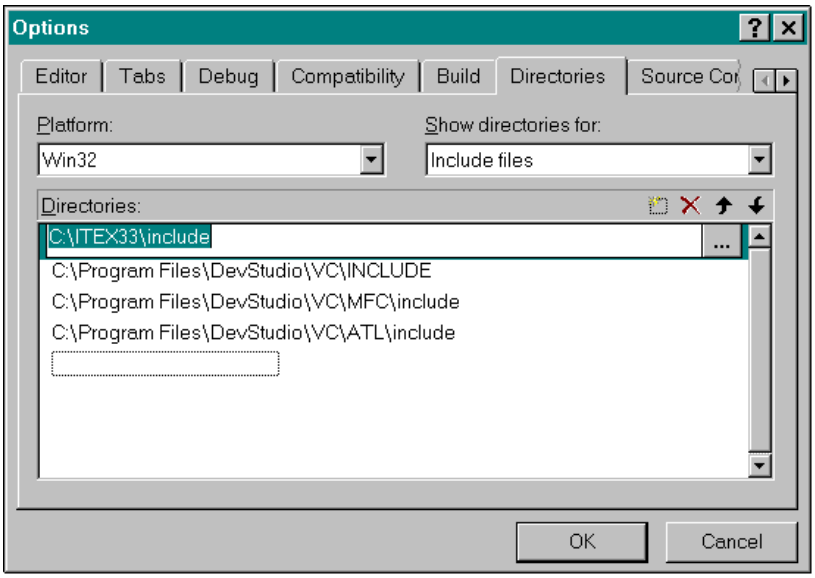


Figure 45. Check ITEX Include Directory

Verify the include path: C:\<*ITEXROOT*>\include (shown in Figure 45). Verify the library path: C:\<*ITEX-ROOT*>\lib (shown in Figure 46).

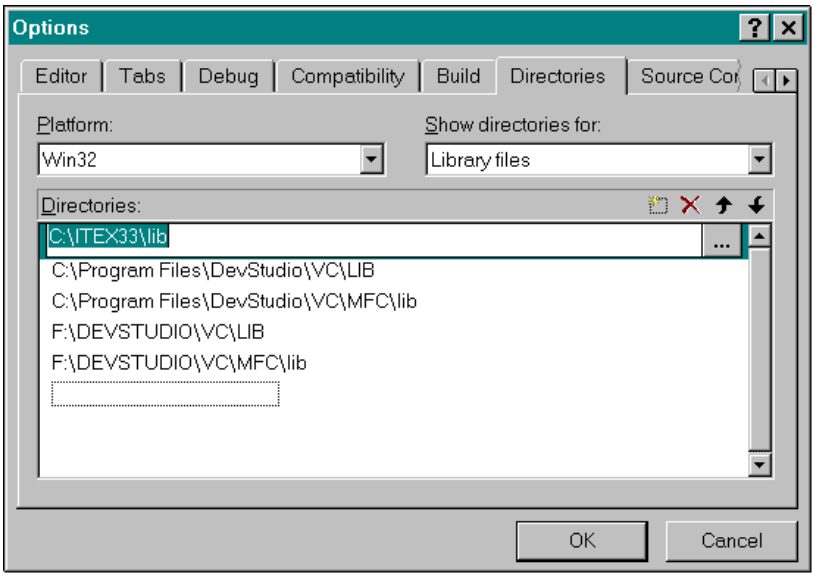


Figure 46. Check ITEX Library Directory

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

You should now have an IC-ASYNC AppWizard generated application running with one live image window that looks similar to Figure 47.

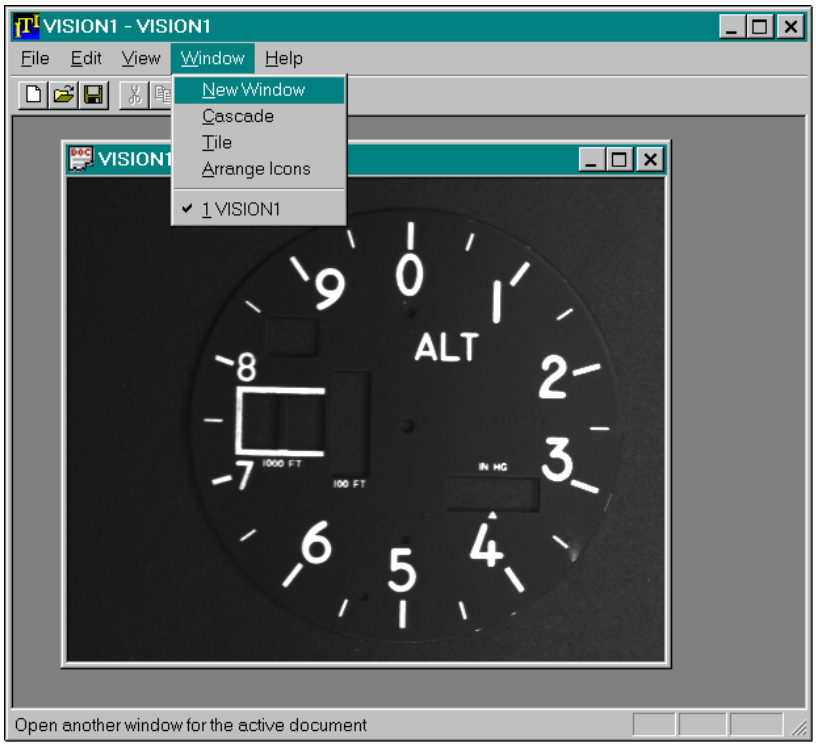


Figure 47. IC-ASYNC AppWizard Application

Try adding more image windows by selecting **"Window - New Window".**

The AppWizard setup portion is now complete, a project has been created and built so let's now take a look at the ITEX code.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IC-ASYNC GENERATED CODE

The three key questions we want to know about as far as the application code is concerned are: How is the system initialized? How are the interrupts implemented? and How do we create and implement image processing threads?.

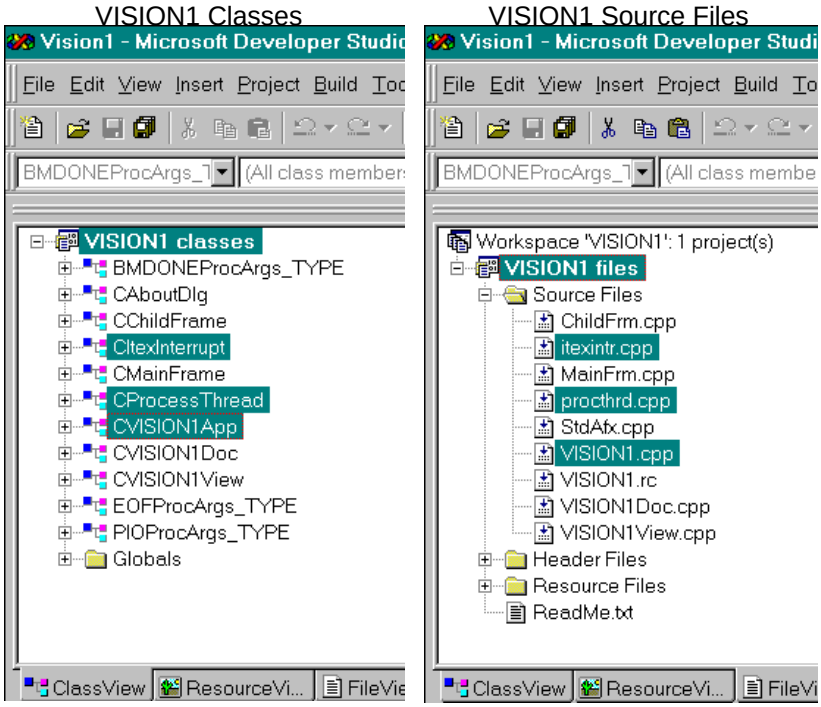


Figure 48. VISION1 Classes and Source Files

Although there's certainly more to explore, these three are the most important to the application. Listed in Figure 48 are all the source files and classes generated by the IC-ASYNC AppWizard in the Microsoft Dev-Studio, the highlighted ones are the key ITEX files we need to examine.

NOTE *Code listings in this section show only the code fragments relating to ITEX.*

System Initialization.

We'll start with the beginning of the CVision1 application located in VISION1.CPP. Take a look at the CVision1App Initialization. Here's where ITEX initializes the system. The function, InitializeITEX(), is responsible for loading the ITEX configuration file then detecting and initializing each IC-ASYNC in the system.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

VISION1.CPP: Application Initialization.

```
//////////////////////////////////////////
// The one and only CVISION1App object
CVISION1App theApp;
//////////////////////////////////////////
// CVISION1App initialization
```

Start of application

```
BOOL CVISION1App::InitInstance()
{
    // initialize IC-ASYNC hardware
    if(InitalizeITEX() == FALSE)
```

ITEX System Init. call

```
        return FALSE;
        . . .
    // initialize interrupt objects
    InitializeItexIntr();
```

ITEX Interrupt Init. call

```
    . . .
    return TRUE;
}
//////////////////////////////////////////
BOOL CVISION1App::InitalizeITEX()
{
// Standard initialization of the boards and will appear in all applications using
this hardware
     . .
// Load the Configuration file
if(itx_load_cnf("icatest.cnf") != ITX_NO_ERROR) {
```

Load ITEX Configuration File

```
         . .
        return FALSE;
        }
}
    . . .
// initialize system 0 boards
if(itx_init_sys(0) != ITX_NO_ERROR) {
        . . .
        return FALSE;
    }
//detect number of IC-ASYNC boards in the system
while(m_pBrdMod[m_iBoardCount] =
      itx_get_modcnf(0, "ICA", SEQ0 +m_iBoardCount)){
```

Detect the number of IC-ASYNC boards in system

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
        . . .
    return TRUE;
}
////////////////////////////////////////
BOOL CVISION1App::InitializeItexIntr()
{
        . . .
```

Initialize Interrupt Objects: End-of-Frame,

```
    m_pIntrObj[THREAD_EOF] = new CItexInterrupt(pVMod, ICA_INTR_EOF, ItexIntrEOF,
&EOFProcArgs, NULL, NULL);
    m_pIntrObj[THREAD_BMDONE] = new CItexInterrupt(pVMod, ICA_INTR_BMDONE,
ItexIntrBMDONE, &BMDONEProcArgs, NULL, NULL);
    m_pIntrObj[THREAD_PIO] = new CItexInterrupt(pVMod, ICA_INTR_PIO, ItexIntrPIO,
&PIOProcArgs, NULL, NULL);
    //use m_pIntrObj[XXXXXXX]->Resume(); to start execution of any of the interrupt
objects
    //look at all member functions of CItexInterrupt in ItexIntr.cpp
    return TRUE;
}
```

The function call **InitializeItexIntr()** initializes the interrupt objects: EOF, BMDONE, and PIO. By default, these objects are 'sleeping' at initialization time, to start execution of these interrupt objects, uncomment the line of code right after their initialization (shown at the bottom of the previous code listing for VISION1.CPP). These interrupts can be enabled at startup by checking "Resume interrupt objects" in Step 8 of the AppWizard setup.

Interrupt Handling

Not every application may require the use of interrupts. Interrupt driven applications free the host CPU from doing unnecessary tasks, such as waiting for an image acquisition to complete. Interrupts are generated from many different events, hardware or software. Here we're looking at hardware events. During setup, we chose to enable EOF (end-of-frame), BMDONE (bus master transfer of image has completed), and PIO (digital port I/O).

ITEX provides an interrupt object class interface that's very easy to implement in your application (see ITXINTR.CPP code). For each interrupt object created, a thread procedure is generated and connected to it. The programmer has two options to either 'suspend' or 'resume' thread procedures. 'Resume' immediately starts execution of an interrupt thread and 'suspend' stops thread execution.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

ITEX Interrupt Implementation: ITEXINTR.CPP.

```
// ItexIntr.cpp: implementation of the CItexInterrupt class.
//
//////////////////////////////////////////////
    . . .
// default interrupt timeout
static const int iDefaultTimeout = 1000; // miliseconds
// Procedure that all threads are running
UINT IntrThreadProc(LPVOID pParam)
{
    . . .
        Connect to hardware interrupt
// connect interrupt that this thread will service
    if (itx_intr_connect(pParent->m_pMod, pParent->m_iEvent, &pParent->m_Intr) !=
ITX_NO_ERROR ){
        ::MessageBox(NULL,"Could not connect interrupt", "Error", MB_OK |
MB_ICONSTOP);
        return 0;
    );
    . . .
    //enter thread's main loop
    while(pParent->m_bContinue == TRUE){
        Thread sleeps in main loop until interrupt is generated, then executes user function.
    //wait for an interrupt
    bIntrTimedOut = (WaitForSingleObject(pParent->m_Win32Event, iDefaultTimeout) !=
WAIT_OBJECT_0);
        itx_intr_event_rearm(pParent->m_Intr);
        if (!bIntrTimedOut){
    //got interrupt execute user interrupt function
            if(pParent->m_pFunc != NULL)
                (*pParent->m_pFunc)(pParent->m_pFuncParam);
        }else{
            //timed-out waiting for an interrupt execute user's time-out function
            if(pParent->m_pTimeoutFunc != NULL)
                (*pParent->m_pTimeoutFunc)(pParent->m_pTimeoutFuncParam);
        };
    };
    . . .
    //disconnect interrupt
        Disconnect from interrupt event and clear hardware interrupt.
    itx_intr_disconnect(pParent->m_Intr);
    return 0;
}
```

ITEXINTR.CPP shows the procedure used by all threads: **IntrThreadProc()** First, we connect to the specific IC-ASYNC hardware interrupt, then the thread is put to sleep until the interrupt comes along. When

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

the interrupt occurs, the user's interrupt function gets executed. Code for each corresponding interrupt can be added to VISION1.CPP (shown in the following code listing for VISION1.CPP). After the user's function completes, the thread goes back to sleep again until the next interrupt occurs.

In VISION1.CPP has a predefined structure to place parameters that are passed to the user function. Inside this structure, a counter is defined. Below the structure is the function where the user places code that gets executed when the interrupt is generated. When EOF (End-of-Frame) interrupt occurs, this code gets executed, in this case the counter increments by one. The other two interrupt procedures (BMDONE and PIO) are placed in this file as well.

VISION1.CPP: Adding User Code to Interrupt Procedures

```
VISION1.CPP: Adding User Code to Interrupt Procedures.
////////////////////////////////////////////////
// CVISION1App commands
////////////////////////////////////////////////
// CVISION1App parameter structures and interrupt procedures
//define structure used to pass arguments to interrupt procedure
struct EOFProcArgs_TYPE{
    unsigned long int_counter;
```

Add counter variable.

```
    // TODO: below add other data that you want to
    // pass to corresponding interrupt procedure
```

Add user parameters to pass into interrupt procedure.

```
    //
    //
}EOFProcArgs;
void ItexIntrEOF(LPVOID Param)
{
    EOFProcArgs_TYPE *pParam;
    pParam = (EOFProcArgs_TYPE*) Param;
    (pParam->int_counter)++;
```

Increment Counter.

```
    // TODO: add code to be exected when
    // corresponding ITEX interrupt is asserted
```

Add user code to interrupt procedure here.

```
    //
    //
};
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Image Processing Threads.

Earlier in AppWizard Step 7, we chose **Host Buffer** so the image would be transferred into host memory from IC-ASYNC making it available for processing. When Host Buffer is selected as the image source file, PROCTHRD.H and PROCTHRD.CPP are generated. These files implement a CprocessThread object that's called to perform some image processing task. A separate thread is created to perform image processing routines. It operates on a circular buffer of images stored in CPU host memory. The circular buffer size is determined by FRAMES_IN_HOSTBUFF (set to 5 by default, located in the file VISION1View.H).

The process thread executes in a loop the following way. When the thread begins execution, the first step is to get a pointer to the latest frame deposited in the host buffer is retrieved (or another option is to wait for the current to finish acquisition and grab that one.

This frame is 'locked' down to prevent the 'host grab' from overwriting the buffer currently being processed (Figure 49). The other 4 frames will continue to be written to while the locked frame is being processed.

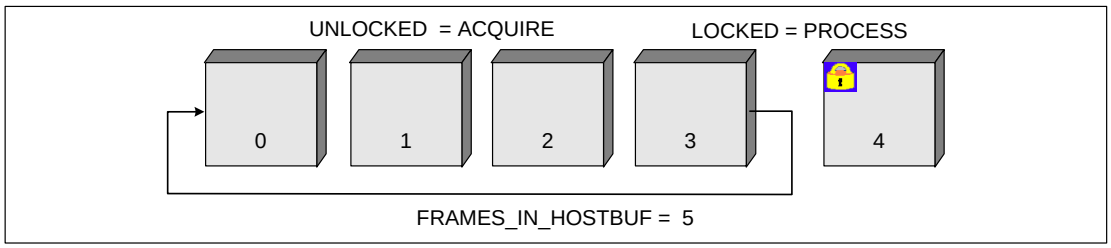


Figure 49. Unlocked versus Locked Image Frames

The locked frame gets processed, for example, in this case by the **FrameThreshold** function (just uncomment the function and recompile the program). After the frame is processed, it's sent to display and the frame becomes unlocked and available to the circular buffer for acquisition.

Image Processing Thread: ProcThrd.cpp

```
// ProcThrd.cpp: image processing thread object
//
//////////////////////////////////////////////
#include "stdafx.h"
#include "VISION1.h"
#include "VISION1View.h"
#include "ProcThrd.h"
    . . .
//////////////////////////////////////////////
// CProcessThread
// code below is just an example of the image processing function that
// can be executed by image processing thread - remove when not used
void FrameThreshold(BYTE *imgBuff, int imgSizex, int imgSizey, int threshold)
```

FrameThreshold() image processing functions (user code).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
{
#define BLACK 0
#define WHITE 255
#define THRESHOLD_LEVEL 128
    . . .
return;
}
// This is the image processing and display thread
DWORD WINAPI ProcessThread(LPVOID pOwner)
{
CProcessThread *pParent = (CProcessThread *)pOwner;
    . . .
// set board in mode: grab into host circular buffer
hGrabID = itx_host_grab(pView->m_pViewBoardMod, pView->m_pbyHostBuffer,
FRAMES_IN_HOSTBUFF);
```

Start acquiring frames into circular buffer in host memory.

```
while (!pParent->m_bShutdown) {
// Wait for a new frame to become available, and lock it
// so it can't be overwritten while we are processing.
if((iSeqNum=itx_grab_latest_seqnum(hGrabID,TRUE,&pHostBuffFrame,TRUE))>=0){
```

Get latest frame and lock it so it's not overwritten during processing.

```
    // ****************************************
    //  uncomment line below to perform example of image processing
    // FrameThreshold(pHostBuffFrame, iWidth, iHeight, THRESHOLD_LEVEL);
    // ****************************************
    // TODO - insert any image processing code here
```

Insert user code for processing image (uncomment FrameThreshold and recompile code).

```
    //
    //
    //
    // Update the address associated with the image source to
    // point to the frame we just processed
    itx_set_host_img_src_addr(pView->m_hImgSrc, pHostBuffFrame);
    // Display the processed frame in our client window
    if(iFrameCounter++%iDisplayFreq == 0)
        itx_display_conn(pView->m_hImgConn);
    // Release the frame back into the active circular acquistion buffer
        itx_host_grab_release(hGrabID,iSeqNum);
```

Display processed frame then unlock it so it can acquire a new frame.

```
    }
}
// stop grab operation
    itx_host_grab_stop(hGrabID);
```

Stop acquiring images.

```
    return 0;
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Dual-Tap Cameras

Acquiring images from dual-tap cameras requires a few modifications to the pre-generated code shown previously.

Modification to Vision1View.h

```
// Vision1View.h : interface of the CVision1View class
//
. . .
    MODCNF *m_pViewBoardMod;          // VMod used by this view
    // dual-tap support added
    MODCNF *m_pSlaveViewBoardMod;
    // end dual-tap support added
. . .
```

Modification to Vision1View.CPP

```
// Vision1View.cpp : implementation of the CVision1View class
//
. . .
//initialize display related variables
    CDualchnApp *theapp=(CDualchnApp*)AfxGetApp();
    m_iViewBoard = theapp->m_iActiveBoard;
    m_iViewVMod = theapp->m_iActiveVMod;
    m_pViewBoardMod = theapp->m_pVMod[m_iViewBoard][m_iViewVMod];
    // dual-tap support added
    if ((m_iViewVMod&1)==0)
        m_pSlaveViewBoardMod = theapp->m_pVMod[m_iViewBoard][m_iViewVMod+1];
    else
        m_pSlaveViewBoardMod = NULL;
    // end dual-tap added
. . .
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Modification to ProcThrd.CPP

```
// ProcThrd.cpp: image processing thread object
//
////////////////////////////////////////////
. . .
// This is the image processing and display thread
DWORD WINAPI ProcessThread(LPVOID pOwner)
{
. . .
    int iSeqNum;
    HITXGRAB hGrabID;
    int iDisplayFreq = pView->m_nFrameDispFreq;
    long iFrameCounter = 0;
    int iWidth = pView->m_wViewWidth;
    int iHeight = pView->m_wViewHeight;
    // set board in mode: grab into host circular buffer
    // dual-tap support added
    if (pView->m_pSlaveViewBoardMod)
        hGrabID = itx_host_start_dualtap_acq(pView->m_pViewBoardMod,
pView->m_pSlaveViewBoardMod, DUALTAP_FIELD_TOGGLE_MODE, 0, 0,
        iWidth, iHeight, pView->m_pbyHostBuffer, FRAMES_IN_HOSTBUFF,
ITX_INFINITE_FRAMES);
    else
    // end dual-tap support added
        hGrabID = itx_host_grab(pView->m_pViewBoardMod, pView->m_pbyHostBuffer,
FRAMES_IN_HOSTBUFF);
. . .
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

FOUR CHANNEL TRIGGERED ACQUISITION

Description

This tutorial shows how to configure the IC-ASYNC to acquire from four separate cameras that are externally triggered (using frame reset). Each channel uses it's own trigger input to initiate acquisition. In this example, all four cameras are configured to output full-frame, non-interlaced video (at 640H by 480V).

The source code provided was generated by the IC-ASYNC AppWizard exactly as in the previous tutorial. Only a few lines of code (highlighted in this section) need to be added to support 4-channel acquisition.

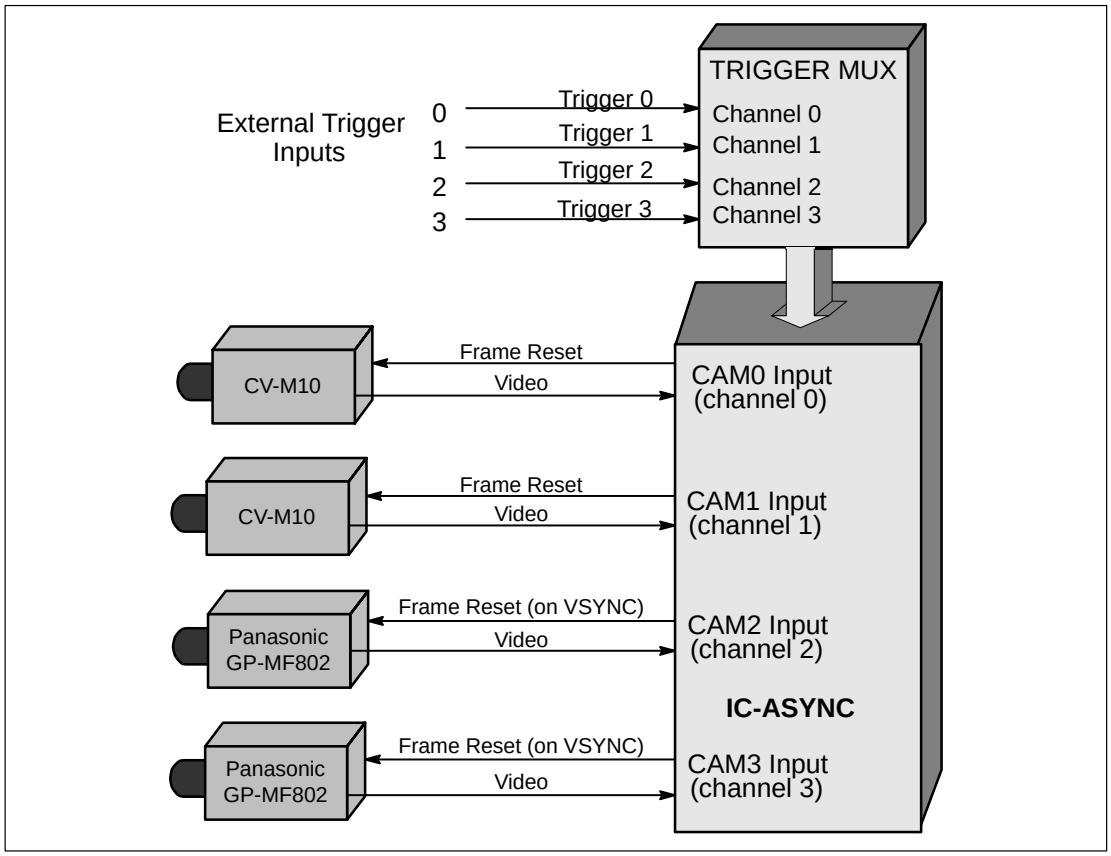


Figure 50. Four Camera Setup

Source Files

QuadVis.CPP	Main ITEX Application (and initialization)
QuadVis.H	Main ITEX header file
QuadVisView.CPP	Display control
procthrd.CPP	Spawns 4 acquisition threads
ica4chtest.CNF	ITEX camera configuration file

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Discussion

QuadVis is a self-contained application that spawns four processing and display threads. Each time a trigger occurs on a particular channel, a new picture is acquired, transferred to the host for processing, then displayed (using DIBs). QuadVis uses the camera config file ICA4CHTEST.CNF. This file can be modified to suit your application.

Each camera is connected to one of the CAM0-3 break-out input ports. Video ports CAM0 and CAM1 have CV-M10 cameras hooked up to them and CAM2 and CAM3 ports have Panasonic GP-MF802s connected. All four cameras are put in trigger mode. Figure 51 illustrates the camera switch configurations, and Figure 52 illustrates the cables.

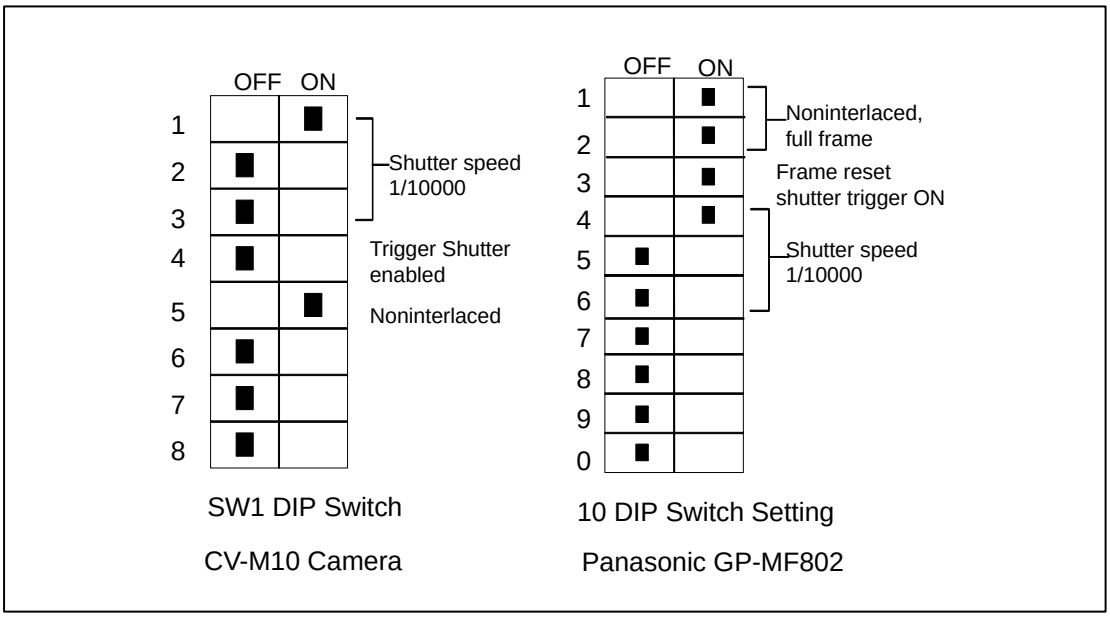


Figure 51. Camera Configuration Switches

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

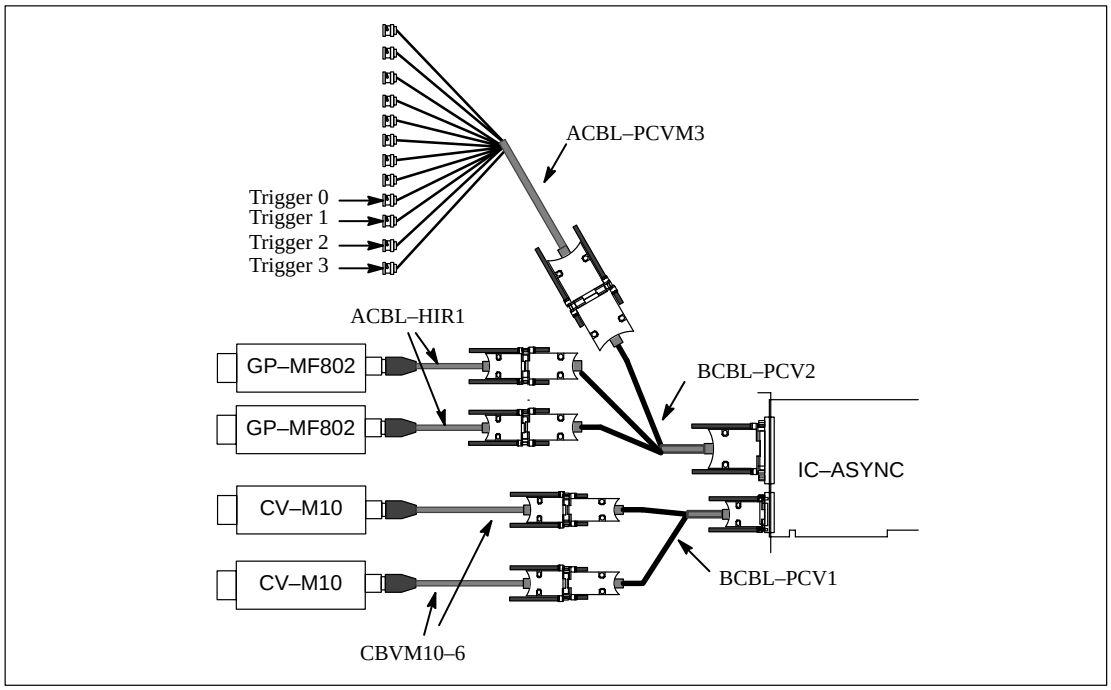


Figure 52. Camera Cable Configuration

No acquisition takes place until a trigger pulse is received. When a trigger pulse comes in, the camera is immediately driven by a frame reset forcing it to take a picture. Notice in Figure 50, where the frame reset pulse is output. On ports CAM0 and CAM1 frame reset is asserted on the 'frame reset' pin output. However, on ports CAM2 and CAM3 frame reset has been redirected to the VSYNC pin output because the Panasonic MF-802 requires its trigger input to be on its VSYNC line in. This change can be made in the application (as we'll see later) or set in the camera configuration file. Figure 53 shows the display output from this application.

In the initialization section of QuadVis.CPP, each channel is assigned an external trigger input (channel 0 -> trigger 0, etc.). Frame reset is also enabled on each channel. In the file procthrd.CPP, external trigger gets enabled for all 4 channels. QuadVisView.CPP is where display windows are created for each channel. The following lines of code were added to the AppWizard pre-generated code.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

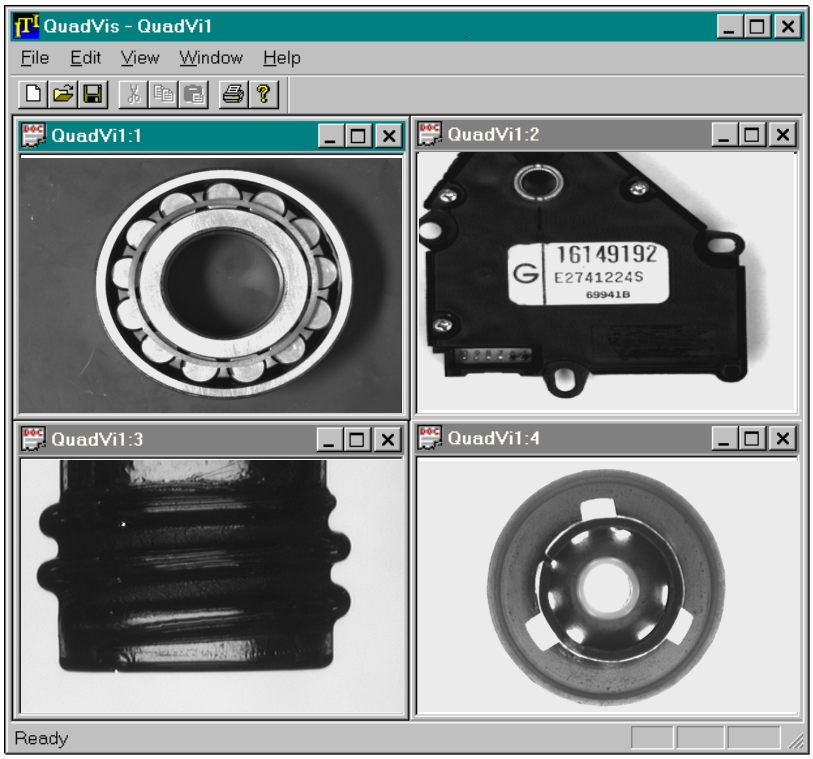


Figure 53. Four Channel Acquisition and Display

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Added Code

In the initialization section of QuadVis.CPP, each channel is assigned an external trigger input. Frame reset is enabled on each channel.

Code Added to QuadVis.CPP

```
BOOL CQuadVisApp::InitalizeITEX()
{
// following code is a standard initialization of the
// boards and will appear in all applications using this hardware
. . .
    //return FALSE if no board was found
    if(m_iBoardCount==0){
        ::MessageBox(NULL, "Can't find ITI board", "Error", MB_OK | MB_ICONSTOP);
        return FALSE;
    }
    // Added code to support 4 channel acquisition
    // Configure external trigger inputs
    // ch0 - trigger0                ch2 - trigger2
    // ch1 - trigger1                ch3 - trigger3
    itx_set_param(m_pVMod[0][0], TRIGGER_INPUT, TRIGGER_EXT_INPUT0);
    itx_set_param(m_pVMod[0][1], TRIGGER_INPUT, TRIGGER_EXT_INPUT1);
    itx_set_param(m_pVMod[0][2], TRIGGER_INPUT, TRIGGER_EXT_INPUT2);
    itx_set_param(m_pVMod[0][3], TRIGGER_INPUT, TRIGGER_EXT_INPUT3);

    // Enable frame reset output on all 4 channels
    itx_set_param(m_pVMod[0][0], FRAME_RESET, FRAME_RESET_ENABLED);
    itx_set_param(m_pVMod[0][1], FRAME_RESET, FRAME_RESET_ENABLED);
    itx_set_param(m_pVMod[0][2], FRAME_RESET, FRAME_RESET_ENABLED);
    itx_set_param(m_pVMod[0][3], FRAME_RESET, FRAME_RESET_ENABLED);
    return TRUE;
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

In the file procthrd.CPP, external trigger gets enabled for all 4 channels.

Code Added to procthrd.CPP

```
// This is the image processing and display thread
DWORD WINAPI ProcessThread(LPVOID pOwner)
{
    CProcessThread *pParent = (CProcessThread *)pOwner;
. . .
    // set board in trigger mode; grab into host circular buffer
    // Added code to support 4 channel acquisition
    itx_set_param(pView->m_pViewBoardMod, TRIGGER_SOURCE, EXTERNAL_TRIGGER);
    itx_set_param(pView->m_pViewBoardMod, TRIGGER_ENABLE, ENABLE_TRIGGER);
```

Select external trigger source for each channel, and enable it.

```
    hGrabID = itx_host_grab(pView->m_pViewBoardMod, pView->m_pbyHostBuffer,
FRAMES_IN_HOSTBUFF);
    while (!pParent->m_bShutdown) {
        // Wait for a new frame to become available, and lock it
        // so it can't be overwritten while we are processing and displaying it
. . .
```

QuadVisView.CPP is where display windows are created for each channel.

Code Added to QuadVisView.CPP

```
////////////////////////////////////////////////////////
// CQuadVisView construction/destruction
CQuadVisView::CQuadVisView()
{
    //initialize display related variables
    CQuadVisApp *theapp=(CQuadVisApp*)AfxGetApp();
    m_iViewBoard = theapp->m_iActiveBoard;
    // Added code to support 4 channel acquisition
    m_iViewVMod = (theapp->m_childCount++)%4;
```

Associates channel to a display window

```
. . .
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Code Added to QuadVis.h

```
////////////////////////////////////////////////
// CQuadVisApp:
// See QuadVis.cpp for the implementation of this class
//
class CQuadVisApp : public CWinApp
{
public:
   // Added code to support 4 channel acquisition
   DWORD m_childCount;
   CQuadVisApp();
   ~CQuadVisApp();
   // ITI hardware related routines
   BOOL InitalizeITEX();
. . .
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

APPENDIX A
VIDEO CONNECTIONS

CONNECTORS

IC-ASYNC provides three connections to the outside world. On the I/O plate there are two D-Sub connectors (15-pin and 26-pin) for camera and miscellaneous connections. At the top edge of the board is a dual row post header connector which interfaces through a ribbon cable to another I/O plate in one of the spare chassis slots. This provides parallel I/O capability in the form of 8 input and 8 output, high-drive TTL channels. The connector on the "auxiliary" I/O plate is a standard 25-pin D-Sub. There are two video connectors, each with support for 2 cameras.

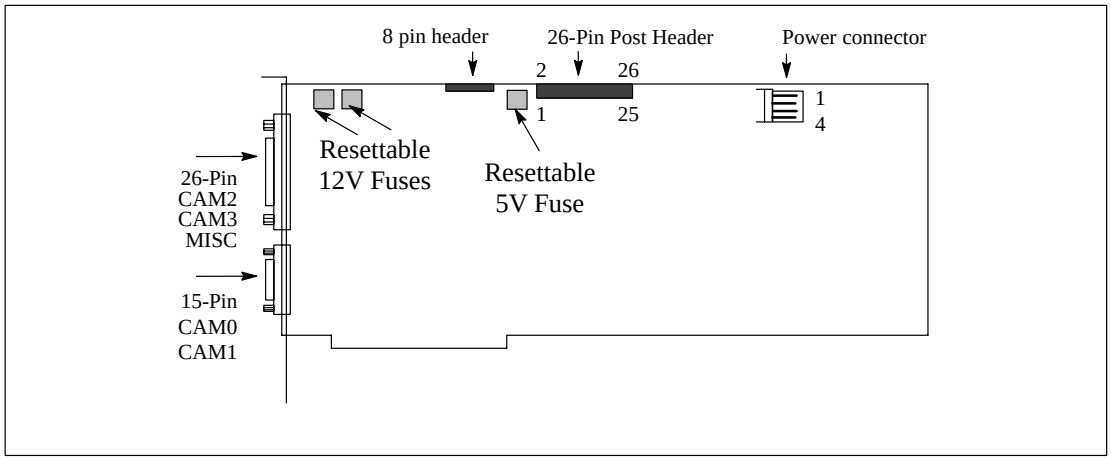


Figure A-1. IC-ASYNC Board and Connectors

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

15-Pin Connector CAM0 and CAM1

The 15-pin connector is pin compatible with standard camera adapter cables for single camera interfacing. A special breakout cable is required to split this into two independent camera connectors. Each camera D-Sub supplies up to 1A of 12V power.

Table A-1. 15-Pin Connector

Pin #	*Signal name*	*Description*
1	CAM1	Video Input - camera 1
2	CAM0	Video input - camera 0
3	n/c	no connection
4	DGND1	Digital ground - camera 1
5	DGND0	Digital ground - camera 0
6	AGND1	Analog ground - camera 1
7	AGND0	Analog ground - camera 0
8	FRESET1	Frame reset - camera 1
9	FRESET0	Frame reset - camera 0
10	+ 12V_GND	Power return
11	CHSYNC1	Composite/Horizontal Sync - camera 1
12	VSYNC1	Vertical Sync - camera 1
13	CHSYNC0	Composite/Horizontal Sync - camera 0
14	VSYNC0	Vertical Sync - camera 0
15	+ 12V	Power

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

26-Pin Connector CAM2, CAM3, and MISC

The second connector is a 26-pin D-Sub that provides additional signals for trigger, strobe and I/O. A special breakout cable will be required to split this into two independent camera connectors and a Miscellaneous connector. Each camera D-Sub supplies up to 1A of 12V power.

Table A-2. 26-Pin Connector

Pin #	*Signal name*	*Description*
1	CAM3	Video Input - camera 3
2	CAM2	Video input - camera 2
3	DGND	Digital ground
4	IN1	General purpose input 1
5	OUT1	General purpose output 1
6	STROBE0	Strobe output 0
7	STROBE1	Strobe output 1
8	STROBE2	Strobe output 2
9	STROBE3	Strobe output 3
10	AGND_3	Camera ground 3
11	AGND_2	Camera ground 2
12	FRESET3	Frame reset 3
13	FRESET2	Frame reset 2
14	+12V_GND	+12V ground
15	IN0	General purpose input 0
16	OUT0	General purpose output 0
17	TRIGGER0	Trigger input 0
18	TRIGGER1	Trigger input 1
19	CHSYNC3	Composite/Horizontal Sync - camera 3
20	VSYNC3	Vertical Sync - camera 3
21	CHSYNC2	Composite/Horizontal Sync - camera 2
22	VSYNC2	Vertical Sync - camera 2
23	+12V	+12 Volt power
24	DGND	Digital ground
25	TRIGGER2	Trigger input 2
26	TRIGGER3	Trigger input 3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Parallel I/O Connector

IC-ASYNC provides a digital I/O capability for controlling or monitoring external events. The digital input and output lines, which are available on a 26-pin header at the top of the PC card, can be cabled to a 25-pin D-Sub connector that occupies an open slot in the chassis.

Table A-3. 26-Pin Dual-Row Header

Header Pin #	*Signal name*	*Description*	*D-Sub Connector Pin #*
1	GND	Digital ground	1
3	GND	Digital ground	2
5	GND	Digital ground	3
7	GND	Digital ground	4
9	IN1	Digital Input pin 1	5
11	IN3	Digital Input pin 3	6
13	IN5	Digital Input pin 5	7
15	IN7	Digital Input pin 7	8
17	OUT0	Digital Output pin 0	9
19	OUT2	Digital Output pin 2	10
21	OUT4	Digital Output pin 4	11
23	OUT6	Digital Output pin 6	12
25	+5V	+5 V power output	13
2	STROBE_O	Strobe Output	14
4	STROBE_I	Strobe Input	15
6	I/O_INT	Interrupt Input	16
8	IN0	Digital Input pin 0	17
10	IN2	Digital Input pin 2	18
12	IN4	Digital Input pin 4	19
14	IN6	Digital Input pin 6	20
16	+5V	+5 V power output	21
18	OUT1	Digital Output pin 1	22
20	OUT3	Digital Output pin 3	23
22	OUT5	Digital Output pin 5	24
24	OUT7	Digital Output pin 7	25
26	n/c	no connection	n/a

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

BREAK-OUT CABLES

There are five standard cables available for camera interfacing and one to break-out the parallel I/O port to an unused PC slot. The cables and descriptions are as follows:

Table A-4. Cables for IC-ASYNC

Order Code	*Label #*	*Description*
BCBL-PCV1	19-4814	Breakout cable 1. Converts IC-ASYNC 15-pin D-Sub into two 15-pin D-Sub connectors for CAM0 and CAM1.
BCBL-PCV2	19-4815	Breakout cable 2. Converts IC-ASYNC 26-pin D-Sub into two 15-pin D-Sub connectors for CAM2 and CAM3 + one 26-pin D-Sub for miscellaneous.
ACBL-PCVM1	19-4811	Miscellaneous cable 1. Converts IC-ASYNC 26-pin D-Sub into 6 BNC connectors for trigger0, strobe0 and I/O (2-bit) interfacing.
ACBL-PCVM2	19-4812	Miscellaneous cable 2. Converts IC-ASYNC 26-pin D-Sub into 8 BNC connectors for trigger0/1, strobe0/1 and I/O (2-bit) interfacing.
ACBL-PCVM4	19-4813	Miscellaneous cable 4. Converts IC-ASYNC 26-pin D-Sub into 12 BNC connectors for trigger0-3, strobe0-3 and I/O (2-bit) interfacing.
PCBL-PCV	19-4816	Parallel I/O breakout. Converts IC-ASYNC 26-pin dual row header to a 25-pin female D-Sub.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

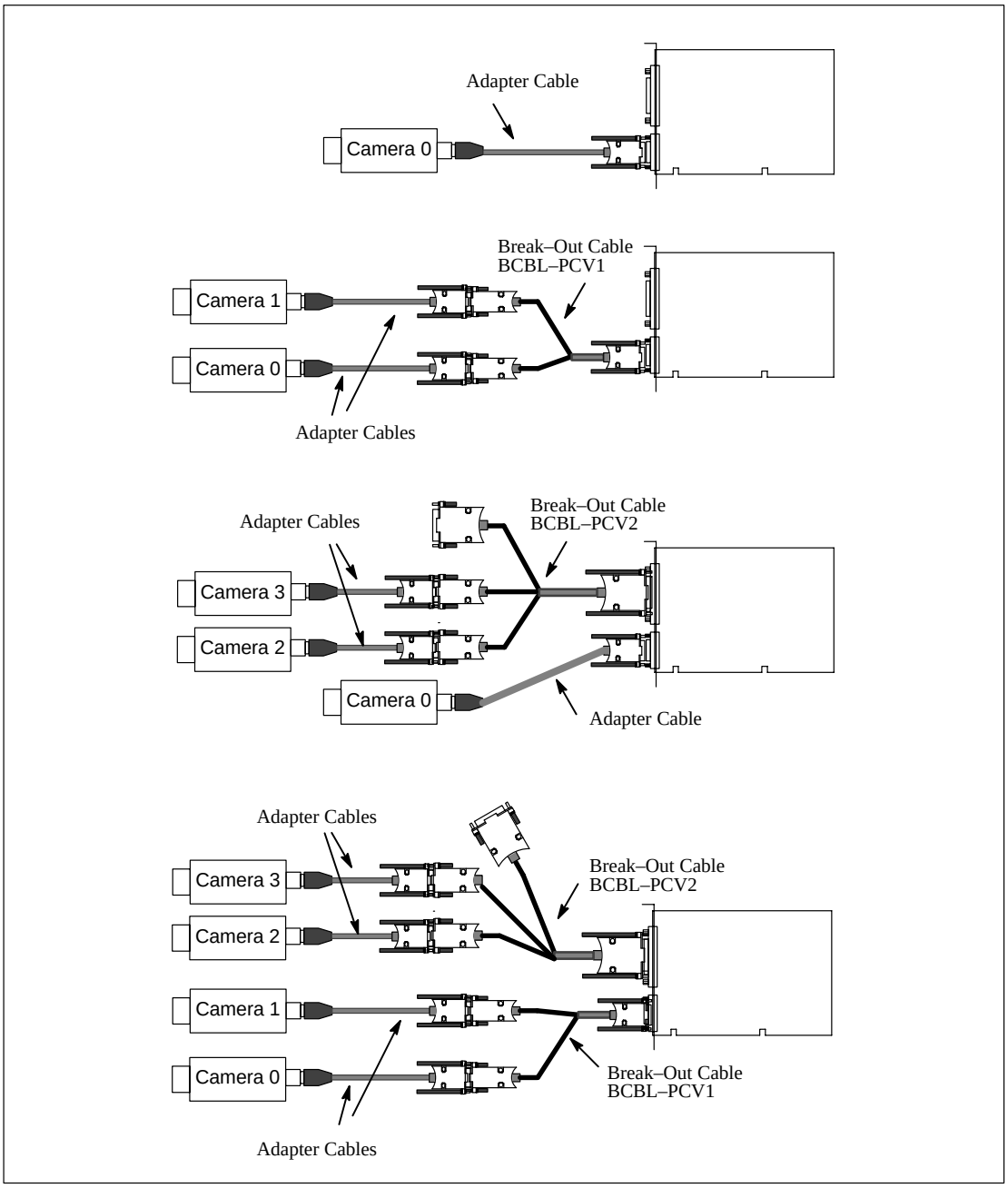


Figure A-2. Cameras and Cables

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-5. BCBL-PCV1 Pin-out

CAM0	CAM1	Signal Description	PCV15
2	–	CAM0 video input	2
7	–	Analog ground	7
9	–	FRESET0	9
5	–	Digital ground	5
13	–	C/HSYNC0	13
14	–	VSYNC0	14
15	15	+12 V power	15
10	10	12V return	10
–	2	CAM1 video input	1
–	7	Analog ground	6
–	9	FRESET1	8
–	5	Digital ground	4
–	13	C/HSYNC1	11
–	14	VSYNC1	12

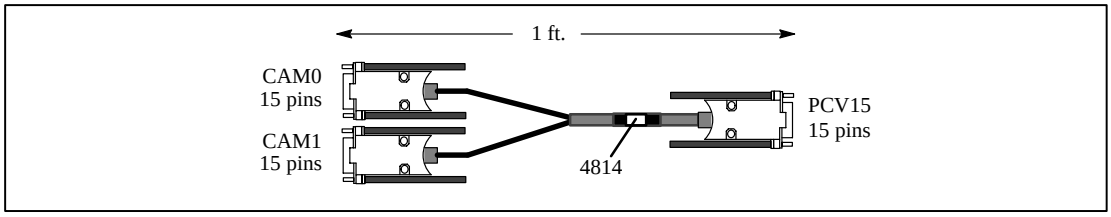


Figure A-3. BCBL-PCV1

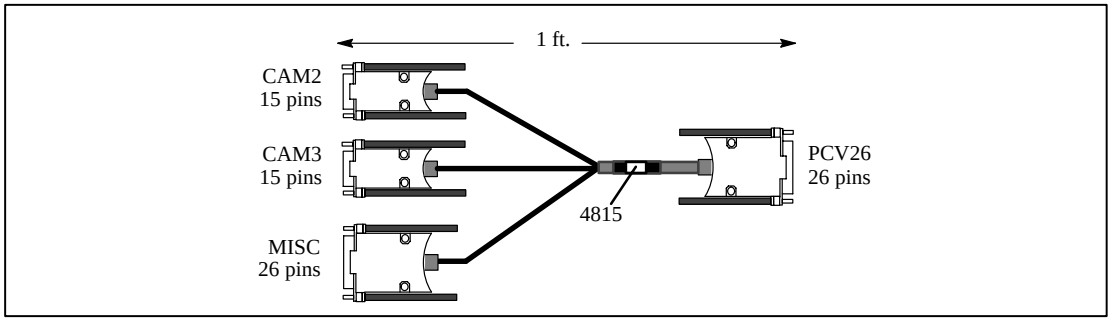


Figure A-4. BCBL-PCV2

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-6. BCBL-PCV2 Pin-out

CAM2	CAM3	MISC	Signal Description	PCV26
2	–	–	CAM2 video input	2
7	–	–	Analog ground	11
9	–	–	FRESET2	13
5	–	–	Digital ground	24
13	–	–	CHSYNC2	21
14	–	–	VSYNC2	22
15	15	–	+12V power	23
10	10	–	12V return	14
–	2	–	CAM3 video input	1
–	7	–	Analog ground	10
–	9	–	FRESET3	12
–	5	–	Digital ground	3
–	13	–	CHSYNC3	19
–	14	–	VSYNC3	20
		6	STROBE0	6
		7	STROBE1	7
		8	STROBE2	8
		9	STROBE3	9
		15	INPUT0	15
		16	OUTPUT0	13
		3	Digital ground	3
		17	TRIGGER0	17
		18	TRIGGER1	18
		25	TRIGGER2	25
		26	TRIGGER3	26
		4	INPUT1	4
		5	OUTPUT1	5
		24	Digital ground	24

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

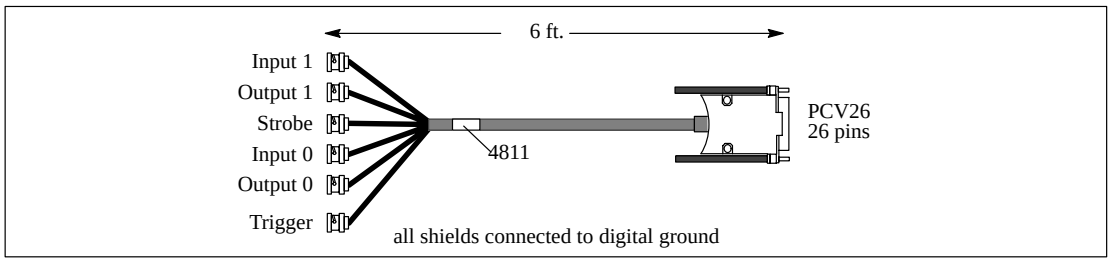


Figure A-5. ACBL-PCVM1

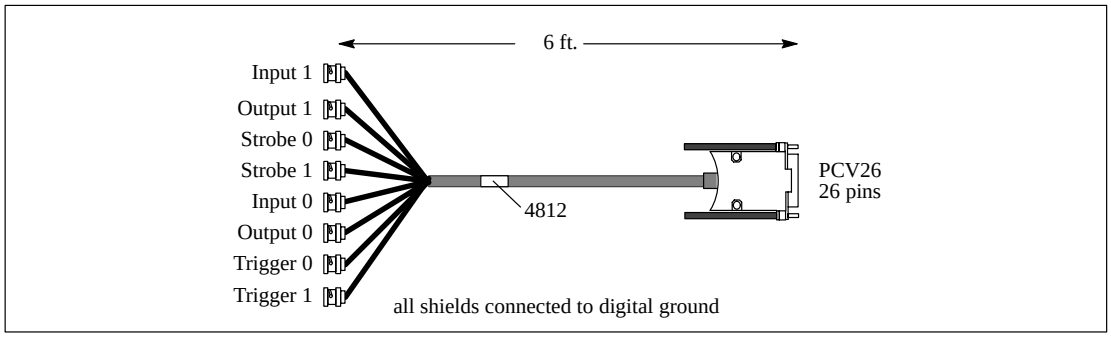


Figure A-6. ACBL-PCVM2

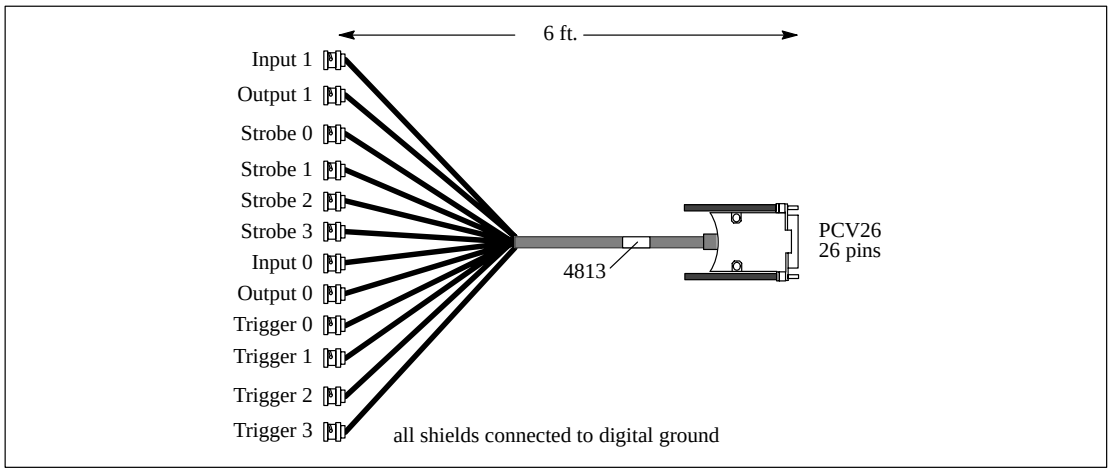


Figure A-7. ACBL-PCVM3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Camera Adapter Cables

Coreco Imaging, Inc has interfaced with many standard cameras and developed an extensive inventory of camera cables. Here is a sampling of the camera cables available for various camera models.

For a full listing of cameras and cables, refer to : http://www.imaging.com for the camera configuration guide at:
http://www.imaging.com/producttour/camguide/camguide.html

Manufacturer	*Model*	*Cable Number*	*Connectors*
various	—	ACBL–BNC	5 BNC
JAI	CV–M10E/C	CBVM10–6	12–pin Hirose & 6–pin Hirose
JAI	CV–M10BX	ACBL–VSCAN	12–pin Hirose
Sony	XC series	ACBL–HIR1	12–pin Hirose
Pulnix	TM series	ACBL–HIR2	12–pin Hirose
Sony	XC series, in	ACBL-DTAP26	12–pin Hirose
	dual–tap mode	ACBL-DTAP15	12–pin Hirose

BNC Camera Adaptor

This cable is compatible with general monochrome video cameras with BNC connectors. The ordering code for this cable is ACBL-BNC1. Table A-7 gives the pin-out, and Figure A-8 illustrates this cable. This cable is also used with the Cidtec 2250 and 3750 cameras in non-interlaced RS-170 compatible mode.

Table A-7. ACBL-BNC Cable Pin-Out

BNC Label	Signal Description	15-pin
CAMERA	Video input (single-ended)	2
”	Shield	7
MISC	FRESET, frame reset; VSCLOCK	9
”	Shield	5
HSYNC	Composite/horizontal sync; HDrive	13
”	Shield	5
VSYNC	Vertical sync; VDrive	14
”	Shield	5
CLOCK	Digital ground or no connection	4
”	Shield	5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

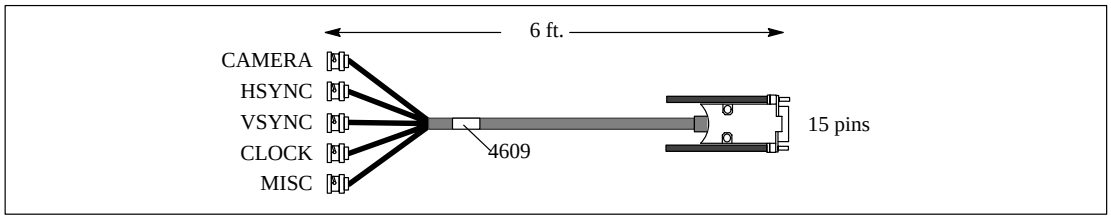


Figure A-8. ACBL-BNC Adaptor Cable

NOTE *The clock input is not used on the IC-ASYNC. Do not connect this cable to the camera; this cable may be connected to ground by the IC-ASYNC.*

Hirose (JIT) 1

This camera is compatible with the Sony XL series, and the Panasonic GPMF series analog monochrome cameras. The order code for this cable is ACBL-HIR1. Table A-8 gives the pin-out, and Figure A-9 illustrates this cable.

Table A-8. ACBL-HIR1 Cable Pin-Out

12-Pin	Signal Description	15-pin
1	+12 Volt return	10
2	+12 Volts DC	15
3	Analog ground	7
4	Video input (single-ended)	2
5	Digital ground	5
6	HDRIVE composite/horizontal sync	13
7	VDRIVE Vertical sync	14
8	Digital ground	5
9	Digital Ground or no connection	4
10	no connection	—
11	no connection	—
12	Digital ground	5

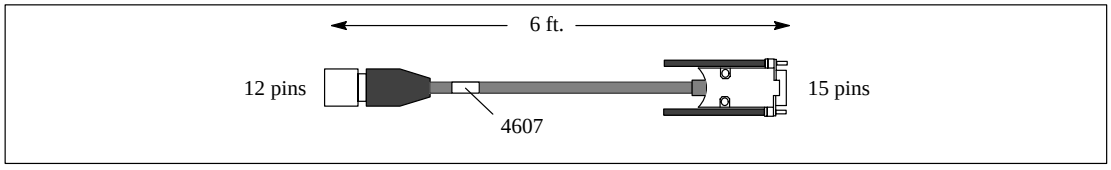


Figure A-9. ACBL-HIR1 Adaptor Cable

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Hirose (JIT) 2

This camera is compatible with the Pulnix TM Series analog monochrome cameras. The order code for this cable is ACBL-HIR2. Table A-9 gives the pin-out, and Figure A-10 illustrates this cable.

Table A-9. ACBL-HIR2 Cable Pin-Out

12-pin	Signal Description	15-pin
1	+12 Return	10
2	+12 Volts DC	15
3	Analog ground	7
4	Video input (single-ended)	2
5	(no connection)	—
6	(no connection)	—
7	VDRIVE vertical sync	14
8	Digital ground	5
9	HDRIVE composite/horizontal sync	13
10	(no connection)	—
11	(no connection)	—
12	Digital ground	5

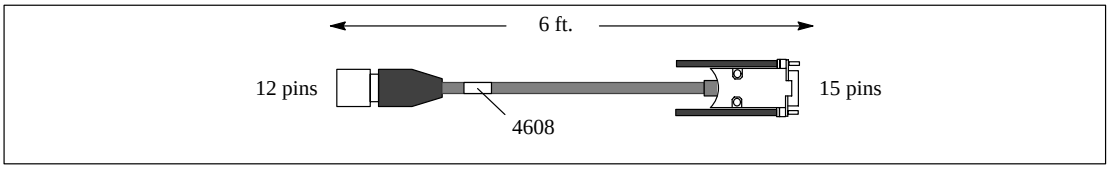


Figure A-10. ACBL-HIR2 Adaptor Cable

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CV-M10E/C

This cable is compatible with the JAI CV-M10E and CM-M10C Progressive Scan analog monochrome cameras. This cable supports these cameras in PLL and XTAL modes. The order code for this cable is CBVM10-6. Table A-10 gives the pin-out, and Figure A-11 illustrates this cable.

Table A-10. CBVM10-6 Cable Pin-Out

CVM10 trigger-in	CVM10 Video/sync	Signal Description	15-pin
1,2	-	(no connection)	-
3	-	Digital ground	5
4,6	-	(no connection)	-
5	-	CVM10 trigger input, Frame Reset not a trigger input to IC-ASYNC	9
	1	+12 Volt return	10
	2	+12 Volts DC	15
	3	Analog ground	7
	4	Video input (single-ended)	2
	5	Analog ground	6
	6	HDRIVE horizontal sync	13
	7	VDRIVE Vertical sync	14
	8	Analog ground	8
	9,10	(no connection)	-
	11,12	(no connection)	-

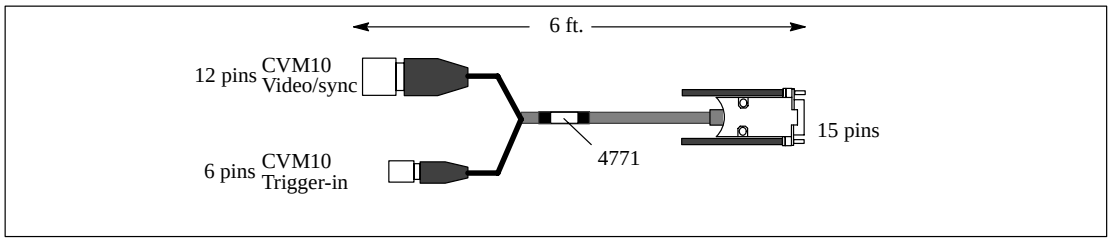


Figure A-11. CBVM10-6 Adaptor Cable

NOTE *The CMV10 "Trigger–in" connector carries the IC-ASYNC frame reset output, to trigger the Camera. This is not a trigger input to the IC-ASYNC.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CV-M10BX

This cable is compatible with the JAI CV-M10BX Progressive Scan analog monochrome camera. The BX version outputs a pixel clock, and is therefore suitable for Variable Scan mode. The order code for this cable is ACBL–VSCAN. Table A-11 gives the pin-out, and Figure A-12 illustrates this cable. IC-ASYNC must be programmed for variable scan mode for this camera.

Table A-11. ACBL–VSCAN Cable Pin-Out

12-Pin	Signal Description	15-pin
1	+12 Volt return	10
2	+12 Volts DC	15
3	Analog ground	7
4	Video input (single-ended)	2
5	Digital ground	5
6	HD or HDRIVE	13
7	WEN or VDRIVE	14
8	Digital ground	5
9	Clock	9
10	no connection	—
11	no connection	—
12	Digital ground	5

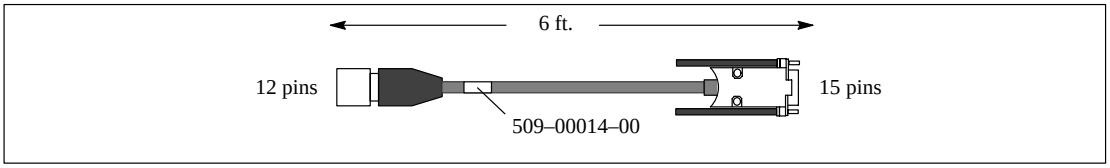


Figure A-12. ACBL–VSCAN Adaptor Cable

NOTE *The CV-M10BX should be triggered using the "Trigger–in" connector on the camera.*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Dual-Tap Cameras

There are two cables available for the Dual–Tap versions of the JAI CV-M10 and Sony XC7500 cameras. One cable is for the 15-Pin CAM0 and CAM1 connector, and the other cable is for the 26-Pin CAM2 and CAM3 connector. These two cables plug directly into the IC-ASYNC connectors, and do not use the break-out cables. Only one cable is needed for one camera in dual-tp mode.

Table A-12. ACBL–DTAP–15P Cable

12-Pin	Signal Description	15-pin
1	+12 Volt return	10
2	+12 Volts DC	15
3	Analog ground	7
4	Video 0	2
5	Digital ground	5
6	HDRIVE	13
7	VDRIVE	14
8	Analog ground	6
9	Video 1	1
10	+12 Volt return	10
11	+12 Volts DC	15
12	Digital ground	4

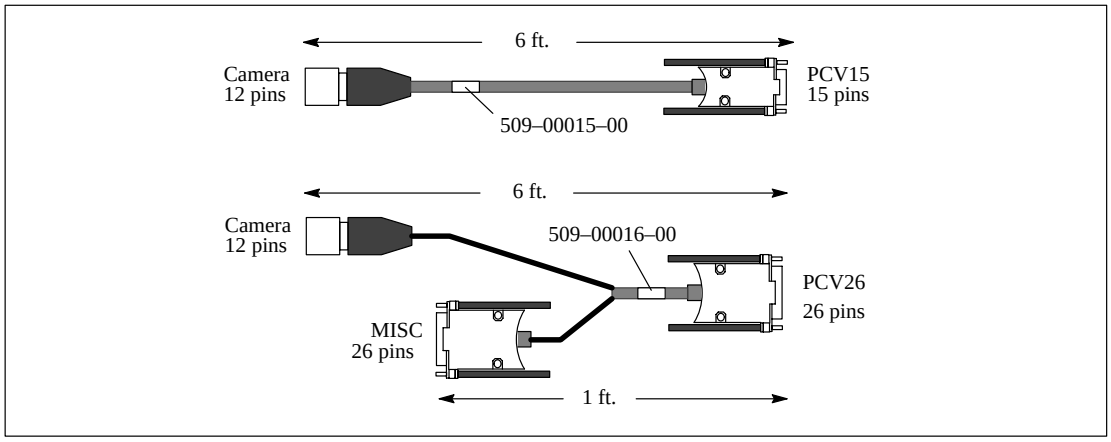


Figure A-13. Dual-Tap Camera Adapter Cables

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table A-13. ACBL–DTAP–26P Cable

Camera 12-Pin	Signal Description	PCV 26-pin
1	+12 Volt return	14
2	+12 Volts DC	23
3	Analog ground	11
4	Video 2	2
5	Digital ground	3
6	HDRIVE	21
7	VDRIVE	22
8	Analog ground	10
9	Video 3	1
10	+12 Volt return	14
11	+12 Volts DC	23
12	Digital ground	24

MISC. 26-Pin	Signal Description	PCV 26-pin
1,2	no connection	–
3	Digital ground	3
4	Input 1	4
5	Output 1	5
6	Strobe 0 output	6
7	Strobe 1 output	7
8	Strobe 2 output	8
9	Strobe 3 output	9
10, 11, 12	no connection	–
13, 14	no connection	–
15	Input 0	15
16	Output 0	16
17	Trigger 0 input	17
18	Trigger 1 input	18
19, 20, 21	no connection	–
22, 23	no connection	–
24	Digital ground	24
25	Trigger 2 input	25
26	Trigger 3 input	26

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP