EPIX®

XCLIB™ LIBRARY

Programming Library for PIXCI® Imaging Boards

$$\nabla^2 f\,(x,y) \equiv \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{d_2\text{-}f}{f}$$

```
for(;;)  {
  capture();
  difference();
  lowpass();
  if (findblobs())
    reportlargest();
}
```

$$\begin{bmatrix} \cos\varnothing & \sin\varnothing & 0 \\ -\sin\varnothing & \cos\varnothing & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pixel Clock:	**45MHz**
Pixel Jitter:	**1.0 ns**
Resolution:	**6144 x 4096**
Buffers:	**253**
Pixel Bits:	**8**
Camera:	**Area Scan**
Trigger:	**TTL**
Field Delay:	**5**

$$\frac{1}{d_1} + \frac{1}{d_2} = \frac{1}{f}$$

Blobs Found:	*259*
Center of Mass:	*(23.65, 78.78)*
Subpixel Edge @:	*(97.3498, 87.5000)*

$$\frac{2\lambda}{\pi n \omega_0}$$

$$\delta = 1.22\frac{\lambda d_i}{l} \qquad s.n.r. = \frac{\bar{x}}{\sqrt{\Sigma\,(x_i\text{-}\bar{x})^2/n}}$$

FEATURES

- **Quick & Easy Access to PIXCI® Imaging Boards.**
- **DLL for 32 bit Windows 95, 98, ME Applications.**
- **DLL for 32 bit Windows NT, 2000, XP, Vista, 7 Apps.**
- **DLL for 64 bit Windows XP, Vista, 7 Apps.**
- **Object Code for 32 bit & 64 bit Linux Programs.**
- **Object Code for 32 bit DOS Programs.**
- **High Level Functions eliminate Register Level Programming.**
- **Custom Area-Scan or Line-Scan Formats.**
- **RS-170, NTSC, or S-Video Formats.**
- **Analog or Digital Video Interfaces.**
- **CCIR, PAL, or S-Video Formats.**
- **Dual Buffer "Ping-Pong" Capture.**
- **Multi-Buffer FIFO Capture for Video to Disk.**
- **Image Sequence Capture.**
- **Triggered Image Capture.**
- **Image display on S/VGA.**

PIXCI® IMAGING POWER

The XCLIB Library empowers C/C++ and Windows, DOS, and Linux programmers to control the PIXCI® series of imaging boards. XCLIB supports all versions of the PIXCI® imaging boards:[1]

- The PIXCI® D†, D24†, D32†, D2X, and D3X for digital output area-scan and line-scan cameras with 8, 10, 12, 14, and other bit depths, in monochrome or color,
- The PIXCI® D2X or D2X1 for use with SILICON VIDEO® 1281M†, 1281C†, 1310†, 1310C†, 2112†, and 2112C† cameras.
- The PIXCI® SI, SI1, or SI4 for use with SILICON VIDEO® 5C10, 5M10, 642M, 642C, 643M, 643C, 9C10, 9M001, 9M001C, and 9T001C cameras.
- The PIXCI® CL1, CL2, CL3SD, E1†, E1DB†, E4, E4DB, EB1, EB1-PoCL, EC1, ECB1, ECB1-34, ECB2, EL1, and EL1DB, for Camera Link area-scan and line-scan cameras.
- The PIXCI® A† for analog output area-scan and line-scan cameras with 8, 10, 12, and other bit depths, in monochrome or color, and
- The PIXCI® SV2†, SV3†, SV4†, SV5, SV5A, and SV5B for NTSC, PAL, RS-170, CCIR, and S-Video.

Multiple PIXCI® frame grabbers can be operated selectively or simultaneously, allowing parallel control for multi-camera viewing of a single event or object, or allowing selective control for capturing multiple, independent, events or objects.

Through the unique flexibility of XCLIB and the PIXCI® imaging boards, single frames or video sequences from standard RS-170, NTSC, PAL, or CCIR or other nonstandard video sources can be captured, analyzed[2], displayed, and archived.[3]

CONVENIENCE & COMPATIBILITY

XCLIB provides high level services, allowing concentration on the imaging application, rather than focusing on board level programming.

XCLIB automatically detects and manages all versions of the PIXCI® imaging boards and their intended cameras. Fundamental services, such as setting resolution, capturing, and accessing image data, are the same regardless of camera or version of imaging board.

The XCLIB C/C++ library allows embedding control of PIXCI® imaging boards into user-written applications. Under Windows, the XCLIB DLL also enables access from existing Windows applications, such as Visual Basic, Delphi, LabView, Matlab, and many others, which support "calling" into DLLs.

SOPHISTICATED SOLUTIONS

XCLIB allows user-written programs to apply the PIXCI® imaging boards to applications such as image enhancement, archival, analysis and measurement; event and motion study; document capture; cine-loop and ultrasound image capture; particle analysis; visual inspection; machine vision and quality control. Join the scientists and engineers in medical, industrial, and research environments who rely upon EPIX® imaging solutions.

IMAGE PROCESSING PRODUCTS FOR RESEARCH AND INDUSTRY

FEATURES

Flexible Video Formats - Capture standard monochrome RS-170 and CCIR formats, composite color NTSC and PAL formats, color S-Video formats, as well as other nonstandard video formats.[4]

Flexible Video Modes - Set the number of pixels per line, subsampling of pixels in the line, lines per field, and subsampling of lines within the field.[5] Capture interlaced video as a single interlaced image; as two images, one per field; or as a single image of the odd or even field. Supports video rate control of capture into frame buffers.

Tradeoff capture resolution against number of frame buffers. With 4 Mbyte image memory, for example, resolution of 512x240 at 8 bits per pixel yields 34 frame buffers, 752x480 yields 11 frame buffers, and 2048x1020 yields 2 frame buffers.

Extended Features - Built-in functions provide easy import and export of images in TIF and BMP file formats, and exporting in PCX and TGA formats.

Under Windows: Built-in services provide display of captured or modified images on the S/VGA, in all or part of, one or more windows.

Image Access - Read and write frame buffers with efficient block read and write. Read and write pixels sequentially within area of interest, without concern for AOI boundaries or line interlacing. Direct access to image memory with "C" pointer.

For color cameras, access pixels as grey levels, RGB, or HSB (Hue, Saturation, Brightness), regardless of the camera's or PIXCI® imaging board's native color space. Pixels may also be accessed in the board's native color space (YCrCb for the PIXCI® SV2, SV3, SV4, SV5, SV5A, SV5B), or the camera's native color space (such as raw Bayer data for Bayer output color cameras).

Camera & Device Control - Sense general purpose input signals, control general purpose output signals.[6] Obtain elapsed field/frame count.

Capture sequences of images at video rate, or sub-video rate. Capture images in response to external trigger.

Dual Buffer 'Ping-Pong' Capture - Capture at video rate into alternate buffers, allowing analysis of one buffer while capturing into the other.

Easy Programming - All basic features are available via Simple C Functions (SCF); easy to use subroutines don't require familiarity with "C" structures, pointers, etc. An example source program demonstrates use of SCF subroutines. Additionally, an extended "structured" and "object oriented" interface provides complete control of video formats, resolutions, features and options.

High Level Services - Image frame buffer access provides pixel data, interlaced if appropriate, in correct "optical order", and independent of the camera's internal scanning or multi-channel configuration.

All XCLIB video services support "no-wait" and "check for completion" modes. During frame capture the host computer is not busy transferring pixel data nor waiting for the next vertical blanking interval, but can proceed with concurrent processing.

Sequence Capture - Capture sequences of images at video rate, or sub-video rate, to the extent of available host computer memory. Under Windows 95 or 98, a special "Memory Reservoir" feature allows reserving 100+ MBytes of host computer memory for image frame buffers. Under Windows NT, 2000, XP, Vista, or 7 a special "Forceful Allocation" technique allows reserving almost all computer memory (up to 4 GByte on 32 bit machines) for image frame buffers, excepting the 64 to 512+ MBytes required by Windows and BIOS.

Multi-Buffer FIFO Capture for Video to Disk - Capture at video rate into a circular queue of buffers, allowing image analysis or disk I/O to lag behind video capture.

Direct Live Video - For the PIXCI® SV2, SV3, SV4, SV5, SV5A, SV5B: With a suitable S/VGA card and Windows DirectDraw support, built-in services allow displaying live video with overlays, transferring pixel data directly to the S/VGA in all or part of a window,[7] without the programmer using the DirectDraw API/SDK.

```
  pxd_PIXCIopen("", "NTSC", "");             // Use standard NTSC, or
//pxd_PIXCIopen("", "CCIR", "");             // .. CCIR, or
//pxd_PIXCIopen("", "", "VIDEO.FMT");        // .. customized format?

pxd_doSnap(0x1, 1, 0);                        // and wait for result

// Access monochromatic data
unsigned char   buf[1024];                    // size >= # pixels per AOI line
unsigned long   c = 0;
int       i, j, y;
for (y = 0;; y++) {
    i = pxd_readuchar(0x1, 1, 0, y, -1, y+1, buf, sizeof(buf), "Grey");
    if (i <= 0)                               // read next line of image
        break;                                // until there are no more
    for (j = 0; j < i; j++)                   // scan and ..
        if (buf[j]  < 64)                     // .. test and count pixels
            c++;                              // .. with value less than 64
}

// Access one line of RGB data
pxd_readuchar(0x1, 1, 0, 0, -1, 1, buf, sizeof(buf), "RGB");
printf("Pixel at (0,20) is %d/%d/%d\n", buf[20+0], buf[20+1], buf[20+2];

// Access one line of Red component of RGB data
i = pxd_readuchar(0x1, 1, 0, 0, -1, 1, buf, sizeof(buf), "RofRGB");
for (j = 0, c = 0; j < i; j++)
    c += buf[j];
printf("Red component average is %f\n", (double)count/i);
```

Frame buffer access examples

XCLIB TOPICS AND FUNCTIONS

Installation, Configuration, and Usage — Compile Time.

Introduction.
Library Installation.
Compiling.
Stack.
Heap.
Memory Allocation Hooks.
Compile Time Inclusions.
Common Types and Concepts.
Types.
Modifiers.
Call Back Types.
Usage Assumptions.
On Frame Buffers and Memory.
On the Semantics of Snap.
On Imaging Boards with Multiple Functional Units.
On Multiple Imaging Boards.
On Multiple Identical Imaging Boards.
On Serial Ports.
On Video Format Configuration.
On Video State ID's.
On Error Codes.
On Structures and Signatures.
Common Structures.
Coordinates, Images, and Image Window Structures.
Imaging Board, Driver, and Video Format Configuration Structures.

Installation and Configuration — Run Time.

Driver Configuration Parameters.
Windows 95, 98, ME Environment.
Windows NT, 2000, XP, XP(x64), Vista, Vista(x64), 7, 7(x64) Environment.
Tenberry DOS4GW DOS Extender Environment.
Linux 32 Bit & Intel i386 Environment.
Linux 64 Bit & Intel x86-64 Environment.

SCF Style Interface.

Buffer's Capture Field Count ‡.
Buffer's Capture General Purpose Input ‡.
Buffer's Capture System Time ‡.
Last Captured Buffer: Buffer Number ‡.
Last Captured Buffer: Field Count ‡.
Last Captured Buffer: System Time ‡.
Camera Link CC Output Signal: Set ‡.
Camera Link CC Output Signal: Get ‡.
Get PXIMAGE: Access Imaging Board Frame Buffer ‡.
Get PXIMAGE3: Access Imaging Board Frame Buffers ‡.
Get PXIMAGE: Release Access to Imaging Board Frame Buffers ‡.
Get PXIMAGE3: Release Access to Imaging Board Frame Buffers ‡.
Video Snap and Wait ‡.
Windows: Register Event upon Video Field ‡.
Windows: Unregister Event upon Video Field ‡.
Windows: Register Event upon Captured Video Field ‡.
Windows: Unregister Event upon Captured Video Field ‡.
Windows: Register Event upon General Purpose Trigger ‡.
Windows: Unregister Event upon General Purpose Trigger ‡.
Windows: Register Event upon Fault ‡.
Windows: Unregister Event upon Fault ‡.
DOS: Register Callback upon Video Field ‡.
DOS: Unregister Callback upon Video Field ‡.
DOS: Register Callback upon Captured Video Field ‡.
DOS: Unregister Callback upon Captured Video Field ‡.
DOS: Register Callback upon General Purpose Trigger ‡.
DOS: Unregister Callback upon General Purpose Trigger ‡.
DOS: Register Callback upon Fault ‡.
DOS: Unregister Callback upon Fault ‡.
Linux: Register Signal upon Video Field ‡.
Linux: Unregister Signal upon Video Field ‡.
Linux: Register Signal upon Captured Video Field ‡.
Linux: Unregister Signal upon Captured Video Field ‡.
Linux: Register Signal upon General Purpose Trigger ‡.
Linux: Unregister Signal upon General Purpose Trigger ‡.
Linux: Register Signal upon Fault ‡.
Linux: Unregister Signal upon Fault ‡.
General Purpose Input Signal: Sense ‡.
General Purpose Input Signal: Reset ‡.
General Purpose Output Signal: Set ‡.
General Purpose Output Signal: Get ‡.
General Purpose Trigger Count: Sense ‡.
Video Snap ‡.
Video Snap Pair ‡.
Video Live ‡.
Video Live Pair Alternate ‡.
Video Live Sequence ‡.
Video UnLive ‡.
Video Abort ‡.
Video Live Trigger ‡.
Video Live Sequence Trigger ‡.
Video Initiated Status Check ‡.
Get Frame Buffer Image Aspect Ratio ‡.
Get Frame Buffer Image Bit Depth ‡.
Get Frame Buffer Image Color Components ‡.
Get Frame Buffer Image Field Dimension ‡.
Get Frame Buffer Image Horizontal Dimension ‡.
Get Frame Buffer Image Vertical Dimension ‡.
Get Frame Buffer Image Count ‡.
Get Imaging Board Driver ID ‡.
Get Imaging Board Include Files ID ‡.
Get Imaging Board Library ID ‡.
Get Imaging Board's Frame Buffer Size ‡.
Get Imaging Board's Model ‡.
Get Imaging Board's Submodel ‡.
Get Imaging Board Unit Count ‡.
Load Image: BMP Format ‡.
Load Image: TIFF Format ‡.
Save Image: BMP Format ‡.
Save Image: PCX Format ‡.
Save Image: TARGA Format ‡.
Save Image: TIFF Format ‡.
Load Frame Buffers ‡.
Save Frame Buffers ‡.
Errors: Translate Error Code to String ‡.
Errors: Check & Report Faults ‡.
PIXCI$^{(R)}$ Imaging Board: Open and Set Video Format Configuration ‡.
PIXCI$^{(R)}$ Imaging Board: Close ‡.
Read Pixel Values as Unsigned Char ‡.
Read Pixel Values as Unsigned Short ‡.
Write Pixel Values as Unsigned Char ‡.
Write Pixel Values as Unsigned Short ‡.
Windows: Create Device Independent Bitmap (DIB) ‡.
Windows: Release Device Independent Bitmap (DIB) ‡.
Windows: Direct Video Display, Init ‡.
Windows: Direct Video Display, Done ‡.
Windows: Direct Video Display, Start ‡.
Windows: Direct Video Display, Stop ‡.
Windows: Display Image Frame Buffer On Device Context ‡.
Serial Port: Set Configuration ‡.
Serial Port: Read ‡.
Serial Port: Write ‡.
Camera Link Serial: Initialize ‡.
Camera Link Serial: Close ‡.
Camera Link Serial: Read ‡.
Camera Link Serial: Write ‡.
Camera Link Serial: Read Available ‡.
Camera Link Serial: Read Flush ‡.
Camera Link Serial: Error Text ‡.
Camera Link Serial: Ports ‡.
Camera Link Serial: Ports ID ‡.
Camera Link Serial: Manufacturer ID ‡.
Camera Link Serial: Get Baud Rates ‡.
Camera Link Serial: Set Baud Rate ‡.
Set EXSYNC and PRIN Values.
Get EXSYNC Counter Value.
Get PRIN Counter Value.
Set EXSYNC and PRINC Mode Bits.
Get EXSYNC Mode Bits.
Get PRINC Mode Bits.
Set Video Input Multiplexer ‡.
Set Video Contrast and Brightness ‡.
Set Video Hue and Saturation ‡.
Get Video Input Multiplexer ‡.
Get Video Contrast ‡.
Get Video Brightness ‡.
Get Video Hue ‡.
Get Video U Saturation ‡.
Get Video V Saturation ‡.
Set Bright White Balance ‡.
Set Dark White Balance ‡.
Get Video Field Count ‡.
Get Video Field Characteristic ‡.
Get Video Field Characteristic ‡.
Set Video Configuration as per Compiled Include.
Set SV2112 Exposure and Gain ‡.
Get SV2112 Exposure ‡.
Get SV2112 Gain ‡.
Set SV2112 Resolution and Timing ‡.
Get SV2112 Decimation ‡.
Get SV2112 Aoi Top Edge ‡.
Get SV2112 Aoi Left Edge ‡.
Get SV2112 Pixel Clock Frequency ‡.
Get SV2112 Scan Direction ‡.
Set SV2112 Video And Trigger Modes ‡.
Set SV2112 Controlled Exposure And Frame Rate ‡.
Get SV2112 Video Mode ‡.
Get SV2112 Controlled Video Mode ‡.
Get SV2112 Controlled Trigger Mode ‡.
Get SV2112 Controlled Frame Rate ‡.
Get SV2112 Controlled Exposure ‡.
Set SV1310 Exposure, Gain, and Offset ‡.
Get SV1310 Exposure ‡.
Get SV1310 Gain ‡.
Get SV1310 Offset ‡.
Set SV1310 Color Specific Gain ‡.
Get SV1310 Color Specific Gain ‡.
Set SV1310 Resolution and Timing ‡.
Get SV1310 Subsample ‡.
Get SV1310 Aoi Top Edge ‡.
Get SV1310 Aoi Left Edge ‡.
Get SV1310 Pixel Clock Frequency ‡.
Get SV1310 Readout Direction ‡.
Get SV1310 Frame Period ‡.
Set SV1310 Video And Trigger Modes ‡.
Set SV1310 Controlled Frame Rate ‡.
Get SV1310 Video Mode ‡.
Get SV1310 Controlled Video Mode ‡.
Get SV1310 Controlled Trigger Mode ‡.
Get SV1310 Controlled Frame Rate ‡.
Get SV1310 Controlled Exposure ‡.
Set SV1281 Exposure, Gain, and Offset ‡.
Get SV1281 Exposure ‡.
Get SV1281 Gain ‡.
Get SV1281 Offset ‡.
Set SV1281 Resolution and Timing ‡.
Get SV1281 Aoi Top Edge ‡.
Get SV1281 Aoi Left Edge ‡.
Get SV1281 Pixel Clock Frequency ‡.
Set SV1281 Video And Trigger Modes ‡.
Set SV1281 Controlled Frame Rate ‡.
Get SV1281 Video Mode ‡.
Get SV1281 Controlled Video Mode ‡.
Get SV1281 Controlled Trigger Mode ‡.
Get SV1281 Controlled Frame Rate ‡.
Set SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Exposure and Gain.
Set SV9T001/SV9C10 Exposure and Digital Gain ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Exposure ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Gain ‡.
Get SV9T001/SV9C10 Digital Gain ‡.
Set SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Resolution and Timing ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Aoi Top Edge ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Aoi Left Edge ‡.
Get SV9M001/SV5C10/SV5M10/SV9C10 Scan Direction ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Subsample ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Pixel Clock Frequency ‡.
Set SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Video And Trigger Modes ‡.
Set SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Controlled Exposure And Frame Rate ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Video Mode ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Controlled Video Mode ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Controlled Trigger Mode ‡.
Get SV9M001/SV9T001/SV5C10/SV5M10/SV9C10 Controlled Frame Rate ‡.
Set SV642 Exposure and Gain ‡.
Get SV642 Exposure ‡.
Get SV642 Gain ‡.
Get SV642 Gain Range ‡.
Set SV642 Resolution and Timing ‡.
Get SV642 Aoi Top Edge ‡.
Get SV642 Aoi Left Edge ‡.
Get SV642 Pixel Clock Frequency ‡.
Set SV642 Video And Trigger Modes ‡.
Set SV642 Controlled Exposure And Frame Rate ‡.
Get SV642 Video Mode ‡.
Get SV642 Controlled Video Mode ‡.
Get SV642 Controlled Trigger Mode ‡.
Get SV642 Controlled Frame Rate ‡.
Set SV643 Exposure and Gain ‡.
Get SV643 Exposure ‡.
Get SV643 Gain ‡.
Set SV643 Resolution and Timing ‡.
Get SV643 Aoi Top Edge ‡.
Get SV643 Aoi Left Edge ‡.
Get SV643 Pixel Clock Frequency ‡.
Set SV643 Video And Trigger Modes ‡.
Set SV643 Controlled Exposure And Frame Rate ‡.
Get SV643 Video Mode ‡.
Get SV643 Controlled Video Mode ‡.
Get SV643 Controlled Trigger Mode ‡.
Get SV643 Controlled Frame Rate ‡.
Get SV643 Controlled Exposure ‡.

Structured Style Interface.

PIXCI$^{(R)}$ Imaging Board: Structured Open.
PIXCI$^{(R)}$ Imaging Board: Structured Close.
Get Imaging Board and Driver Information.
Get Extended Imaging Board and Driver Information.
Obtain Last Logged Fault.
Get Driver Parameters.
Set Driver Parameters.
Allocate Copy of Video State.
Free Copy of Video State.
Compare Pair of Video State Copies.
Define Video State.
Delete Video State.
Get Video State.
Export Copy of Video State.
Import Copy of Video State.
Fixup Video State.
Fill Video State.
Sign Video State.
Set Default Configuration Copy of Video State.
Declare Instance of Video Format Configuration Structures.
Initialize Instance of Video Format Configuration Structures.
Obtain Pointer into Imaging Board Memory.
Free Pointer into Imaging Board Memory.
Obtain pximage Access into Imaging Board Memory.
Obtain pximage3 Access into Imaging Board Memory.

XCLIB TOPICS AND FUNCTIONS (Continued)

Obtain Filtered pximage Access into Imaging Board Memory.
Obtain Filtered pximage3 Access into Imaging Board Memory.
Read/Write Imaging Board Memory.
Read/Write Imaging Board Memory, Sparsely.
Get Live/Snap Status.
Get Current Video Status.
Get Extended Video Status.
Get Current Video Status Value.
Get Current Buffer Status.
Initiate Snap into Frame Buffer.
Initiate Snap Pair into Frame Buffers.
Initiate Continuous Capture into Frame Buffer.
Initiate Continuous Alternate Capture into Frame Buffers.
Initiate Continuous Capture into Memory Space.
Initiate Sequence Capture into Frame Buffers.
Initiate Triggered Capture into Frame Buffer.
Terminate Live/Snap into Frame Buffer.
Abort Live/Snap into Frame Buffer.
Set Camera Configuration.
Set Video Configuration.
Adjust Video Configuration.
General Purpose Input Signal: Sense.
General Purpose Input Signal: Reset.
General Purpose Output Signal: Set.
General Purpose Output Signal: Get.
Device Control.
Load Frame Buffers as Stream.
Save Frame Buffers as Stream.
Errors: Translate Error Code to String.
Errors: Check & Report Faults.
Load Image: BMP Format.
Load Image: TIFF Format.
Save Image: BMP Format.
Save Image: PCX Format.
Save Image: TARGA Format.
Save Image: TIFF Format.
Read Pixel Values as Unsigned Char.
Read Pixel Values as Unsigned Short.
Read Pixel Values as Unsigned Int.
Write Pixel Values as Unsigned Char.
Write Pixel Values as Unsigned Short.
Write Pixel Values as Unsigned Int.
Windows: Make Device Independent Bitmap.
Windows: Free Device Independent Bitmap.
Windows: Display Image Frame Buffer On Device Context.
Camera Control: SILICON VIDEO$^{(R)}$ 2112.
Camera Control: SILICON VIDEO$^{(R)}$ 1310.
Camera Control: SILICON VIDEO$^{(R)}$ 1281.
Camera Control: SILICON VIDEO$^{(R)}$ 9M001.
Camera Control: SILICON VIDEO$^{(R)}$ 9T001.
Camera Control: SILICON VIDEO$^{(R)}$ 5M10/5C10.
Camera Control: SILICON VIDEO$^{(R)}$ 9C10.
Camera Control: SILICON VIDEO$^{(R)}$ 642.
Camera Control: SILICON VIDEO$^{(R)}$ 643.

Device Independent Image Access Interface.
PXIMAGE: Image and Pixel Specifications.
PXIMAGE: Set & Get Area of Interest Window.
PXIMAGE: Get Advice, Errors, Info & Options.
PXIMAGE: Set to Access Sequential Pixels.
PXIMAGE: Read Sequential Pixels.
PXIMAGE: Write Sequential Pixels.
PXIMAGE: Wait for Sequential Access Completion.
PXIMAGE: Obtain Sequential Access Pixel Count.
PXIMAGE: Set to Access Non-Sequential Pixels.
PXIMAGE: Prepare for Non-Sequential Pixel Access.
PXIMAGE: Access Non-Sequential Pixels.
PXIMAGE: Wait for Non-Sequential Access Completion.
PXIMAGE: Set to Obtain Pointer to Pixels.
PXIMAGE: Obtain Pointer to Pixels.
PXIMAGE: Release Pointer to Pixels.

Application Notes.
Conversion of Application Programs from XCOBJ.
Using a Serial (RS-232) Configurable Camera with the PIXCI$^{(R)}$ CL1, CL2, CL3SD, D, D24, D32, D2X, D3X, E1, E1DB, E4, E4DB, EB1, EB1-PoCL, EC1, ECB2, EL1, EL1DB, SI, or SI4.
Initialization of RS-232 or Camera Link Serial Configured Cameras.
Initialization of Camera Link Serial or I^{2}C Configured Cameras.
Adjustments to RS-232 or Camera Link Serial Configured Cameras.
Adjustments to I^{2}C Configured Cameras.
Using a Video Format Configuration Exported by XCAP.
Capture Directly into Memory Space of Other Devices.
PIXCI$^{(R)}$ SV4 SV5, SV5A, or SV5B and DirectDraw.
PIXCI$^{(R)}$ and StreamStor.
Adjusting White Balance of a Bayer Output Camera with the PIXCI$^{(R)}$ Imaging Board.
Processing Images from a Bayer Output Camera with the PIXCI$^{(R)}$ CL, D, E, or SI Series.

Release Notes.

Distribution Contents.

Index.

‡. SCF function included with XCLIB-Lite.

Image Processing & Analysis - The optional **PXIPL Image Processing and Analysis Library**, provides a wide selection of pre-written imaging routines.

The major categories include: processing, enhancements, graphic lines and shapes, text overlay, printing, morphology, filters and edge detectors, transforms, convolutions, sequence integration and averaging, image printing, image copy and resizing, single image and image pair normalizations, blob analysis, histograms and moments, image load and save, calibration, correlation, subpixel accuracy measurements, particle tracking, and image load/save with additional file formats.

PXIPL simplifies development of user-written programs in applications such as image enhancement, archival, analysis, and measurement; event and motion study; document capture; particle analysis; visual inspection; machine vision and quality control.

The PXIPL functions are not restricted to processing images which were captured by EPIX® imaging boards. Images from any source, residing in PC memory, can be any size and any number of colors, limited only by availability of PC memory, and the CPU word size.[8]

Typical PXIPL functions provide a broad spectrum of operations, allowing a single function to do the work of many functions. For example, a convolution function accepts parameters describing the image buffer, the area of interest within the buffer, the convolution size *N*, and the *N*x*N* kernel coefficients. This single function allows convolving with a 3x3, 9x9, 31x31, or 99x99 kernel size, limited only by available PC memory.

XCLIB-Lite Version - XCLIB is available in a reduced, economical, version with limited features — sufficient to support many, common, applications.

XCLIB-Lite may be used with multiple PIXCI® imaging boards, but allows selecting and operating only one board at a time. For a PIXCI® E1DB, E4DB, ECB2, EL1DB, or E4DB imaging boards, or for a PIXCI® SI4 imaging board, XCLIB-Lite allows operating only one functional half, or quarter, respectively, of a selected board.

XCLIB-Lite provides the same Simple C Functions (SCF) for snapping images, capturing sequences, and dual buffer (ping-pong) capture, and for reading and writing single or blocks of pixel values in RGB or monochrome; but does not provide the "structured" interface, bit depth conversions, optional color space conversions to HSB and YUV/YCrCb, nor allow reading or writing only a single color component of RGB. For Bayer pattern cameras, XCLIB-Lite does provide the same de-mosaicked, color corrected, white balanced, and gamma corrected ready-to-use RGB pixel data as XCLIB, but does not provide access to raw Bayer data.

XCLIB-Lite does not allow using more than 64 MBytes of frame buffer memory, and does not provide direct (i.e. "pointer") access to frame buffers. XCLIB-Lite can't be used with the **PXIPL Image Processing and Analysis Library**.

For the PIXCI® A, D, D24, D2X, D3X, CL1, CL2, CL3SD, E1, E1DB, E4, E4DB, EB1, EB1-PoCL, EC1, ECB1, ECB1-34, ECB2, EL1, and EL1DB, imaging boards when used with a camera in other than free-run mode, XCLIB-Lite allows importing a video setup file specifying the exposure, frame/line period, (i.e. the so-called "PRIN(C)" and "EXSYNC"), and other camera control parameters but does not provide for altering these imaging board parameters via function calls. However, for SILICON VIDEO® cameras XCLIB-Lite provides the same camera specific (SCF) functions and features as XCLIB.

XCLIB-Lite allows use of a video setup configuration file, but does not allow the file to be compiled into the application. XCLIB-Lite provides a subset of XCLIB's functions; programs written using XCLIB-Lite are upward compatible with XCLIB.

XCLIB-Lite is available only for Windows 95, 98, ME, NT, 2000, XP, Vista, and 7 (32 bit).

XCLIB™ LIBRARY for PIXCI® Imaging Board

FEATURES

```
HDC  hDC; RECT rect;
  pxd_PIXCIopen("", "RS-170", "");           // Use standard RS-170, or
//pxd_PIXCIopen("", "NTSC", "");             // .. NTSC, or
//pxd_PIXCIopen("", "NTSC/YC", "");          // .. NTSC S-VIDEO, or
//pxd_PIXCIopen("", "CCIR", "");             // .. CCIR, or
//pxd_PIXCIopen("", "PAL", "");              // .. PAL, or
//pxd_PIXCIopen("", "PAL/YC", "");           // .. PAL S-VIDEO, or
//pxd_PIXCIopen("", "", "VIDEO.FMT");        // .. customized formats
                                             //    & resolution?
hDC = GetDC(hWnd);                           // get handle to window
GetClientRect(hWnd, &rect);                  // get size of window
rect.right++; rect.bottom++;                 // inclusive->exclusive
SetStretchBltMode(hDC,STRETCH_DELETESCANS);  // set window display modes
pxd_doSnap(0x1, 1, 0);                       // snap image into buffer 1
pxd_renderStretchDIBits(0x1, 1, 0, 0, -1,-1,// specify frame buffer & AOI
            0, hDC,                          // specify window
            rect.right/4, rect.bottom/4,     // window upper left corner
            rect.right/2, rect.bottom/2,     // window size
            0);                              // display!
ReleaseDC(hWnd, hDC);                        // release handle
```

Image capture and VGA display example (Windows only).

```
pxd_doSnap(0x3, 2, 0);              // snap image into buffer 2 of
                                    // board 0x1 and buffer 2 of
                                    // board 0x2.
pxd_saveBmp(0x1,"IMAGE1.BMP",2,     // save 256x128 AOI of buffer 2 of
            0,0,256,128,0,0)        // board 0x1 in .bmp format (in
                                    // DOS versions too!)
pxd_saveBmp(0x2,"IMAGE2.BMP",2,     // save full buffer 2 of board 0x2
            0,0,-1,-1,0,0)          // in .bmp format
```

Dual board image capture and save example.

```
int b;
  pxd_PIXCIopen("", "NTSC", "");           // Use standard NTSC, or
//pxd_xcopen("", "CCIR", "");              // .. CCIR, or a saved
//pxd_xcopen("", "", "video.fmt");         // .. nonstandard video?
for (b = 1; b <= pxd_imageZdim(); b++) {   // Run thru all buffers.
    printf("Key ENTER to capture buffer #%ld\n", b);
    while (getchar() != '\n') ;            // Wait for user trigger
    pxd_doSnap(0x1, b, 0);                 // Capture into buffer 'b'
}
printf("All buffers captured.\n");
```

Interactive sequence capture example.

```
  pxd_PIXCIopen("", "NTSC/YC", "");        // Use standard S-Video, or
//pxd_PIXCIopen("", "PAL", "");            // .. PAL
pxd_renderDirectVideoInit(1, hWnd);        // init direct video to S/VGA
pxd_renderDirectVideoLive(1, hWnd,         // display live video on S/VGA
            0,0,INT_MAX,INT_MAX,           // location & extent in window
            RGB(0,0,189),RGB(0,0,189));    // color keyed to allow graphic
hDC = GetDC(hWnd)                          // overlays over live video
RECT rect;
SetRect(&rect, 20, 20, 40, 40);            // any standard GDI/MFC calls
DrawFocusRect(hDC, &rect);                 // can be used to draw graphics
ReleaseDC(hWnd, hDC);                      // OVER the live video
```

Live video to S/VGA with graphics overlay example (Windows only).

SPECIFICATIONS

IMAGING BOARD:

Any PIXCI® series imaging board.

Up to eight (32 bit O.S.) or sixteen (64 bit O.S.) identical, or four non-identical, PIXCI® imaging boards can be controlled.[9]

ENVIRONMENT:

- Watcom C/C++ V11.0 32 bit in F model. For Tenberry (Rational) DOS extender, 80386 or better.
- Windows 95, 98, ME 32 bit DLLs, for any compiler or Windows application.
- Windows NT (V4.0), 2000, XP, Vista, 7 32 bit DLLs, for any compiler or Windows application.
- Windows XP(x64), Vista(x64), 7(x64) 64 bit DLLs, for any compiler or Windows application.
- Linux V2.4.8 or later kernel on Intel i386.
- Linux V2.6 or later kernel on Intel x86-64.

XCLIB-Lite only available for Windows 95, 98, ME, NT, 2000, XP, Vista, 7 (32 bit).

Other environments available on request.

LICENSING:

Licensing permits royalty free inclusion of library routines into programs using EPIX® PIXCI® series imaging boards.

SOFTWARE INCLUDES:

As required by environment: Object code library (.lib), Dynamic Link Library (.dll), Object code archive (.a), Windows 95, 98, ME Driver (.vxd), Windows NT, 2000, XP, Vista, 7, XP(x64), Vista(x64), 7(x64) Driver (.sys), and/or Linux driver (.o, .ko).

C prototype files (.h).

Example C/C++ programs demonstrating typically used PIXCI® imaging board features, with Windows (GDI), DOS (stdio), or Linux (stdio & Gnome) interface (.c/.cpp).

Example Visual Basic, VB.NET, C#.NET programs. Labview example available separately.

XCLIB function declaration files for Visual Basic, VB.NET, C#.NET, Matlab.

HTML manual.

Printed manual (not w. XCLIB-Lite).

†. These older products are no longer in production, but are still supported by XCLIB.
[1] Software may require upgrade to operate future versions or options of PIXCI® imaging boards.
[2] XCLIB provides access to pixel data so that the application can perform analysis. The optional PXIPL library provides pre-written analysis routines.
[3, 4] Support for different video formats is dependent upon the particular camera and PIXCI® imaging board used.
[5] Support for varying resolution is dependent upon the particular camera and PIXCI® imaging board used.
[6] Support for general purpose input and output signals is dependent upon the particular PIXCI® imaging board used.
[7] Support for live video via DirectDraw is dependent upon the native color space of the particular PIXCI® imaging board used, and on the capabilities of the S/VGA card used.
[8] The size of a single image may not exceed 2^{32}-1 bytes.
[9] Each functional half of a PIXCI® E1DB, EL1DB, or E4DB, or each functional quarter of a PIXCI® SI4, counts as one board.

Specifications subject to change without notice.

EPIX® imaging products are made in the USA.

EPIX, PIXCI, QUICK SET VIDEO, XCAP, XCIP, XCOBJ, and XCLIB are trademarks or registered trademarks of EPIX, Inc. Other brand, product, and company names are trademarks or registered trademarks of their respective owners.

Copyright © 2009 EPIX, Inc. All rights reserved. 10-Nov-2009.

EPIX®

EPIX, Incorporated
381 Lexington Drive
Buffalo Grove, IL 60089 USA
Tel - 847 465 1818
Fax - 847 465 1919
epix@epixinc.com
www.epixinc.com