



FrameLink Express™

FrameLink to FrameLink Express migration notes

CONFIDENTIAL NOTICE:

Copyright ©2008, Imperx, Inc. All rights reserved.

Any unauthorized use, duplication or distribution of this document or any part thereof, without the prior written consent of Imperx Corporation is strictly prohibited.

Imperx, Inc.
6421 Congress Ave Ste. 204
Boca Raton, FL 33487 USA

DOC-0014-0003
Rev. RA01
06/30/09



Revision History

Revision History			
RA01	Apr-11-2009	C. Margorin	Initial Release

Imperx, Inc.
6421 Congress Ave Ste. 204
Boca Raton, FL 33487 USA

DOC-0014-0003
Rev. RA01
06/30/09

Table Of Contents

1. Frame grabber initialization.....	4
2. Device port specific functions.....	4
3. UART interface initialization.....	4
4. UART interface configuration.....	5
4.1. Baud rate configuration.....	5
4.2. Time-out parameters.....	5
5. UART interface additional functionality.....	5
6. Image acquisition initialization.....	5
7. Image acquisition configuration.....	6
7.1. Bit depth.....	6
7.2. Tap reconstruction.....	7
7.3. Image size.....	7
8. Image acquisition.....	7
8.1. Single image snap mode.....	7
8.2. Stream grabbing mode.....	7
9. Image unpacking and saving.....	8
9.1. Image unpacking.....	8
9.2. Image saving.....	8
10. Camera configuration file.....	9
11. Function changes.....	9
11.1. Initialization functions.....	9
11.1.1. Initialize frame grabber (VCECLB_Init()).....	9
11.1.2. Release frame grabber (VCECLB_Done()).....	9
11.2. Serial communication port (UART interface).....	9
11.2.1. Get access to UART interface (VCECLB_UART_InitEx).....	9
11.2.2. Release access to UART interface (VCECLB_UART_DoneEx).....	10
11.2.3. Set baud rate (VCECLB_UART_SetBREx).....	10
11.2.4. Get baud rate (VCECLB_UART_GetBREx).....	10
11.2.5. Set timeout value (VCECLB_UART_SetTimeoutsEx).....	11
11.2.6. Get timeout value (VCECLB_UART_GetTimeoutsEx).....	11
11.2.7. Write data (VCECLB_UART_WriteEx).....	12
11.2.8. Read data (VCECLB_UART_ReadEx).....	12
11.2.9. Data available in port (VCECLB_UART_GetNumBytesAvailEx).....	13
11.2.10. Flush incoming buffer (VCECLB_UART_FlushEx).....	13
11.2.11. Create virtual serial port (VCECLB_UART_CreateSerialPortEx).....	13
11.2.12. Destroy virtual serial port (VCECLB_UART_DestroySerialPortEx).....	14
11.3. Image acquisition functions.....	14
11.3.1. Get access to image acquisition (VCECLB_GetDMAAccessEx).....	14
11.3.2. Release access to image acquisition (VCECLB_ReleaseDMAAccessEx).....	15
11.3.3. Camera on line status (VCECLB_GetCameraStatusEx).....	15
11.3.4. Learn frame size (VCECLB_LearnFrameSizeEx).....	15
11.3.5. Set buffer queue length (VCECLB_SetBufferCountEx).....	16
11.3.6. Configure frame grabber (VCECLB_PrepareEx).....	16
11.3.7. Start grabbing (VCECLB_StartGrabEx).....	17
11.3.8. Stop grabbing (VCECLB_StopGrabEx).....	17
11.3.9. Grabbing callback (GrabFrame_CallBackEx).....	18

11.3.10.Set snapping timeout (VCECLB_SetDMATimeoutEx).....	18
11.3.11.Snap image (VCECLB_SnapEx).....	18
11.3.12.Reset image acquisition (VCECLB_ResetGrabberEx).....	19
11.4.Camera control and pulse generator.....	19
11.4.1.Configure Camera Control strobes states (VCECLB_SetCCStateEx).....	19
11.4.2.Retrieve Camera Control strobes states (VCECLB_GetCCStateEx).....	20
11.4.3.Configure pulse generator (VCECLB_SetCCPulseGeneratorEx).....	20
11.4.4.Get pulse generator configuration (VCECLB_GetCCPulseGeneratorEx).....	21
11.4.5.Start pulse generator (VCECLB_StartPulseGeneratorEx).....	21
11.5.Pixel unpacking and image saving functions.....	22
11.5.1.Pixel unpacking (VCECLB_UnpackRawPixelsEx).....	22
11.5.2.Bitmap file saving unpacking (VCECLB_SaveRawToBMPFile).....	22
11.5.3.JPEG file saving unpacking (VCECLB_SaveRawToJPGFile).....	23
11.5.4.TIFF file saving unpacking (VCECLB_SaveRawToTIFFFile).....	23

1. Frame grabber initialization

The frame grabber initialization methods has not been changed.

The main sequence is the same:

```
HANDLE hGrabber = VCECLB_Init(); // Open frame grabber
if(hGrabber == NULL) // Frame grabber is not accessible
    ... // Process error condition
... // Do other stuff
VCECLB_Done(hGrabber); // Close frame grabber
```

2. Device port specific functions

Most of device port specific functions, such as UART read/write, image acquisition and pulse generator functions, has been extended with '*port*' parameter.

The '*port*' parameter is zero-based value, specifies port number in Base CameraLink configuration: value '0' stands for 'Base #1' port on card and value '1' stands for 'Base #2' port on card.

Some of operations on ports (such as UART read/write and image acquisition) should be restricted to one application only. For this operation additional initialization and releasing functions has been invented. See corresponding interface initialization chapter below, for more information about this functions.

3. UART interface initialization

To allow simultaneous access from number of application to different interfaces and ports of frame grabber additional initialization functions has been invented. For UART interface typical procedure showed below.

```
HANDLE hGrabber; // Handle to frame grabber
char port; // Input port
char data[256]; // Data in/out buffer
VCECLB_Error err = VCECLB_UART_InitEx(hGrabber, port); // Acquire access to UART
if(err != VCECLB_Err_Success) // Port busy.
    ... // Handle error
VCECLB_UART_SetBREx(hGrabber, port, 9600); // Set UART baud rate
... // Other stuff
strcpy(data, "uart_command\r"); // Prepare output data
DWORD bufSize = strlen(data); // Determine length of output data
VCECLB_UART_WriteEx(hGrabber, port, data, &bufSize, 0); // Send output data to
    camera with default timeout
... // Other stuff
bufSize = sizeof(data); // Determine length of input data buffer
VCECLB_UART_ReadEx(hGrabber, port, data, &bufSize, 0); // Read incoming data
    from camera with default timeout
... // Do other stuff
VCECLB_UART_DoneEx(hGrabber, port); // Release access to UART
```

4. UART interface configuration

4.1. Baud rate configuration

The baud rate parameter has been changed from non-obvious, frame grabber specific constants to

baud rate value. i.e. instead of using the constant value 'VCECLB_UART_BR_9600' which was a 0xD7 (215), just value '9600' should be used.

4.2. Time-out parameters

It becomes possible to set default time-out parameters using the Windows API standard structure COMMTIMEOUTS. This structure specifies inter-character time-out value for reading operation and total timeout values for both read and write operations.

Currently, for backward compatibility there are two ways to set read/write time-out value:

- 1) Using the **VCECLB_UART_SetTimeoutsEx** function.
- 2) Using the time-out value of read/write functions

Non-zero time-out parameter of read/write functions has greater priority and overrides the total time-out value set by **VCECLB_UART_SetTimeoutsEx** function.

For more information about COMMTIMEOUTS structure please refer to MSDN Library (<http://msdn.microsoft.com/en-us/library/aa363190.aspx>)

5. UART interface additional functionality

Some new functions has been added to UART interface, to support CameraLink v1.13 specification:

- 1) **VCECLB_UART_GetNumBytesAvailEx** function retrieves number of bytes ready to be read from driver input buffer.
- 2) **VCECLB_UART_FlushEx** function empties driver input buffer. All data, that has not been read from UART but already been sent by camera will be lost.

6. Image acquisition initialization

To allow simultaneous access from number of application to different interfaces and ports of frame grabber additional initialization functions has been invented. For image acquisition interface typical procedure showed below.

```
HANDLE hGrabber; // Handle to frame grabber
char port; // Input port
VCECLB_CameraDataEx cameraConfig; // Camera specific parameters
VCECLB_Error err = VCECLB_GetDMAAccessEx(hGrabber, port); // Acquire access to
    image acquisition on port
if(err != VCECLB_Err_Success) // Port busy.
    ... // Handle error
VCECLB_PrepareEx(hGrabber, port, &cameraConfig); // Configure image acquisition
    engine
... // Other stuff
VCECLB_StartGrabEx(hGrabber, port, 0, &GrabCallback, pCallbackData); // Start
    grabbing
... // Other stuff
VCECLB_StopGrabEx(hGrabber, port); // Stop grabbing
... // Do other stuff
VCECLB_ReleaseDMAAccessEx(hGrabber, port); // Release access to image
    acquisition
```

7. Image acquisition configuration

One of the biggest changes between FrameLink and FrameLink Express SDK is the camera

parameters structure. The comparison of the structures is given bellow.

FrameLink	FrameLink Express
VCECLB_CameraData	VCECLB_CameraDataEx
WidthPreValid	WidthPreValid
Width	Width
WidthPostValid	WidthPostValid
HeightPreValid	HeightPreValid
Height	Height
HeightPostValid	HeightPostValid
CameraMode	BitDepth, TapConfig.taps
CameraControl	<i>Obsolete, moved to separate function</i>
Strobes	IgnoreDVAL, InvertDVAL, InvertLVAL, InvertFVAL
ReverseTap	TapConfig.tapDirection
TapOrder	TapConfig.tapOutput
ColorMode	<i>Obsolete, moved to VCECLB_RawPixelInfoEx.BayerPattern</i>
BayerMode	<i>Obsolete, moved to VCECLB_RawPixelInfoEx.BayerStart</i>
red_gain, green_gain, blue_gain	<i>Obsolete, moved to VCECLB_RawPixelInfoEx.red_offset, VCECLB_RawPixelInfoEx.green_offset, VCECLB_RawPixelInfoEx.blue_offset</i>

Image processing parameters, such as Bayer Pattern, RGB gain and RGB offset has been moved from camera configuration structure to Raw Pixel information structure.

7.1. Bit depth

The bit depth parameter has been changed from non-obvious constant values, that also includes number of taps, to value of bit depth.

i.e. instead of:

```
CameraData.CameraMode = VCECLB_CM_2_12;
```

you should specify

```
CameraDataEx.BitDepth = 12;
```

```
CameraDataEx.TapConfig.taps = 2;
```

7.2. Tap reconstruction

Tap reconstruction parameters has been extended with additional parameters, such as number of taps (up to 4) and channels (up to 2), order of taps and channel (interleaved, or separated) and directions (up-down or bottom-up for channel and left-right or right-left for taps). Total number of camera output taps should be equal to product of number of channels and number of taps in tap configuration structure.

To simplify selection of parameters, common used modes has been predefined, and can be retrieved using **VCECLB_GetPredefinedTapConfig** function

7.3. Image size

To detect camera size **VCECLB_LearnFrameSizeEx** function has been invented. This function returns height and width of one tap.

Note that this does not account for single vs. dual tap mode.

For width parameter this function simply counts the number of clocks per line during which LVAL is asserted and DVAL is asserted (unless Ignore_DVAL is enabled).

For height parameter this function simply counts the number of rising edges of LVAL during the time that FVAL is asserted.

To get correct (not sheared) picture, sum WidthPreValid+Width+WidthPostValid should be equal to product of DetectedWidth and TapConfig.taps.

8. Image acquisition

In FrameLink Express two image acquisition configuration functions (**VCECLB_PrepareSnap** and **VCECLB_PrepareGrab**) has been merged to one function **VCECLB_PrepareEx**. So it becomes possible to mix both grabbing mode in one application, and use them simultaneously.

8.1. Single image snap mode

Functions **VCECLB_SnapFrame**, **VCECLB_LockBuffer** and **VCECLB_UnlockBuffer** has been removed.

Function **VCECLB_ContinuousSnap** has been extended with port number and renamed to **VCECLB_SnapEx** function.

Since **VCECLB_SnapEx** function uses Direct Memory Access (DMA) engine it is required to pass pointer to memory, aligned to host machine memory page size (usually 4096 byte). The easiest way to get such memory buffer is to use **VirtualAlloc** Windows API function to allocate memory buffer. For more information about this function please refer to MSDN Library ([http://msdn.microsoft.com/en-us/library/aa366887\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa366887(VS.85).aspx))

This mode uses DMA timeout parameter, that can be set using **VCECLB_SetDMATimeoutEx** function.

8.2. Stream grabbing mode

The callback function has been changed, to support additional information about grabbed frame, such as time stamp, frame number, etc.

It becomes possible to call stop grab from the callback function (in FrameLink SDK it causes the dead lock).

Number of buffers used in internal grabbing queue could be changed using **VCECLB_SetBufferCountEx** function. This function should be called before **VCECLB_PrepareEx**. Minimum number of buffers limited to 3.

9. Image unpacking and saving.

Another big changes is concerned pixel unpacking function.

The Handle to DIB data becomes obsolete, but has been kept for backward compatibility.

Parameters for unpack functions has been changed, so now it is not needed to have initialized and configured frame grabber, so it is possible to process raw data after grabbing session, when frame grabber is not accessible anymore.

There is a new structure, that defines rules for raw pixel unpacking. This structure has an original camera parameters structure as one of field.

Fields, that previously was declared as 'gain' parameters for color channels has been renamed to 'offset' fields to reflect their essence. Additionally, floating point 'gain' parameters has been added.

New couple of fields for Lookup Table has been added. **LookupTableLevels** field contains array of levels to compare with, **LookupTableValues** field contains array of value to replace corresponding level. This fields should be set to NULL to ignore LUT processing, or set to pointer to array with size equal maximum number of values.

```
RawPixelInfo.LookupTableLevels = new INT32 [1<<RawPixelInfo.cameraData.BitDepth]
RawPixelInfo.LookupTableValues = new INT32 [1<<RawPixelInfo.cameraData.BitDepth]
```

It is recommended to have **RawPixelInfo.LookupTableLevels** array sorted.

9.1. Image unpacking

Instead of overall output buffer size, **VCECLB_UnpackRawPixelsEx** function uses image stride size (i.e. number of bytes between starts of two adjacent rows).

If **VCECLB_UnpackRawPixelsEx** function has been called with invalid (NULL) pointer to output buffer, or stride size passed to function is too small to hold the line of output buffer, recommended stride size returns via *pStrideSize* parameter and function returns with error **VCECLB_Err_badArgument**.

Additional parameters to control output format has been added. This parameters controls:

- 1) Output bit depth (for non 8bit only)
 - a) Should output data be reduced to 8 bit.
 - b) Should output data be kept as is (e.g. for 12bit occupy lower 12 bit of 16bit value)
 - c) Should output data be normalized (i.e. for 12bit value of 4095(0xFFFF) will became 65520 (0xFFFF0))
- 2) Number of output channels for color images (3 or 4 channels)
- 3) Image direction (Top-Down or Bottom-Up)

9.2. Image saving

Different functions to save images in Bitmap, JPEG and TIFF format has been added. All functions support both ANSI (multi-byte) and Unicode (wide-char) file names.

Saving functions receives raw pixel information structure, that describes the raw data and unpacking rules.

TIFF saving function also receives Output format parameter, to control output file bit depth.

10. Camera configuration file

Function to read and save camera configuration has been added. Read function supports both FrameLink application camera configuration format ('cam' file) and new FrameLink Express Camera XML File

11. Function changes

11.1. Initialization functions

11.1.1. Initialize frame grabber (VCECLB_Init())

No any changes.

11.1.2. Release frame grabber (VCECLB_Done())

No any changes.

11.2. Serial communication port (UART interface)

11.2.1. Get access to UART interface (VCECLB_UART_InitEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_UART_Init(  
    HANDLE hVCECLB  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_InitEx(  
    HANDLE hVCECLB,  
    char port  
);
```

Changes:

Input port number has been added.

Call to initialization function becomes mandatory. If initialization function has not been called before any other UART function, VCECLB_Err_notInitialized error occurs.

Function verifies, that no other process has acquire UART interface for specified port, and returns error condition, if port is busy.

Each successful call to initialization function call should be balanced by a corresponding call to **VCECLB_UART_DoneEx**.

11.2.2. Release access to UART interface (VCECLB_UART_DoneEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_DoneEx(  
    HANDLE hVCECLB,  
    char port  
);
```

Changes:

Each successful call to **VCECLB_UART_InitEx** should be balanced by a corresponding call to **VCECLB_UART_DoneEx**.

11.2.3. Set baud rate (VCECLB_UART_SetBREx)

FrameLink prototype:

```
VCECLB_Error VCECLB_UART_SetBR(  
    HANDLE hVCECLB,  
    BYTE baudRate  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_SetBREx(  
    HANDLE hVCECLB,  
    char port,  
    ULONG baudRate  
);
```

Changes:

Input port number has been added.

If **VCECLB_UART_InitEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

Baud rate parameter has been changed from enumeration constants to baud rate speed value. i.e. instead of **VCECLB_UART_BR_9600 (0xD7)** value 9600 should be specified.

11.2.4. Get baud rate (VCECLB_UART_GetBREx)

FrameLink prototype:

```
VCECLB_Error VCECLB_UART_GetBR(  
    HANDLE hVCECLB,  
    BYTE *pBaudRate  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_GetBREx(  
    HANDLE hVCECLB,  
    char port,  
    ULONG *pBaudRate  
);
```

Changes:

Input port number has been added.

If **VCECLB_UART_InitEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

Baud rate parameter has been changed from enumeration constants to baud rate speed value. i.e. instead of **VCECLB_UART_BR_9600 (0xD7)** value 9600 will be returned.

11.2.5. Set timeout value (VCECLB_UART_SetTimeoutsEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_SetTimeoutsEx(  
    HANDLE hVCECLB,  
    char port,  
    LPCOMMTIMEOUTS lpCommTimeouts  
);
```

Changes:

Input port number has been added.

Sets default timeout values for UART Read/Write operations.

If **VCECLB_UART_InitEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.2.6. Get timeout value (VCECLB_UART_GetTimeoutsEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_GetTimeoutsEx(  
    HANDLE hVCECLB,  
    char port,  
    LPCOMMTIMEOUTS lpCommTimeouts  
);
```

Changes:

Input port number has been added.

Retrieves default timeout values for UART Read/Write operations.

If **VCECLB_UART_InitEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.2.7. Write data (**VCECLB_UART_WriteEx**)

FrameLink prototype:

```
VCECLB_Error VCECLB_UART_Write(HANDLE hVCECLB,  
    BYTE *buffer,  
    DWORD *bufferSize,  
    DWORD serialTimeout  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_WriteEx(  
    HANDLE hVCECLB,  
    char port,  
    BYTE *buffer,  
    DWORD *bufferSize,  
    DWORD serialTimeout  
);
```

Changes:

Input port number has been added.

If **VCECLB_UART_InitEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

Non-zero timeout parameter overrides default constant timeout values.

11.2.8. Read data (**VCECLB_UART_ReadEx**)

FrameLink prototype:

```
VCECLB_Error VCECLB_UART_Read(  
    HANDLE hVCECLB,  
    BYTE *buffer,  
    DWORD *bufferSize,  
    DWORD serialTimeout  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_ReadEx(  
    HANDLE hVCECLB,  
    char port,  
    BYTE *buffer,  
    DWORD *bufferSize,  
    DWORD serialTimeout);
```

Changes:

Input port number has been added.

If **VCECLB_UART_InitEx** function has not been called for this specific port before

VCECLB_Err_notInitialized error occurs.

Non-zero timeout parameter overrides default constant timeout values.

11.2.9. Data available in port (VCECLB_UART_GetNumBytesAvailEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_GetNumBytesAvailEx(  
    HANDLE hVCECLB,  
    char port,  
    UINT32 *numBytes  
);
```

Changes:

Retrieves number of bytes available in input buffer of UART interface.

If **VCECLB_UART_InitEx** function has not been called for this specific port before VCECLB_Err_notInitialized error occurs.

11.2.10. Flush incoming buffer (VCECLB_UART_FlushEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_FlushEx(  
    HANDLE hVCECLB,  
    char port  
);
```

Changes:

Clears input buffer of UART interface.

If **VCECLB_UART_InitEx** function has not been called for this specific port before VCECLB_Err_notInitialized error occurs.

11.2.11. Create virtual serial port (VCECLB_UART_CreateSerialPortEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_UART_CreateSerialPort(  
    HANDLE hVCECLB,  
    DWORD dwPortNumber  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_CreateSerialPortEx(  
    HANDLE hVCECLB,  
    char port,  
    DWORD dwPortNumber  
);
```

Changes:

Input port number has been added.

If **VCECLB_UART_InitEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.2.12. Destroy virtual serial port (VCECLB_UART_CreateSerialPortEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_DeleteSerialPort(  
    HANDLE hVCECLB  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UART_DeleteSerialPortEx(  
    HANDLE hVCECLB,  
    char port  
);
```

Changes:

Input port number has been added.

If **VCECLB_UART_InitEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.3. Image acquisition functions

11.3.1. Get access to image acquisition (VCECLB_GetDMAAccessEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_GetDMAAccessEx(  
    HANDLE hVCECLB,  
    char port  
);
```

Changes:

Call to get access function becomes mandatory. If initialization function has not been called before

any other image acquisition function, VCECLB_Err_notInitialized error occurs.

Function verifies, that no other process has acquire image acquisition interface for specified port, and returns error condition, if port is busy.

Each successful call to initialization function call should be balanced by a corresponding call to **VCECLB_ReleaseDMAAccessEx**.

11.3.2. Release access to image acquisition (VCECLB_ReleaseDMAAccessEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_ReleaseDMAAccessEx (
    HANDLE hVCECLB,
    char port
);
```

Changes:

Each successful call to initialization function call should be balanced by a corresponding call to **VCECLB_ReleaseDMAAccessEx**.

11.3.3. Camera on line status (VCECLB_GetCameraStatusEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_GetCameraStatus (
    HANDLE hVCECLB,
    BYTE *pCameraStatus
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_GetCameraStatusEx (
    HANDLE hVCECLB,
    char port,
    UINT8 *pCameraOnline
);
```

Changes:

Input port number has been added.

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before VCECLB_Err_notInitialized error occurs.

11.3.4. Learn frame size (VCECLB_LearnFrameSizeEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_LearnFrameSizeEx (
    HANDLE hVCECLB,
    char port,
    ULONG *pPixels,
    ULONG *pLines
);
```

Changes:

Allows to calculate tap size.

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.3.5. Set buffer queue length (VCECLB_SetBufferCountEx)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SetBufferCountEx (
    HANDLE hVCECLB,
    char port,
    ULONG bufferCount
);
```

Changes:

Allows to change grabbing queue length.

Should be called before **VCECLB_PrepareEx** function

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.3.6. Configure frame grabber (VCECLB_PrepareEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_Prepare (
    HANDLE hVCECLB,
    VCECLB_CameraData *pCameraData
);
VCECLB_Error VCECLB_PrepareGrab (
    HANDLE hVCECLB,
    VCECLB_CameraData *pCameraData
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_PrepareEx (
    HANDLE hVCECLB,
    char port,
    VCECLB_CameraDataEx *pCameraData
);
```

Changes:

Input port number has been added.

Grab and snap mode prepare has been merged to one function

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.3.7. Start grabbing (VCECLB_StartGrabEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_StartGrab(  
    HANDLE hVCECLB,  
    VCECLB_GrabFrame_Callback lpfnCallback,  
    LPVOID lpUserData  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_StartGrabEx(  
    HANDLE hVCECLB,  
    char port,  
    int queueMode,  
    VCECLB_GrabFrame_CallbackEx lpfnCallback,  
    LPVOID lpUserData  
);
```

Changes:

Input port number has been added.

queueMode is reserved and should be zero

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.3.8. Stop grabbing (VCECLB_StopGrabEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_StopGrab(  
    HANDLE hVCECLB  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_StopGrabEx(  
    HANDLE hVCECLB,  
    char port  
);
```

Changes:

Input port number has been added.

Can be called from callback function.

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before

VCECLB_Err_notInitialized error occurs.

11.3.9. Grabbing callback (GrabFrame_CallBackEx)

FrameLink prototype:

```
VCECLB_Error GrabFrame_CallBack(  
    VCECLB_Error error,  
    LPVOID lpPixelsBuffer,  
    DWORD dwBufferSize,  
    LPVOID lpUserData  
);
```

FrameLink Express prototype:

```
VCECLB_Error GrabFrame_CallBackEx(  
    LPVOID lpUserData,  
    VCECLB_FrameInfoEx *pFrameInfo  
);
```

Changes:

Extra parameters passed via Frame Information structure.

11.3.10. Set snapping timeout (VCECLB_SetDMATimeoutEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_SetDMATimeout(  
    HANDLE hVCECLB,  
    DWORD dwDMATimeout  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SetDMATimeoutEx(  
    HANDLE hVCECLB,  
    char port,  
    DWORD timeout  
);
```

Changes:

Input port number has been added.

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before VCECLB_Err_notInitialized error occurs.

11.3.11. Snap image (VCECLB_SnapEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_ContinuousSnap (
    HANDLE hVCECLB,
    LPVOID lpBuffer,
    DWORD dwBufferSize,
    LPVOID lpTSBuf,
    DWORD dwTSBufSize,
    DWORD *pdwSnappedFrames,
    DWORD *pdwAlignedFrameSize
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SnapEx (
    HANDLE hVCECLB,
    char port,
    void *lpBuffer,
    UINT32 dwBufferSize,
    void *lpTSBuf,
    UINT32 dwTSBufSize,
    UINT32 *pdwSnappedFrames,
    UINT32 *pdwAlignedFrameSize
);
```

Changes:

Input port number has been added.

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.3.12. Reset image acquisition (VCECLB_ResetGrabberEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_ResetGrabber (
    HANDLE hVCECLB
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_ResetGrabberEx (
    HANDLE hVCECLB,
    char port
);
```

Changes:

Input port number has been added.

If **VCECLB_GetDMAAccessEx** function has not been called for this specific port before **VCECLB_Err_notInitialized** error occurs.

11.4. Camera control and pulse generator

11.4.1. Configure Camera Control strobes states (VCECLB_SetCCStateEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_SetCameraCtrlBits (
    HANDLE hVCECLB,
    BYTE pCtrlBits
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SetCCStateEx (
    HANDLE hVCECLB,
    char port,
    VCECLB_CCStrobesEx *pCCStrobes
);
```

Changes:

Input port number has been added.

Strobes states configures via structure, that holds array of strobe states, instead of bit-fields.

New states has been added.

11.4.2. Retrieve Camera Control strobes states (VCECLB_GetCCStateEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_GetCameraCtrlBits (
    HANDLE hVCECLB,
    BYTE *pCtrlBits
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_GetCCStateEx (
    HANDLE hVCECLB,
    char port,
    const VCECLB_CCStrobesEx *pCCStrobes
);
```

Changes:

Input port number has been added.

Strobes states retrieves via structure, that holds array of strobe states, instead of bit-fields.

New states has been added.

11.4.3. Configure pulse generator (VCECLB_SetCCPulseGeneratorEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_SetCCPulseGenerator (
    HANDLE hVCECLB,
    WORD wPulseWidth,
    WORD wPulsePeriod,
    BOOL bPulseLimited,
    BYTE nPulseCount
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SetCCStateEx (
    HANDLE hVCECLB,
    char port,
    const VCECLB_PulseGeneratorEx *pPulseGenerator
);
```

Changes:

Input port number has been added.

Pulse generator configures via structure.

Pulse generator has been extended with slave pulse generator.

11.4.4. Get pulse generator configuration (VCECLB_GetCCPulseGeneratorEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_GetCCPulseGenerator (
    HANDLE hVCECLB,
    WORD *pwPulseWidth,
    WORD *pwPulsePeriod,
    BOOL *pbPulseLimited,
    BYTE *pnPulseCount
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_GetCCPulseGeneratorEx (
    HANDLE hVCECLB,
    char port,
    VCECLB_PulseGeneratorEx *pPulseGenerator
);
```

Changes:

Input port number has been added.

Strobes states retrieves via structure, that holds array of strobe states, instead of bit-fields

11.4.5. Start pulse generator (VCECLB_StartPulseGeneratorEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_StartPulseGenerator(  
    HANDLE hVCECLB,  
    BOOL bStart);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_StartPulseGeneratorEx(  
    HANDLE hVCECLB,  
    char port,  
    UINT8 bStart  
);
```

Changes:

Input port number has been added.

11.5. Pixel unpacking and image saving functions

11.5.1. Pixel unpacking (VCECLB_UnpackRawPixelsEx)

FrameLink prototype:

```
VCECLB_Error VCECLB_UnpackPixels(  
    HANDLE hVCECLB,  
    BYTE *lpPackedBuffer,  
    DWORD dwPackedSize,  
    BYTE *lpUnpackedBuffer,  
    DWORD *pdwUnpackedSize  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_UnpackRawPixelsEx(  
    const VCECLB_RawPixelInfoEx *pPixelInfo,  
    const void *pRawPixelsC,  
    void *pPixels,  
    INT_PTR *pStrideSize,  
    UINT8 OutputFormat,  
    ULONG *pOutputBitDepth  
);
```

Changes:

Handle to frame grabber has been replaced with Raw pixel information structure.

Raw pixel buffer size has been removed.

Unpacked pixel buffer size has been replaced with Unpacked pixel stride size.

Output format flags has been added.

11.5.2. Bitmap file saving unpacking (VCECLB_SaveRawToBMPFile)

FrameLink prototype:

```
VCECLB_Error VCECLB_SaveDIBToFile(  
    HGLOBAL hDIB,  
    char *lpszFileName  
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SaveRawToBMPFileHandle(  
    const VCECLB_RawPixelInfoEx *pPixelInfo,  
    const void *pRawPixelsC,  
    HANDLE hFile  
);  
VCECLB_Error VCECLB_SaveRawToBMPFileA(  
    const VCECLB_RawPixelInfoEx *pPixelInfo,  
    const void *pRawPixelsC,  
    const char *lpszFileName  
);  
VCECLB_Error VCECLB_SaveRawToBMPFileW(  
    const VCECLB_RawPixelInfoEx *pPixelInfo,  
    const void *pRawPixelsC,  
    const wchar_t *lpszFileName  
);
```

Changes:

HDIB has been replaced with Raw pixel information structure and Raw pixel buffer.
Wide-char (Unicode) version has been added in addition to File Handle version.

11.5.3. JPEG file saving unpacking (VCECLB_SaveRawToJPGFile)

FrameLink prototype:

Not exist

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SaveRawToJPGFileA(  
    const VCECLB_RawPixelInfoEx *pPixelInfo,  
    const void *pRawPixelsC,  
    const char *lpszFileName,  
    UINT8 nQuality  
);  
VCECLB_Error VCECLB_SaveRawToJPGFileW(  
    const VCECLB_RawPixelInfoEx *pPixelInfo,  
    const void *pRawPixelsC,  
    const wchar_t *lpszFileName,  
    UINT8 nQuality  
);
```

Changes:

Wide-char and Multi-byte version of function has been implemented.

11.5.4. TIFF file saving unpacking (VCECLB_SaveRawToTIFFFile)

FrameLink prototype:

```
VCECLB_Error VCECLB_GetTiffData (
    HANDLE hVCECLB,
    HANDLE *phTIFFData
);
VCECLB_Error VCECLB_ConvertPixels2Tiff (
    HANDLE hVCECLB,
    BYTE *pBuffer,
    DWORD dwSize,
    HANDLE *phTIFFData
);
VCECLB_Error VCECLB_GetTiffData_LSB (
    HANDLE hVCECLB,
    HANDLE *phTIFFData
);
VCECLB_Error VCECLB_ConvertPixels2Tiff_LSB (
    HANDLE hVCECLB,
    BYTE *pBuffer,
    DWORD dwSize,
    HANDLE *phTIFFData
);
VCECLB_Error VCECLB_ReleaseTiff (
    HANDLE hVCECLB,
    HANDLE *phTIFFData
);
VCECLB_Error VCECLB_TIFFAddComent (
    HANDLE hTIFFData,
    char *comment,
    SYSTEMTIME *st
);
VCECLB_Error VCECLB_SaveTIFFToFile (
    HANDLE hTIFFData,
    char *filename
);
```

FrameLink Express prototype:

```
VCECLB_Error VCECLB_SaveRawToTIFFFileA (
    const VCECLB_RawPixelInfoEx *pPixelInfo,
    const void *pRawPixelsC,
    const char *lpszFileName,
    UINT8 OutputFormat
);
VCECLB_Error VCECLB_SaveRawToJPGFileW (
    const VCECLB_RawPixelInfoEx *pPixelInfo,
    const void *pRawPixelsC,
    const wchar_t *lpszFileName,
    UINT8 OutputFormat
);
```

Changes:

Set of functions has been replaced with one function, that handles all saving modes.

Comment adding feature has been removed.