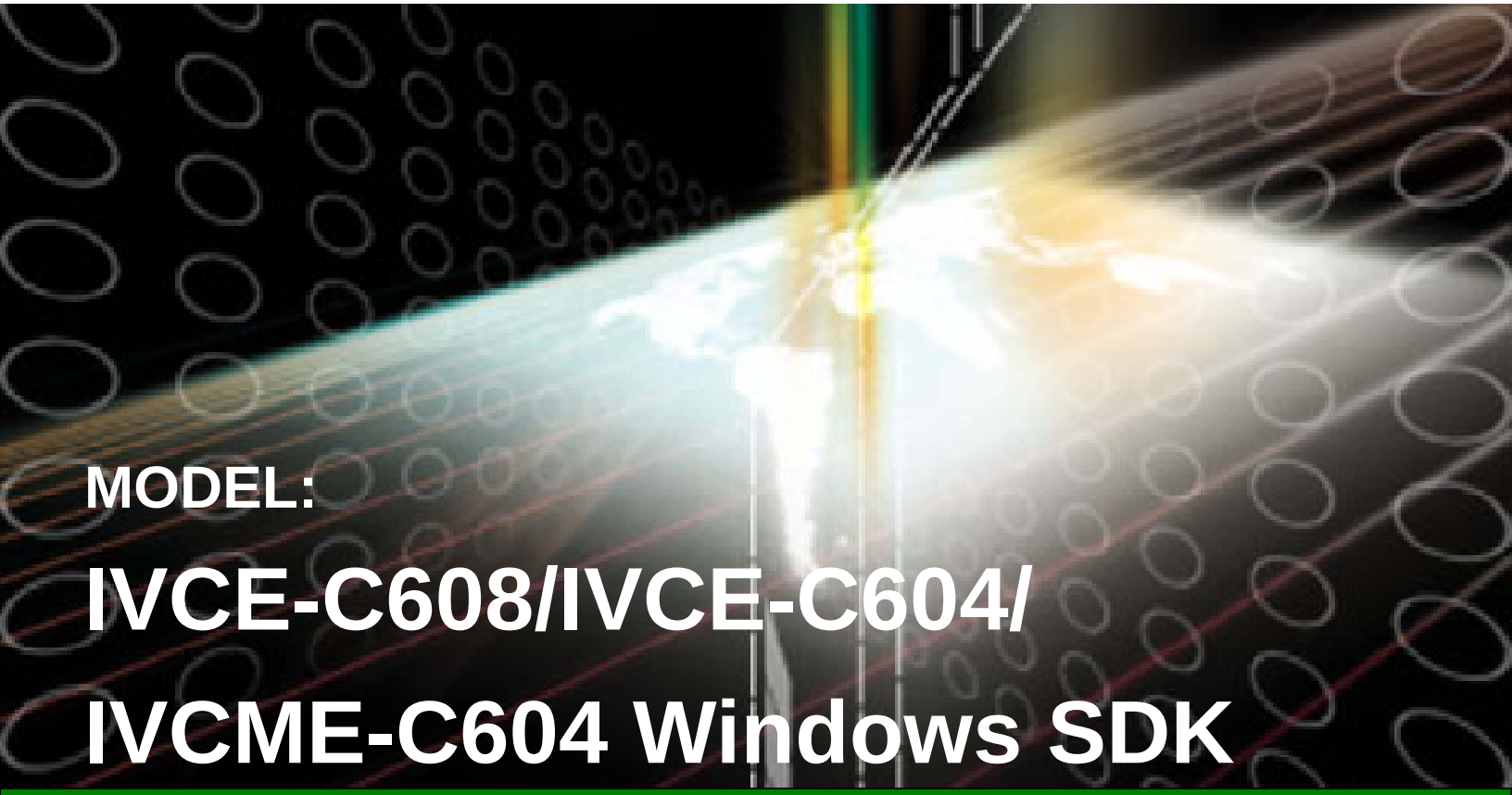
IEI Technology Corp.



MODEL:

IVCE-C608/IVCE-C604/ IVCME-C604 Windows SDK

Detailed Descriptions to SDK Function

User Manual

Rev. 1.00 – 17 January, 2012

RoHS Lead Free

Revision

Date	Version	Changes
17 January, 2012	1.00	Initial release

Table of Contents

1.1 Integrating SDK for Application Development

Step 1: Include the “IVCXXC6XXXdefines.h” header file into the application project after creating visual studio project.

Step 2: Set the included library file to “IVCXXC6XXXSDK.lib”.

Step 3: Copy the “IVCXXC6XXXSDK.dll” to **C:\Windows\System32** or put the .dll file under the same folder of the execution file.

Step 4: The user can start using SDK function calls in the application.

NOTE:

Please refer IEI Video Capture Test Suite application which is developed using this SDK.

1.2 IEI Capture Card Models

- IVCE-C608
- IVCE-C604
- IVCME-C604

1.3 System Requirements

- Microsoft Windows XP
- Microsoft Windows 7
- Microsoft DirectX 9.0c

1.4 Detailed Descriptions of SDK Function

1.4.1 InitializeSDK()

Syntax: IVCC_API BOOL InitializeSDK();

Parameters: NULL

Return value:

Return Code	Description
TRUE	The library was initialized successfully.
FALSE	The library is not initialized.

Remarks: Call this function to initialize capture card.

If SDK initilzed sucessfully returns TRUE otherwise FALSE.

Note : You need to initialize the SDK library before you call any of the library functions

1.4.2 GetNumberofCards()

Syntax: IVCC_API HRESULT GetNumberofCards(BYTE* NoofCards);

Parameters: NoofCards – pointer to BYTE. Returns no of cards installed.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG,

E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get number of card was success.
S_FALSE	Get number of card was failed

Remarks: Call this function to get number of cards installed in PC.

1.4.3 GetBoardID()

Syntax: IVCC_API HRESULT GetBoardID(BYTE cardID,BYTE* boradid);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

NoofCards – pointer to BYTE. Returns board ID.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get board ID of card was success.
S_FALSE	Get board ID of card was failed.

Remarks: Call this function to get hardware board ID of the capture card.

1.4.4 GetFirmwareVersion()

Syntax: IVCC_API HRESULT GetFirmwareVersion(BYTE cardID,BYTE *fVersion);

Parameters: CardID- Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

fVersion - pointer to BYTE.returns firmware version.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get firmware version was success.
S_FALSE	Get firmware version was failed.

Remarks: Call this function to get firmware version of capture card.

1.4.5 GetCardIDs()

Syntax: IVCC_API HRESULT GetCardIDs(BYTE *cardIDs,BYTE cardType);

Parameters: cardIDs – pointer to BYTE. Returns installed card IDs.

cardType – Specify PCIe or PCIe Mini

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get card IDs version was success.
S_FALSE	Get card IDs version was failed.

Remarks: Call this function to get card IDs installed in PC.

Specify card type as any one of the following:

PCIE_4	0 – IVCE-C604
PCIE_8	1 – IVCE-C608
MINIPCIE_4	2 – IVCME-C604

1.4.6 GetNumberofDecoder()

Syntax: IVCC_API HRESULT GetNumberofDecoder(BYTE cardID,BYTE* noofDecoder);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

noofDecoder – pointer to BYTE. Returns no of decoder available.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get number of decoder was success.
S_FALSE	Get number of decoder was failed.

Remarks: Call this function to get number of camera decoders availabile in capture card.

1.4.7 GetCameraInfo()

Syntax: IVCC_API HRESULT GetCameraInfo(BYTE cardID,PCAMERAINFO pstructCameraInfo);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pstructCameraInfo – pointer to structure. Verify structure section to know about PCAMERAINFO.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get camera information was success.
S_FALSE	Get camera information was failed.

Remarks: Call this function to get camera information like numbers of cameras selected, which camera is selected etc..

1.4.8 SetCameraInfo ()

Syntax: IVCC_API HRESULT SetCameraInfo(BYTE cardID,PCAMERAINFO pstructCameraInfo);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pstructCameraInfo – pointer to structure. Verify structure section to know about PCAMERAINFO.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set camera information was success.
S_FALSE	Set camera information was failed.

Remarks: Call this function to set camera information like numbers of cameras selected, which camera is selected etc..

1.4.9 GetDecoderDurationInfo ()

Syntax: IVCC_API HRESULT GetDecoderDurationInfo(BYTE cardID,PDEC_DURATION_INFO pDurationInfo);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pDurationInfo – pointer to structure. Verify structure section to know about PDEC_DURATION_INFO.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get decoder information was success.
S_FALSE	Get decoder information was failed.

Remarks: Call this function to get decoder information and number of fields displayed etc..

1.4.10 SetDecoderDurationInfo ()

Syntax: IVCC_API HRESULT SetDecoderDurationInfo(BYTE cardID,PDEC_DURATION_INFO pDurationInfo);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pDurationInfo – pointer to structure. Verify structure section to know about PDEC_DURATION_INFO.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set decoder information was success.
S_FALSE	Set decoder information was failed.

Remarks: Call this function to set decoder information and number of fields displayed etc..

1.4.11 GetFrameRates ()

Syntax: IVCC_API HRESULT GetFrameRates(BYTE cardID,ULONG *fps);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

fps – pointer to ULONG. Returns frame rates of selected cameras.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get frame rate was success.
S_FALSE	Get frame rate was failed.

Remarks: Call this function to get frame rates of selected cameras.

1.4.12 IsNTSCStandard ()

Syntax: IVCC_API HRESULT IsNTSCStandard(BYTE cardID,BOOL *bNTSC);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

bNTSC – pointer to Boolean. Returns the video standard.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Checking video standard was success.
S_FALSE	Checking video standard was failed.

Remarks: Call this function to get video standard. NTSC standard returns TRUE otherwise FALSE.

FALSE – PAL TRUE – NTSC

1.4.13 IsRotationModeSelected ()

Syntax: IVCC_API HRESULT IsRotationModeSelected(BYTE cardID,BOOL *pRotation);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pRotation – pointer to Boolean. Returns rotation mode selected or not.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Checking rotation mode was success.
S_FALSE	Checking rotation mode was failed.

Remarks: Call this function to get rotation mode selected or not. Camera rotation mode selected returns TRUE otherwise FALSE.

1.4.14 EnableRotationMode ()

Syntax: IVCC_API HRESULT EnableRotationMode(BYTE cardID,BOOL bRotation);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

bRotation - Boolean. Set to Enable or disable rotation mode.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Enable/Disable rotation mode was success.
S_FALSE	Enable/Disable rotation mode was failed.

Remarks: Call this function to enable or disable the rotation mode.

TRUE – Enable

FALSE - Disable

1.4.15 GetColorParam ()

Syntax: IVCC_API HRESULT GetColorParam(BYTE cardID,PCOLORPARAMS pstructColorParam);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pstructColorParam – pointer to structure. Verify structure section to know about PCOLORPARAMS.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get decoder color information was success.
S_FALSE	Get decoder color information was failed.

Remarks: Call this function to get decoder color control information like brightness, contrast etc..

1.4.16 SetColorParam ()

Syntax: IVCC_API HRESULT SetColorParam(BYTE cardID,PCOLORPARAMS pstructColorParam);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pstructColorParam – pointer to structure. Verify structure section to know about PCOLORPARAMS.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set decoder color information was success.
S_FALSE	Set decoder color information was failed.

Remarks: Call this function to set decoder color control information like brightness, contrast etc..

1.4.17 GetTimParam ()

Syntax: IVCC_API HRESULT GetTimParam(BYTE cardID,PMANTIM pstructTimParam);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pstructTimParam – pointer to structure. Verify structure section to know about PMANTIM.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get decoder timing information was success.
S_FALSE	Get decoder timing information was failed.

Remarks: Call this function to get decoder timing information, like horizontal and vertical timing.

1.4.18 SetTimParam ()

Syntax: IVCC_API HRESULT GetTimParam(BYTE cardID,PMANTIM pstructTimParam);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pstructTimParam – pointer to structure. Verify structure section to know about PMANTIM.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set decoder timing information was success.
S_FALSE	Set decoder timing information was failed.

Remarks: Call this function to set decoder timing information, like horizontal and vertical timing.

1.4.19 GetAudioDownStreamChannel ()

Syntax: IVCC_API HRESULT GetAudioDownStreamChannel(BYTE cardID,BYTE* channel);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

channel – pointer to BYTE. Returns selected audio channel.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get audio downstream channel was success.
S_FALSE	Get audio downstream channel was failed.

Remarks: Call this function to get audio downstream channel in the capture card. This API only works on the PCIe series

IVCE-C604 has 4 mono audio channels.

Channels	value
Ain 1 and Ain2	0
Ain 3 and Ain4	1

IVCE-C608 has 8 mono audio channels.

Channels	value
Ain 1 and Ain2	0
Ain 3 and Ain4	1
Ain 5 and Ain6	2
Ain 7 and Ain8	3

1.4.20 SetAudioDownStreamChannel ()

Syntax: IVCC_API HRESULT SetAudioDownStreamChannel(BYTE cardID,BYTE channel);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

channel – BYTE. Set value to select audio channel .

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set audio downstream channel was success.
S_FALSE	Set audio downstream channel was failed.

Remarks: Call this function to set audio downstream channel in the capture card. This API only works on the PCIe series

IVCE-C604 has 4 mono audio channels.

Channels	value
Ain 1 and Ain2	0
Ain 3 and Ain4	1

IVCE-C608 has 8 mono audio channels.

Channels	value
Ain 1 and Ain2	0
Ain 3 and Ain4	1
Ain 5 and Ain6	2
Ain 7 and Ain8	3

1.4.21 GetGPIOValues ()

Syntax: IVCC_API HRESULT GetGPIOValues(BYTE cardID,BYTE cardType,BYTE moduleno,BYTE pinnumber,BYTE* Value);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

cardType - BYTE. Set the card type (PCIE or MINIPCIE).

moduleno - BYTE. Set the module number.

pinnumber - BYTE. Set the module pin number to read.

Value - pointer to BYTE. Buffer to read values.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get GPIO values was success.
S_FALSE	Get GPIO values was failed.

Remarks: Call this function to read GPIO values from the capture card.

cardTypes – PCIE_4, PCIE_8, MINIPCIE

moduleno – IVCE-C604 & IVCME-C604 Only CN1

IVCE-C608 CN1 and CN2

Pinnumber – 0 to 3

1.4.22 SetGPIOValues ()

Syntax: IVCC_API HRESULT SetGPIOValues(BYTE cardID,BYTE cardType,BYTE moduleno,BYTE pinnumber,BYTE Value);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

cardType - BYTE. Set the card type (PCIE or MINIPCIE).

moduleno - BYTE. Set the module number.

pinnumber - BYTE. Set the module pin number to read.

Value - BYTE. Value for write.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set GPIO values was success.
S_FALSE	Set GPIO values was failed.

Remarks: Call this function to write GPIO values from the capture card.

cardTypes – PCIE_4, PCIE_8, MINIPCIE

moduleno – IVCE-C604 & IVCME-C604 Only CN1

IVCE-C608 CN1 and CN2

Pinnumber – 4 to 7

1.4.23 EnableWDT ()

Syntax: IVCC_API HRESULT EnableWDT(BYTE cardID);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Watchdog timer enable was success.
S_FALSE	Watchdog timer enable was failed.

Remarks: Call this function to enable watchdog timer. This API works only on PCIe series.

1.4.24 DisableWDT ()

Syntax: IVCC_API HRESULT DisableWDT(BYTE cardID);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Watchdog timer disable was success.
S_FALSE	Watchdog timer disable was failed.

Remarks: Call this function to disable watchdog timer. This API works only on PCIe series.

1.4.25 ToggleWDT ()

Syntax: IVCC_API HRESULT ToggleWDT(BYTE cardID,BYTE value);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

Value – BYTE. Set value 0 or 1.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values.

Return Code	Description
S_OK	Watchdog timer toggle was success.
S_FALSE	Watchdog timer toggle was failed.

Remarks: Call this function to toggle watchdog timer. This API works only on PCIe series.

1.4.26 AESCheck ()

Syntax: IVCC_API HRESULT AESCheck(BYTE cardID,BOOL *bPass);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

bPass – pointer to Boolean. Returns AES pass or fail.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values.

Return Code	Description
S_OK	AES checking was success.
S_FALSE	AES checking was failed.

Remarks: Call this function to check AES functionality. If AES check is pass it returns TRUE. Otherwise, FALSE. This feature used for security purpose. You can protect application or SDK using this feature.

1.4.27 StartUpStream ()

Syntax: IVCC_API HRESULT StartUpstearm(BYTE cardID,WORD cardType,CString strFilePath,int vidChannel,BOOL AudioChannel,BOOL isNTSC,int pixelFormat);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

cardType – BYTE. Set the card type(PCIE or MINIPCIE) .

strFilePath – CString. Set the file path for playback.

vidChannel – int. Set the upstream video channel number. 0 and 1.

AudioChannel – BOOL. Enable Audio channel upstream.

isNTSC – BOOL. Specify the video standard.

pixelFormat – int. Specify the pixel format to play.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Playback was success.
S_FALSE	Playback was failed.

Remarks: Call this function to playback the video file (supports .asf format). This API works only on the PCIe series.

cardTypes – PCIE_4, PCIE_8

video channels – 0 and 1

Audio channel – 2 if audio channel.specify value 2 in vidchannel parameter.

isNTSC – TRUE: NTSC ; FALSE: PAL

pixelFormat – 0: Y422 ; 1: Y411

1.4.28 StopUpStream ()

Syntax: IVCC_API HRESULT StopUpstearm(BYTE cardID,WORD cardType,CString strFilePath,int vidChannel,BOOL AudioChannel,BOOL isNTSC,int pixelFormat);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

cardType – BYTE. Set the card type(PCIE or MINIPCIE) .

strFilePath – CString. Set the file path for playback.

vidChannel – int. Set the upstream video channel number. 0 and 1.

AudioChannel – BOOL. Enable Audio channel upstream.

isNTSC – BOOL. Specify the video standard.

pixelFormat – int. Specify the pixel format to play.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Stop playback was success.
S_FALSE	Stop playback was failed.

Remarks: Call this function to stop playback the video file (supports .asf format). This API works only on the PCIe series.

cardTypes – PCIE_4, PCIE_8

video channels – 0 and 1

Audio channel – 2 if audio channel.specify value 2 in vidchannel parameter.

isNTSC – TRUE: NTSC ; FALSE: PAL

pixelFormat – 0: Y422 ; 1: Y411

1.4.29 StartByPass ()

Syntax: IVCC_API HRESULT StartByPass(BYTE cardID,WORD channel1,WORD channel2,BOOL bNTSC);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

channel1 – WORD. Select channel number to bybass through Videoout 1 .

channel2 – WORD. Select channel number to bybass through Videoout 2 .

bNTSC – BOOL. Set video standard. TRUE – NTSC. FALSE - PAL

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	By pass was success.
S_FALSE	By pass was failed.

Remarks: Call this function to bypass the video channel input to direct TV input or monitor. This API works only on the PCIe series.

1.4.30 StopByPass ()

Syntax: IVCC_API HRESULT StopByPass(BYTE cardID);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Stop bypass was success.
S_FALSE	Stop bypass was failed.

Remarks: Call this function to stop bypass the video channel input to direct TV input or monitor. This API works only on the PCIe series.

1.4.31 SystemReadyLED()

Syntax: IVCC_API HRESULT SystemReadyLED(BYTE cardID,BOOL bEnable);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

bEnable – pointer to Boolean. Set value to enable/disable.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Enable/disable SystemReadyLED was success.
S_FALSE	Enable/disable SystemReadyLED was failed.

Remarks: Call this function to enable or disable SystemReadyLED status. This API works only on the PCIe series. FALSE – Enable; TRUE – Disable

1.4.32 SystemErrorLED()

Syntax: IVCC_API HRESULT SystemErrorLED(BYTE cardID,BOOL bEnable);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

bEnable – pointer to Boolean. Set value to enable/disable.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values.

Return Code	Description
S_OK	Enable/disable SystemErrorLED was success.
S_FALSE	Enable/disable SystemErrorLED was failed.

Remarks: Call this function to enable or disable SystemErrorLED status. This API works only on the PCIe series. FALSE – Enable; TRUE – Disable

1.4.33 GetVGAGain ()

Syntax: IVCC_API HRESULT GetVGAGain(BYTE cardID,PVGA_STRUCT pVGA,DWORD* gainValue);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pVGA – pointer to structure. Verify structure section to know about PVGA_STRUCT.

gainValue – pointer to DWORD. Returns VGA gain value.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get VGA Gain was success.
S_FALSE	Get VGA Gain was failed.

Remarks: Call this function to Get VGA gain values like gain a, gain b etc..

1.4.34 GetAGCGain ()

Syntax: IVCC_API HRESULT GetAGCGain(BYTE cardID,PAGC_STRUCT pAGC,DWORD* gainValue);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pVGA – pointer to structure. Verify structure section to know about PAGC_STRUCT.

gainValue – pointer to DWORD. Returns VGA gain value.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get AGC Gain was success.
S_FALSE	Get AGC Gain was failed.

Remarks: Call this function to Get AGC gain values like gain a, gain b etc..

1.4.35 SetVGAGain ()

Syntax: IVCC_API HRESULT SetVGAGain(BYTE cardID,PVGA_STRUCT pVGA);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pVGA – pointer to structure. Verify structure section to know about PVGA_STRUCT.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set VGA Gain was success.
S_FALSE	Set VGA Gain was failed.

Remarks: Call this function to Set VGA gain values like gain a, gain b etc..

1.4.36 SetAGCGain ()

Syntax: IVCC_API HRESULT SetAGCGain(BYTE cardID,PAGC_STRUCT pAGC);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pVGA – pointer to structure. Verify structure section to know about PAGC_STRUCT.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Set AGC Gain was success.
S_FALSE	Set AGC Gain was failed.

Remarks: Call this function to Set AGC gain values like gain a, gain b etc..

1.4.37 Calibrate ()

Syntax: IVCC_API HRESULT Calibrate(BYTE cardID,WORD camera,WORD gain_type);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

camera - WORD. Set camera number.

gain_type – WORD. 1 – VGA; 2 - AGC gain.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Calibrating Gain was success.
S_FALSE	Calibrating Gain was failed.

Remarks: Call this function to Calibrate VGA and AGC gain values etc..

1.4.38 GetDefaultGain ()

Syntax: IVCC_API HRESULT GetDefaultGain(BYTE cardID,PVGAAGCDEFAULT pVGAAGCDefault);

Parameters: CardID - Each capture card has a unique card ID. Verify GetCardIDs() Fuction to get card IDs.

pVGAAGCDefault – pointer to structure. Verify structure section to know about PVGAAGCDEFAULT.

Return value: HRESULT

This function can return the standard return values E_INVALIDARG, E_OUTOFMEMORY, and E_UNEXPECTED, as well as the following values

Return Code	Description
S_OK	Get Default Gain was success.
S_FALSE	Get Default Gain was failed.

Remarks: Call this function to Get default gain values.

1 – VGA

2 – AGC gain

1.5 Structures

1.5.1 struct structCameraInfo

This structure is used to get and set the camera selection properties.

```
typedef struct structCameraInfo
{
    BOOL bArrSelectedCameras[MAX_CAMERAS];      //Selected camera : true
    BOOL bArrSelectedLLC[MAX_CAMERAS];          // Selected LLC   : true
    DWORD wNoOfCamreasSelected;                 // No of cameras selected
    BOOL bEnableAllCameras;                     // Enable all camera
}CAMERAINFO,*PCAMERAINFO;
```

BOOL bArrSelectedCameras [MAX_CAMERAS]:

This attribute specifies which camera is selected among 16 cameras.

TRUE: Selected; FALSE: Un-selected

Following are the indexes used for selected cameras.

IVCE-C608 – Indexs 0,2,4,6,8,10,12,14.

IVCE-C604 – Indexs 0,2,4,6.

IVCME-C604 – Indexs 0,1,2,3.

BOOL bArrSelectedLLC [MAX_CAMERAS];

This attribute specifies which camera LLC is selected among 16 cameras.

TRUE: Selected; FALSE: Un-selected

DWORD wNoOfCamreasSelected:

This attribute is used to get total number of selected cameras.

BOOL bEnableAllCameras:

This attribute is used to select all cameras.

1.5.2 struct structColorParam

This structure is used to get and set the camera color properties.

```
typedef struct structColorParam
{
        int brightness;
        int contrast;
        int hue_adj;
        int saturation;
        int sharpness;
        WORD prop_id;
        WORD decoder;
}COLORPARAMS,*PCOLORPARAMS;
```

Int Brightness: this attribute is used to change the brightness of the camera. Maximum value is 10000.Default is 5000.

Int Contrast: this attribute is used to change the contrast of the camera. Maximum value is 10000. Default is 5000.

Int hue_adj: this attribute is used to change the Hue of the camera. Maximum value is 10000.

Int Saturation: this attribute is used to change the saturation of the camera. Maximum value is 10000. Default is 5000.

Int Sharpness: this attribute is used to change the sharpness of the camera. Maximum value is 10000. Default is 5000.

WORD Prop_id: this attribute is to define which property needs to be set.

WORD decoder: this attribute is used to define which camera decoder properties need to be set. Values 0 to 7.

1.5.3 struct structDecoderDurationInfo

This structure is used to get and set the camera decoder duration properties.

```
typedef struct structDecoderDurationInfo
{
  WORD      video_dec;
  WORD      duration_counts; Number of fields displayed per VDEC
  BOOLEAN   enabled_all_durations;// 1: VDEC_A is used as duration for all active decoders
}DEC_DURATION_INFO, *PDEC_DURATION_INFO;
```

WORD video_dec: this attribute specifies to which camera decoder duration properties need to set.

WORD duration_counts: this attribute is used to set number of fields displayed per VDEC.

WORD enable_all_durations: this attribute specifies VDEC_A is used as duration for all active decoders.

1.5.4 struct structManTim

This structure is used to get and set the camera decoder timing properties.

```
typedef struct structManTim
{
      DWORD horiz;
      DWORD vert;
      WORD camera;
}MANTIM, *PMANTIM;
```

DWORD horiz: this attribute is used to set horizontal timing property of the decoder. Maximum value is 140. Default is 116.

DWORD vert: this attribute is used to set vertical timing property of the decoder. Maximum value is 70. Default is 26.

DWORD vert: this attribute is used to select a camera to set timing properties.

1.5.5 struct structVGAAGCDefault

This structure is used to get the VGA and AGC default gain properties.

```
typedef struct structVGAAGCDefault
{
        DWORD gain_value;
        WORD camera;
        WORD gain_type;        // 1: VGA, 2: AGC
}VGAAGCDEFAULT, *PVGAAGCDEFAULT;
```

DWORD gain_value: this attribute specifies the gain value.

WORD camera: this attribute is used to specify camera number in which gain value need to be adjusted.

WORD gain_type: this attribute specifies VGA or AGC. 1: VGA; 2: AGC

1.5.6 struct structVGA

```
typedef struct structVGA
{
        BYTE gain_A;           // VGA value for chx_src0
        BYTE gain_B;           // VGA value for chx_src1
        BYTE auto_enable;      // auto mode enable
        BYTE camera;
}VGA_STRUCT, *PVGA_STRUCT;
```

BYTE gain_A: this attribute specifies the VGA gain value of source 0.

BYTE gain_B: this attribute specifies the VGA gain value of source 1.

WORD auto_enable: this attribute is used to calibrate the gain value automatically.

WORD camera: this attribute is used to specify camera number in which gain value need to be adjusted.

1.5.7 struct structAGC

```
typedef struct structAGC
{
        WORD gain_A;
        WORD gain_B;
        WORD auto_enable;
        WORD camera;
}AGC_STRUCT, *PAGC_STRUCT;
```

BYTE gain_A: this attribute specifies the AGC gain value of source 0.

BYTE gain_B: this attribute specifies the AGC gain value of source 1.

WORD auto_enable: this attribute is used to calibrate the gain value automatically.

WORD camera: this attribute is used to specify camera number in which gain value need to be adjusted.