POINT GREY

FlyCapture 2.1

C Language API Programming Reference

Revised September 16, 2010

Point Grey Research Inc.

12051 Riverside Way • Richmond, BC • Canada • V6W 1K7 • **T** (604) 242-9937 • **www.ptgrey.com**

Software Warranty

Point Grey Research warrants to the Original Purchaser, for a period of one (1) year from date of purchase that:

1. The diskette on which the Software is furnished and the accompanying documentation are not defective;
2. The Software is properly recorded upon the diskettes enclosed;
3. The documentation is substantially complete and contains all the information Point Grey Research deems necessary to use the Software;
4. The Software functions substantially as described in the documentation.

Point Grey Research, Inc.'s entire liability and the Original Purchaser's exclusive remedy shall be the replacement of any diskette or documentation not meeting these warranties. On such an occasion, a copy of the paid receipt accompanied with the faulty diskette or documentation must be returned to Point Grey Research, Inc. or an authorized dealer.

Point Grey Research, Inc. expressly disclaims and excludes all other warranties, express, implied and statutory, including, but without limitation, warranty of merchantability and fitness for a particular application or purpose. In no event shall Point Grey Research, Inc. be liable to the Original Purchaser or any third party for direct, indirect, incidental, consequential, special or accidental damages, including without limitation damages for business interruption, loss of profits, revenue, data or bodily injury or death.

Contents

1 Data Structure Index **1**

1.1 Data Structures 1

2 File Index **3**

2.1 File List 3

3 Data Structure Documentation **5**

3.1 fc2AVIOption Struct Reference 5

3.1.1 Field Documentation 5

3.1.1.1 frameRate 5

3.1.1.2 reserved 5

3.2 fc2CameraInfo Struct Reference 6

3.2.1 Field Documentation 7

3.2.1.1 bayerTileFormat 7

3.2.1.2 configROM 7

3.2.1.3 defaultGateway 7

3.2.1.4 driverName 7

3.2.1.5 firmwareBuildTime 7

3.2.1.6 firmwareVersion 7

3.2.1.7 gigEMajorVersion 7

3.2.1.8 gigEMinorVersion 7

3.2.1.9 iidcVer 7

3.2.1.10 interfaceType 7

3.2.1.11 ipAddress 7

3.2.1.12 isColorCamera 7

3.2.1.13 macAddress 7

3.2.1.14 maximumBusSpeed 7

3.2.1.15 modelName 7

3.2.1.16 reserved 7

3.2.1.17 sensorInfo 7
3.2.1.18 sensorResolution 7
3.2.1.19 serialNumber 7
3.2.1.20 subnetMask 7
3.2.1.21 userDefinedName 7
3.2.1.22 vendorName 7
3.2.1.23 xmlURL1 7
3.2.1.24 xmlURL2 7
3.3 fc2Config Struct Reference 8
3.3.1 Field Documentation 8
3.3.1.1 asyncBusSpeed 8
3.3.1.2 bandwidthAllocation 8
3.3.1.3 grabMode 8
3.3.1.4 grabTimeout 8
3.3.1.5 isochBusSpeed 8
3.3.1.6 numBuffers 8
3.3.1.7 numImageNotifications 8
3.3.1.8 reserved 8
3.4 fc2ConfigROM Struct Reference 9
3.4.1 Field Documentation 9
3.4.1.1 chipIdHi 9
3.4.1.2 chipIdLo 9
3.4.1.3 nodeVendorId 9
3.4.1.4 pszKeyword 9
3.4.1.5 reserved 9
3.4.1.6 unitSpecId 9
3.4.1.7 unitSubSWVer 9
3.4.1.8 unitSWVer 9
3.4.1.9 vendorUniqueInfo_0 9
3.4.1.10 vendorUniqueInfo_1 9
3.4.1.11 vendorUniqueInfo_2 9
3.4.1.12 vendorUniqueInfo_3 9
3.5 fc2EmbeddedImageInfo Struct Reference 10
3.5.1 Field Documentation 11
3.5.1.1 brightness 11
3.5.1.2 exposure 11

3.5.1.3 frameCounter 11
3.5.1.4 gain 11
3.5.1.5 GPIOPinState 11
3.5.1.6 ROIPosition 11
3.5.1.7 shutter 11
3.5.1.8 strobePattern 11
3.5.1.9 timestamp 11
3.5.1.10 whiteBalance 11
3.6 fc2EmbeddedImageInfoProperty Struct Reference 12
3.6.1 Field Documentation 12
3.6.1.1 available 12
3.6.1.2 onOff 12
3.7 fc2Format7ImageSettings Struct Reference 13
3.7.1 Field Documentation 13
3.7.1.1 height 13
3.7.1.2 mode 13
3.7.1.3 offsetX 13
3.7.1.4 offsetY 13
3.7.1.5 pixelFormat 13
3.7.1.6 reserved 13
3.7.1.7 width 13
3.8 fc2Format7Info Struct Reference 14
3.8.1 Field Documentation 14
3.8.1.1 imageHStepSize 14
3.8.1.2 imageVStepSize 14
3.8.1.3 maxHeight 14
3.8.1.4 maxPacketSize 14
3.8.1.5 maxWidth 14
3.8.1.6 minPacketSize 14
3.8.1.7 mode 14
3.8.1.8 offsetHStepSize 14
3.8.1.9 offsetVStepSize 14
3.8.1.10 packetSize 14
3.8.1.11 percentage 14
3.8.1.12 pixelFormatBitField 14
3.8.1.13 reserved 14

3.9 fc2Format7PacketInfo Struct Reference . . . 15
3.9.1 Field Documentation . . . 15
3.9.1.1 maxBytesPerPacket . . . 15
3.9.1.2 recommendedBytesPerPacket . . . 15
3.9.1.3 reserved . . . 15
3.9.1.4 unitBytesPerPacket . . . 15
3.10 fc2GigEConfig Struct Reference . . . 16
3.10.1 Field Documentation . . . 16
3.10.1.1 channels . . . 16
3.10.1.2 numChannels . . . 16
3.10.1.3 reserved . . . 16
3.11 fc2GigEImageSettings Struct Reference . . . 17
3.11.1 Field Documentation . . . 17
3.11.1.1 height . . . 17
3.11.1.2 offsetX . . . 17
3.11.1.3 offsetY . . . 17
3.11.1.4 pixelFormat . . . 17
3.11.1.5 reserved . . . 17
3.11.1.6 width . . . 17
3.12 fc2GigEImageSettingsInfo Struct Reference . . . 18
3.12.1 Field Documentation . . . 18
3.12.1.1 imageHStepSize . . . 18
3.12.1.2 imageVStepSize . . . 18
3.12.1.3 maxHeight . . . 18
3.12.1.4 maxWidth . . . 18
3.12.1.5 offsetHStepSize . . . 18
3.12.1.6 offsetVStepSize . . . 18
3.12.1.7 pixelFormatBitField . . . 18
3.12.1.8 reserved . . . 18
3.13 fc2GigEProperty Struct Reference . . . 19
3.13.1 Field Documentation . . . 19
3.13.1.1 isReadable . . . 19
3.13.1.2 isWritable . . . 19
3.13.1.3 max . . . 19
3.13.1.4 min . . . 19
3.13.1.5 propType . . . 19

3.13.1.6 reserved 19
3.13.1.7 value 19
3.14 fc2GigEStreamChannel Struct Reference 20
3.14.1 Field Documentation 20
3.14.1.1 destinationIpAddress 20
3.14.1.2 doNotFragment 20
3.14.1.3 hostPost 20
3.14.1.4 interPacketDelay 20
3.14.1.5 networkInterfaceIndex 20
3.14.1.6 packetSize 20
3.14.1.7 reserved 20
3.14.1.8 sourcePort 20
3.15 fc2Image Struct Reference 21
3.15.1 Field Documentation 21
3.15.1.1 bayerFormat 21
3.15.1.2 cols 21
3.15.1.3 dataSize 21
3.15.1.4 format 21
3.15.1.5 imageImpl 21
3.15.1.6 pData 21
3.15.1.7 rows 21
3.15.1.8 stride 21
3.16 fc2ImageMetadata Struct Reference 22
3.16.1 Field Documentation 22
3.16.1.1 embeddedBrightness 22
3.16.1.2 embeddedExposure 22
3.16.1.3 embeddedFrameCounter 22
3.16.1.4 embeddedGain 22
3.16.1.5 embeddedGPIOPinState 22
3.16.1.6 embeddedROIPosition 22
3.16.1.7 embeddedShutter 22
3.16.1.8 embeddedStrobePattern 22
3.16.1.9 embeddedTimeStamp 22
3.16.1.10 embeddedWhiteBalance 22
3.16.1.11 reserved 22
3.17 fc2InternalContext Struct Reference 23

3.17.1 Field Documentation 23
3.17.1.1 pBusMgr 23
3.17.1.2 pCamera 23
3.18 fc2InternalGuiContext Struct Reference 24
3.18.1 Field Documentation 24
3.18.1.1 pCameraControlDlg 24
3.18.1.2 pCameraSelectionDlg 24
3.19 fc2InternalImageCallback Struct Reference 25
3.19.1 Field Documentation 25
3.19.1.1 pCallback 25
3.19.1.2 pCallbackData 25
3.20 fc2IPAddress Struct Reference 26
3.20.1 Field Documentation 26
3.20.1.1 octets 26
3.21 fc2JPEGOption Struct Reference 27
3.21.1 Field Documentation 27
3.21.1.1 progressive 27
3.21.1.2 quality 27
3.21.1.3 reserved 27
3.22 fc2JPG2Option Struct Reference 28
3.22.1 Field Documentation 28
3.22.1.1 quality 28
3.22.1.2 reserved 28
3.23 fc2LUTData Struct Reference 29
3.23.1 Field Documentation 29
3.23.1.1 enabled 29
3.23.1.2 inputBitDepth 29
3.23.1.3 numBanks 29
3.23.1.4 numChannels 29
3.23.1.5 numEntries 29
3.23.1.6 outputBitDepth 29
3.23.1.7 reserved 29
3.23.1.8 supported 29
3.24 fc2MACAddress Struct Reference 30
3.24.1 Field Documentation 30
3.24.1.1 octets 30

3.25 fc2PGMOption Struct Reference 31
3.25.1 Field Documentation 31
3.25.1.1 binaryFile 31
3.25.1.2 reserved 31
3.26 fc2PGRGuid Struct Reference 32
3.26.1 Detailed Description 32
3.26.2 Field Documentation 32
3.26.2.1 value 32
3.27 fc2PNGOption Struct Reference 33
3.27.1 Field Documentation 33
3.27.1.1 compressionLevel 33
3.27.1.2 interlaced 33
3.27.1.3 reserved 33
3.28 fc2PPMOption Struct Reference 34
3.28.1 Field Documentation 34
3.28.1.1 binaryFile 34
3.28.1.2 reserved 34
3.29 fc2StrobeControl Struct Reference 35
3.29.1 Field Documentation 35
3.29.1.1 delay 35
3.29.1.2 duration 35
3.29.1.3 onOff 35
3.29.1.4 polarity 35
3.29.1.5 reserved 35
3.29.1.6 source 35
3.30 fc2StrobeInfo Struct Reference 36
3.30.1 Field Documentation 36
3.30.1.1 maxValue 36
3.30.1.2 minValue 36
3.30.1.3 onOffSupported 36
3.30.1.4 polaritySupported 36
3.30.1.5 present 36
3.30.1.6 readOutSupported 36
3.30.1.7 reserved 36
3.30.1.8 source 36
3.31 fc2SystemInfo Struct Reference 37

3.31.1 Field Documentation 37
3.31.1.1 byteOrder 37
3.31.1.2 cpuDescription 37
3.31.1.3 driverList 37
3.31.1.4 gpuDescription 37
3.31.1.5 libraryList 37
3.31.1.6 numCpuCores 37
3.31.1.7 osDescription 37
3.31.1.8 osType 37
3.31.1.9 reserved 37
3.31.1.10 screenHeight 37
3.31.1.11 screenWidth 37
3.31.1.12 sysMemSize 37
3.32 fc2TIFFOption Struct Reference 38
3.32.1 Field Documentation 38
3.32.1.1 compression 38
3.32.1.2 reserved 38
3.33 fc2TimeStamp Struct Reference 39
3.33.1 Field Documentation 39
3.33.1.1 cycleCount 39
3.33.1.2 cycleOffset 39
3.33.1.3 cycleSeconds 39
3.33.1.4 microSeconds 39
3.33.1.5 reserved 39
3.33.1.6 seconds 39
3.34 fc2TriggerDelay Struct Reference 40
3.34.1 Field Documentation 40
3.34.1.1 absControl 40
3.34.1.2 absValue 40
3.34.1.3 autoManualMode 40
3.34.1.4 onePush 40
3.34.1.5 onOff 40
3.34.1.6 present 40
3.34.1.7 reserved 40
3.34.1.8 type 40
3.34.1.9 valueA 40

3.34.1.10 valueB 40
3.35 fc2TriggerDelayInfo Struct Reference 41
3.35.1 Field Documentation 42
3.35.1.1 absMax 42
3.35.1.2 absMin 42
3.35.1.3 absValSupported 42
3.35.1.4 autoSupported 42
3.35.1.5 manualSupported 42
3.35.1.6 max 42
3.35.1.7 min 42
3.35.1.8 onePushSupported 42
3.35.1.9 onOffSupported 42
3.35.1.10 present 42
3.35.1.11 pUnitAbbr 42
3.35.1.12 pUnits 42
3.35.1.13 readOutSupported 42
3.35.1.14 reserved 42
3.35.1.15 type 42
3.36 fc2TriggerMode Struct Reference 43
3.36.1 Field Documentation 43
3.36.1.1 mode 43
3.36.1.2 onOff 43
3.36.1.3 parameter 43
3.36.1.4 polarity 43
3.36.1.5 reserved 43
3.36.1.6 source 43
3.37 fc2TriggerModeInfo Struct Reference 44
3.37.1 Field Documentation 44
3.37.1.1 modeMask 44
3.37.1.2 onOffSupported 44
3.37.1.3 polaritySupported 44
3.37.1.4 present 44
3.37.1.5 readOutSupported 44
3.37.1.6 reserved 44
3.37.1.7 softwareTriggerSupported 44
3.37.1.8 sourceMask 44

3.37.1.9 valueReadable 44
3.38 fc2Version Struct Reference 45
3.38.1 Field Documentation 45
3.38.1.1 build 45
3.38.1.2 major 45
3.38.1.3 minor 45
3.38.1.4 type 45

4 File Documentation 47
4.1 FlyCapture2_C.h File Reference 47
4.1.1 Function Documentation 56
4.1.1.1 fc2AVIAppend 56
4.1.1.2 fc2AVIClose 57
4.1.1.3 fc2AVIOpen 57
4.1.1.4 fc2CalculateImageStatistics 57
4.1.1.5 fc2Connect 57
4.1.1.6 fc2ConvertImage 58
4.1.1.7 fc2ConvertImageTo 58
4.1.1.8 fc2CreateAVI 58
4.1.1.9 fc2CreateContext 59
4.1.1.10 fc2CreateGigEContext 59
4.1.1.11 fc2CreateImage 59
4.1.1.12 fc2CreateImageStatistics 59
4.1.1.13 fc2DestroyAVI 60
4.1.1.14 fc2DestroyContext 60
4.1.1.15 fc2DestroyImage 60
4.1.1.16 fc2DestroyImageStatistics 60
4.1.1.17 fc2DetermineBitsPerPixel 61
4.1.1.18 fc2Disconnect 61
4.1.1.19 fc2DiscoverGigECameras 61
4.1.1.20 fc2EnableLUT 62
4.1.1.21 fc2ErrorToDescription 62
4.1.1.22 fc2FireBusReset 62
4.1.1.23 fc2FireSoftwareTrigger 62
4.1.1.24 fc2FireSoftwareTriggerBroadcast 63
4.1.1.25 fc2ForceIPAddressToCamera 63
4.1.1.26 fc2GetActiveLUTBank 63

4.1.1.27 fc2GetCameraFromIndex 64
4.1.1.28 fc2GetCameraFromSerialNumber 64
4.1.1.29 fc2GetCameraInfo 64
4.1.1.30 fc2GetCameraSerialNumberFromIndex 65
4.1.1.31 fc2GetConfiguration 65
4.1.1.32 fc2GetDefaultColorProcessing 65
4.1.1.33 fc2GetDefaultOutputFormat 65
4.1.1.34 fc2GetDeviceFromIndex 66
4.1.1.35 fc2GetEmbeddedImageInfo 66
4.1.1.36 fc2GetFormat7Configuration 66
4.1.1.37 fc2GetFormat7Info 67
4.1.1.38 fc2GetGigEImageBinningSettings 67
4.1.1.39 fc2GetGigEImageSettings 67
4.1.1.40 fc2GetGigEImageSettingsInfo 67
4.1.1.41 fc2GetGigEImagingMode 67
4.1.1.42 fc2GetGigEProperty 67
4.1.1.43 fc2GetGigEStreamChannelInfo 67
4.1.1.44 fc2GetGPIOPinDirection 67
4.1.1.45 fc2GetImageData 68
4.1.1.46 fc2GetImageStatistics 68
4.1.1.47 fc2GetImageTimeStamp 69
4.1.1.48 fc2GetInterfaceTypeFromGuid 69
4.1.1.49 fc2GetLibraryVersion 69
4.1.1.50 fc2GetLUTBankInfo 69
4.1.1.51 fc2GetLUTChannel 70
4.1.1.52 fc2GetLUTInfo 70
4.1.1.53 fc2GetMemoryChannel 70
4.1.1.54 fc2GetMemoryChannelInfo 71
4.1.1.55 fc2GetNumOfCameras 71
4.1.1.56 fc2GetNumOfDevices 71
4.1.1.57 fc2GetNumStreamChannels 71
4.1.1.58 fc2GetProperty 71
4.1.1.59 fc2GetPropertyInfo 72
4.1.1.60 fc2GetRegisterString 72
4.1.1.61 fc2GetStrobe 72
4.1.1.62 fc2GetStrobeInfo 73

4.1.1.63 fc2GetSystemInfo 73
4.1.1.64 fc2GetTriggerDelay 73
4.1.1.65 fc2GetTriggerDelayInfo 73
4.1.1.66 fc2GetTriggerMode 74
4.1.1.67 fc2GetTriggerModeInfo 74
4.1.1.68 fc2GetVideoModeAndFrameRate 74
4.1.1.69 fc2GetVideoModeAndFrameRateInfo 74
4.1.1.70 fc2LaunchBrowser 75
4.1.1.71 fc2LaunchCommand 75
4.1.1.72 fc2LaunchCommandAsync 75
4.1.1.73 fc2LaunchHelp 76
4.1.1.74 fc2QueryGigEImagingMode 76
4.1.1.75 fc2ReadGVCPMemory 76
4.1.1.76 fc2ReadGVCPRegister 76
4.1.1.77 fc2ReadGVCPRegisterBlock 77
4.1.1.78 fc2ReadRegister 77
4.1.1.79 fc2ReadRegisterBlock 77
4.1.1.80 fc2RegisterCallback 78
4.1.1.81 fc2RescanBus 78
4.1.1.82 fc2RestoreFromMemoryChannel 78
4.1.1.83 fc2RetrieveBuffer 78
4.1.1.84 fc2SaveImage 79
4.1.1.85 fc2SaveImageWithOption 79
4.1.1.86 fc2SaveToMemoryChannel 79
4.1.1.87 fc2SetActiveLUTBank 79
4.1.1.88 fc2SetCallback 80
4.1.1.89 fc2SetConfiguration 80
4.1.1.90 fc2SetDefaultColorProcessing 80
4.1.1.91 fc2SetDefaultOutputFormat 81
4.1.1.92 fc2SetEmbeddedImageInfo 81
4.1.1.93 fc2SetFormat7Configuration 81
4.1.1.94 fc2SetFormat7ConfigurationPacket 81
4.1.1.95 fc2SetGigEImageBinningSettings 82
4.1.1.96 fc2SetGigEImageSettings 82
4.1.1.97 fc2SetGigEImagingMode 82
4.1.1.98 fc2SetGigEProperty 82

4.1.1.99 fc2SetGigEStreamChannelInfo 82
4.1.1.100 fc2SetGPIOPinDirection 82
4.1.1.101 fc2SetGPIOPinDirectionBroadcast 83
4.1.1.102 fc2SetImageData 83
4.1.1.103 fc2SetImageDimensions 83
4.1.1.104 fc2SetLUTChannel 84
4.1.1.105 fc2SetProperty 84
4.1.1.106 fc2SetPropertyBroadcast 84
4.1.1.107 fc2SetStrobe 85
4.1.1.108 fc2SetStrobeBroadcast 85
4.1.1.109 fc2SetTriggerDelay 85
4.1.1.110 fc2SetTriggerDelayBroadcast 85
4.1.1.111 fc2SetTriggerMode 86
4.1.1.112 fc2SetTriggerModeBroadcast 86
4.1.1.113 fc2SetUserBuffers 86
4.1.1.114 fc2SetVideoModeAndFrameRate 87
4.1.1.115 fc2StartCapture 87
4.1.1.116 fc2StartCaptureCallback 87
4.1.1.117 fc2StartSyncCapture 88
4.1.1.118 fc2StartSyncCaptureCallback 88
4.1.1.119 fc2StopCapture 88
4.1.1.120 fc2UnregisterCallback 88
4.1.1.121 fc2ValidateFormat7Settings 89
4.1.1.122 fc2WriteGVCPMemory 89
4.1.1.123 fc2WriteGVCPRegister 89
4.1.1.124 fc2WriteGVCPRegisterBlock 90
4.1.1.125 fc2WriteGVCPRegisterBroadcast 90
4.1.1.126 fc2WriteRegister 90
4.1.1.127 fc2WriteRegisterBlock 91
4.1.1.128 fc2WriteRegisterBroadcast 91
4.2 FlyCapture2Defs_C.h File Reference 92
4.2.1 Define Documentation 100
4.2.1.1 FALSE 100
4.2.1.2 FULL_32BIT_VALUE 100
4.2.1.3 MAX_STRING_LENGTH 100
4.2.1.4 TRUE 100

4.2.2 Typedef Documentation 100
4.2.2.1 BOOL 100
4.2.2.2 fc2AsyncCommandCallback 100
4.2.2.3 fc2AVIContext 100
4.2.2.4 fc2BusEventCallback 100
4.2.2.5 fc2CallbackHandle 100
4.2.2.6 fc2Context 100
4.2.2.7 fc2GuiContext 100
4.2.2.8 fc2ImageEventCallback 101
4.2.2.9 fc2ImageImpl 101
4.2.2.10 fc2ImageStatisticsContext 101
4.2.3 Enumeration Type Documentation 101
4.2.3.1 fc2BandwidthAllocation 101
4.2.3.2 fc2BayerTileFormat 101
4.2.3.3 fc2BusCallbackType 101
4.2.3.4 fc2BusSpeed 102
4.2.3.5 fc2ByteOrder 102
4.2.3.6 fc2ColorProcessingAlgorithm 102
4.2.3.7 fc2Error 103
4.2.3.8 fc2FrameRate 104
4.2.3.9 fc2GigEPropertyType 104
4.2.3.10 fc2GrabMode 104
4.2.3.11 fc2GrabTimeout 104
4.2.3.12 fc2ImageFileFormat 105
4.2.3.13 fc2InterfaceType 105
4.2.3.14 fc2Mode 105
4.2.3.15 fc2OSType 106
4.2.3.16 fc2PixelFormat 106
4.2.3.17 fc2PropertyType 107
4.2.3.18 fc2StatisticsChannel 108
4.2.3.19 fc2TIFFCompressionMethod 108
4.2.3.20 fc2VideoMode 108
4.3 FlyCapture2GUI_C.h File Reference 110
4.3.1 Function Documentation 110
4.3.1.1 fc2CreateGUIContext 110
4.3.1.2 fc2DestroyGUIContext 111

4.3.1.3 fc2Disonnect 111
4.3.1.4 fc2GUIConnect 111
4.3.1.5 fc2Hide 111
4.3.1.6 fc2IsVisible 112
4.3.1.7 fc2Show 112
4.3.1.8 fc2ShowModal 112
4.4 FlyCapture2Internal_C.h File Reference 113
4.4.1 Function Documentation 113
4.4.1.1 IsContextValid 113
4.4.1.2 IsGuiContextValid 113
4.4.1.3 SyncCppImageToStruct 113
4.5 FlyCapture2Platform_C.h File Reference 114
4.5.1 Define Documentation 114
4.5.1.1 FLYCAPTURE2_C_API 114
4.5.1.2 FLYCAPTURE2_C_CALL_CONVEN 114

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

fc2AVIOption 5
fc2CameraInfo 6
fc2Config 8
fc2ConfigROM 9
fc2EmbeddedImageInfo 10
fc2EmbeddedImageInfoProperty 12
fc2Format7ImageSettings 13
fc2Format7Info 14
fc2Format7PacketInfo 15
fc2GigEConfig 16
fc2GigEImageSettings 17
fc2GigEImageSettingsInfo 18
fc2GigEProperty 19
fc2GigEStreamChannel 20
fc2Image 21
fc2ImageMetadata 22
fc2InternalContext 23
fc2InternalGuiContext 24
fc2InternalImageCallback 25
fc2IPAddress 26
fc2JPEGOption 27
fc2JPG2Option 28
fc2LUTData 29
fc2MACAddress 30
fc2PGMOption 31
fc2PGRGuid (A GUID to the camera) 32
fc2PNGOption 33
fc2PPMOption 34
fc2StrobeControl 35
fc2StrobeInfo 36
fc2SystemInfo 37
fc2TIFFOption 38
fc2TimeStamp 39

fc2TriggerDelay 40
fc2TriggerDelayInfo 41
fc2TriggerMode 43
fc2TriggerModeInfo 44
fc2Version 45

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

FlyCapture2_C.h 47
FlyCapture2Defs_C.h 92
FlyCapture2GUI_C.h 110
FlyCapture2Internal_C.h 113
FlyCapture2Platform_C.h 114

Chapter 3

Data Structure Documentation

3.1 fc2AVIOption Struct Reference

Data Fields

- float frameRate
- unsigned int reserved [256]

3.1.1 Field Documentation

3.1.1.1 float frameRate

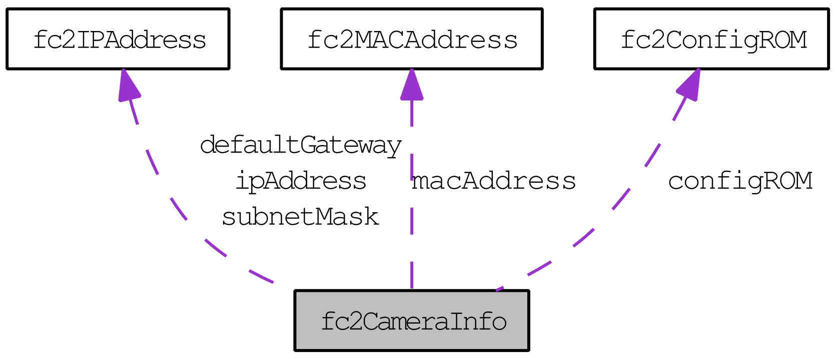
3.1.1.2 unsigned int reserved[256]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.2 fc2CameraInfo Struct Reference

Collaboration diagram for fc2CameraInfo:



Data Fields

- unsigned int serialNumber
- fc2InterfaceType interfaceType
- BOOL isColorCamera
- char modelName [MAX_STRING_LENGTH]
- char vendorName [MAX_STRING_LENGTH]
- char sensorInfo [MAX_STRING_LENGTH]
- char sensorResolution [MAX_STRING_LENGTH]
- char driverName [MAX_STRING_LENGTH]
- char firmwareVersion [MAX_STRING_LENGTH]
- char firmwareBuildTime [MAX_STRING_LENGTH]
- fc2BusSpeed maximumBusSpeed
- fc2BayerTileFormat bayerTileFormat
- unsigned int iidcVer
- fc2ConfigROM configROM
- unsigned int gigEMajorVersion
- unsigned int gigEMinorVersion
- char userDefinedName [MAX_STRING_LENGTH]
- char xmlURL1 [MAX_STRING_LENGTH]
- char xmlURL2 [MAX_STRING_LENGTH]
- fc2MACAddress macAddress
- fc2IPAddress ipAddress
- fc2IPAddress subnetMask
- fc2IPAddress defaultGateway
- unsigned int reserved [16]

3.2.1 Field Documentation

3.2.1.1 fc2BayerTileFormat bayerTileFormat

3.2.1.2 fc2ConfigROM configROM

3.2.1.3 fc2IPAddress defaultGateway

3.2.1.4 char driverName[MAX_STRING_LENGTH]

3.2.1.5 char firmwareBuildTime[MAX_STRING_LENGTH]

3.2.1.6 char firmwareVersion[MAX_STRING_LENGTH]

3.2.1.7 unsigned int gigEMajorVersion

3.2.1.8 unsigned int gigEMinorVersion

3.2.1.9 unsigned int iidcVer

3.2.1.10 fc2InterfaceType interfaceType

3.2.1.11 fc2IPAddress ipAddress

3.2.1.12 BOOL isColorCamera

3.2.1.13 fc2MACAddress macAddress

3.2.1.14 fc2BusSpeed maximumBusSpeed

3.2.1.15 char modelName[MAX_STRING_LENGTH]

3.2.1.16 unsigned int reserved[16]

3.2.1.17 char sensorInfo[MAX_STRING_LENGTH]

3.2.1.18 char sensorResolution[MAX_STRING_LENGTH]

3.2.1.19 unsigned int serialNumber

3.2.1.20 fc2IPAddress subnetMask

3.2.1.21 char userDefinedName[MAX_STRING_LENGTH]

3.2.1.22 char vendorName[MAX_STRING_LENGTH]

3.2.1.23 char xmlURL1[MAX_STRING_LENGTH]

3.2.1.24 char xmlURL2[MAX_STRING_LENGTH]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.3 fc2Config Struct Reference

Data Fields

- unsigned int numBuffers
- unsigned int numImageNotifications
- int grabTimeout
- fc2GrabMode grabMode
- fc2BusSpeed isochBusSpeed
- fc2BusSpeed asyncBusSpeed
- fc2BandwidthAllocation bandwidthAllocation
- unsigned int reserved [16]

3.3.1 Field Documentation

3.3.1.1 fc2BusSpeed asyncBusSpeed

3.3.1.2 fc2BandwidthAllocation bandwidthAllocation

3.3.1.3 fc2GrabMode grabMode

3.3.1.4 int grabTimeout

3.3.1.5 fc2BusSpeed isochBusSpeed

3.3.1.6 unsigned int numBuffers

3.3.1.7 unsigned int numImageNotifications

3.3.1.8 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.4 fc2ConfigROM Struct Reference

Data Fields

- unsigned int nodeVendorId
- unsigned int chipIdHi
- unsigned int chipIdLo
- unsigned int unitSpecId
- unsigned int unitSWVer
- unsigned int unitSubSWVer
- unsigned int vendorUniqueInfo_0
- unsigned int vendorUniqueInfo_1
- unsigned int vendorUniqueInfo_2
- unsigned int vendorUniqueInfo_3
- char pszKeyword [MAX_STRING_LENGTH]
- unsigned int reserved [16]

3.4.1 Field Documentation

3.4.1.1 unsigned int chipIdHi

3.4.1.2 unsigned int chipIdLo

3.4.1.3 unsigned int nodeVendorId

3.4.1.4 char pszKeyword[MAX_STRING_LENGTH]

3.4.1.5 unsigned int reserved[16]

3.4.1.6 unsigned int unitSpecId

3.4.1.7 unsigned int unitSubSWVer

3.4.1.8 unsigned int unitSWVer

3.4.1.9 unsigned int vendorUniqueInfo_0

3.4.1.10 unsigned int vendorUniqueInfo_1

3.4.1.11 unsigned int vendorUniqueInfo_2

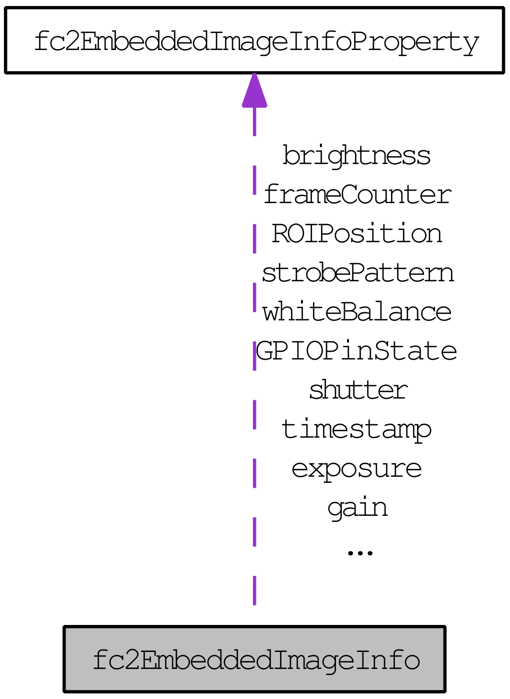
3.4.1.12 unsigned int vendorUniqueInfo_3

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.5 fc2EmbeddedImageInfo Struct Reference

Collaboration diagram for fc2EmbeddedImageInfo:



Data Fields

- fc2EmbeddedImageInfoProperty timestamp
- fc2EmbeddedImageInfoProperty gain
- fc2EmbeddedImageInfoProperty shutter
- fc2EmbeddedImageInfoProperty brightness
- fc2EmbeddedImageInfoProperty exposure
- fc2EmbeddedImageInfoProperty whiteBalance
- fc2EmbeddedImageInfoProperty frameCounter
- fc2EmbeddedImageInfoProperty strobePattern
- fc2EmbeddedImageInfoProperty GPIOPinState
- fc2EmbeddedImageInfoProperty ROIPosition

3.5.1 Field Documentation

3.5.1.1 fc2EmbeddedImageInfoProperty brightness

3.5.1.2 fc2EmbeddedImageInfoProperty exposure

3.5.1.3 fc2EmbeddedImageInfoProperty frameCounter

3.5.1.4 fc2EmbeddedImageInfoProperty gain

3.5.1.5 fc2EmbeddedImageInfoProperty GPIOPinState

3.5.1.6 fc2EmbeddedImageInfoProperty ROIPosition

3.5.1.7 fc2EmbeddedImageInfoProperty shutter

3.5.1.8 fc2EmbeddedImageInfoProperty strobePattern

3.5.1.9 fc2EmbeddedImageInfoProperty timestamp

3.5.1.10 fc2EmbeddedImageInfoProperty whiteBalance

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.6 fc2EmbeddedImageInfoProperty Struct Reference

Data Fields

- BOOL available
- BOOL onOff

3.6.1 Field Documentation

3.6.1.1 BOOL available

3.6.1.2 BOOL onOff

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.7 fc2Format7ImageSettings Struct Reference

Data Fields

- fc2Mode mode
- unsigned int offsetX
- unsigned int offsetY
- unsigned int width
- unsigned int height
- fc2PixelFormat pixelFormat
- unsigned int reserved [8]

3.7.1 Field Documentation

3.7.1.1 unsigned int height

3.7.1.2 fc2Mode mode

3.7.1.3 unsigned int offsetX

3.7.1.4 unsigned int offsetY

3.7.1.5 fc2PixelFormat pixelFormat

3.7.1.6 unsigned int reserved[8]

3.7.1.7 unsigned int width

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.8 fc2Format7Info Struct Reference

Data Fields

- fc2Mode mode
- unsigned int maxWidth
- unsigned int maxHeight
- unsigned int offsetHStepSize
- unsigned int offsetVStepSize
- unsigned int imageHStepSize
- unsigned int imageVStepSize
- unsigned int pixelFormatBitField
- unsigned int packetSize
- unsigned int minPacketSize
- unsigned int maxPacketSize
- float percentage
- unsigned int reserved [16]

3.8.1 Field Documentation

3.8.1.1 unsigned int imageHStepSize

3.8.1.2 unsigned int imageVStepSize

3.8.1.3 unsigned int maxHeight

3.8.1.4 unsigned int maxPacketSize

3.8.1.5 unsigned int maxWidth

3.8.1.6 unsigned int minPacketSize

3.8.1.7 fc2Mode mode

3.8.1.8 unsigned int offsetHStepSize

3.8.1.9 unsigned int offsetVStepSize

3.8.1.10 unsigned int packetSize

3.8.1.11 float percentage

3.8.1.12 unsigned int pixelFormatBitField

3.8.1.13 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.9 fc2Format7PacketInfo Struct Reference

Data Fields

- unsigned int recommendedBytesPerPacket
- unsigned int maxBytesPerPacket
- unsigned int unitBytesPerPacket
- unsigned int reserved [8]

3.9.1 Field Documentation

3.9.1.1 unsigned int maxBytesPerPacket

3.9.1.2 unsigned int recommendedBytesPerPacket

3.9.1.3 unsigned int reserved[8]

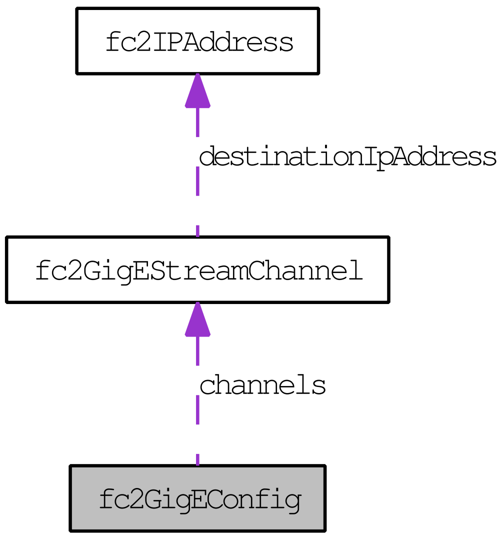
3.9.1.4 unsigned int unitBytesPerPacket

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.10 fc2GigEConfig Struct Reference

Collaboration diagram for fc2GigEConfig:



Data Fields

- unsigned int numChannels
- fc2GigEStreamChannel channels [512]
- unsigned int reserved [8]

3.10.1 Field Documentation

3.10.1.1 fc2GigEStreamChannel channels[512]

3.10.1.2 unsigned int numChannels

3.10.1.3 unsigned int reserved[8]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.11 fc2GigEImageSettings Struct Reference

Data Fields

- unsigned int offsetX
- unsigned int offsetY
- unsigned int width
- unsigned int height
- fc2PixelFormat pixelFormat
- unsigned int reserved [8]

3.11.1 Field Documentation

3.11.1.1 unsigned int height

3.11.1.2 unsigned int offsetX

3.11.1.3 unsigned int offsetY

3.11.1.4 fc2PixelFormat pixelFormat

3.11.1.5 unsigned int reserved[8]

3.11.1.6 unsigned int width

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.12 fc2GigEImageSettingsInfo Struct Reference

Data Fields

- unsigned int maxWidth
- unsigned int maxHeight
- unsigned int offsetHStepSize
- unsigned int offsetVStepSize
- unsigned int imageHStepSize
- unsigned int imageVStepSize
- unsigned int pixelFormatBitField
- unsigned int reserved [16]

3.12.1 Field Documentation

3.12.1.1 unsigned int imageHStepSize

3.12.1.2 unsigned int imageVStepSize

3.12.1.3 unsigned int maxHeight

3.12.1.4 unsigned int maxWidth

3.12.1.5 unsigned int offsetHStepSize

3.12.1.6 unsigned int offsetVStepSize

3.12.1.7 unsigned int pixelFormatBitField

3.12.1.8 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.13 fc2GigEProperty Struct Reference

Data Fields

- fc2GigEPropertyType propType
- BOOL isReadable
- BOOL isWritable
- unsigned int min
- unsigned int max
- unsigned int value
- unsigned int reserved [8]

3.13.1 Field Documentation

3.13.1.1 BOOL isReadable

3.13.1.2 BOOL isWritable

3.13.1.3 unsigned int max

3.13.1.4 unsigned int min

3.13.1.5 fc2GigEPropertyType propType

3.13.1.6 unsigned int reserved[8]

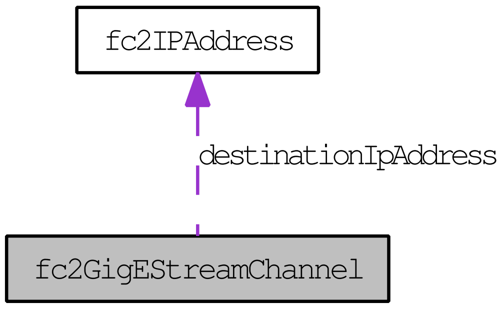
3.13.1.7 unsigned int value

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.14 fc2GigEStreamChannel Struct Reference

Collaboration diagram for fc2GigEStreamChannel:



Data Fields

- unsigned int networkInterfaceIndex
- unsigned int hostPost
- BOOL doNotFragment
- unsigned int packetSize
- unsigned int interPacketDelay
- fc2IPAddress destinationIpAddress
- unsigned int sourcePort
- unsigned int reserved [8]

3.14.1 Field Documentation

3.14.1.1 fc2IPAddress destinationIpAddress

3.14.1.2 BOOL doNotFragment

3.14.1.3 unsigned int hostPost

3.14.1.4 unsigned int interPacketDelay

3.14.1.5 unsigned int networkInterfaceIndex

3.14.1.6 unsigned int packetSize

3.14.1.7 unsigned int reserved[8]

3.14.1.8 unsigned int sourcePort

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.15 fc2Image Struct Reference

Data Fields

- unsigned int rows
- unsigned int cols
- unsigned int stride
- unsigned char ∗ pData
- unsigned int dataSize
- fc2PixelFormat format
- fc2BayerTileFormat bayerFormat
- fc2ImageImpl imageImpl

3.15.1 Field Documentation

3.15.1.1 fc2BayerTileFormat bayerFormat

3.15.1.2 unsigned int cols

3.15.1.3 unsigned int dataSize

3.15.1.4 fc2PixelFormat format

3.15.1.5 fc2ImageImpl imageImpl

3.15.1.6 unsigned char∗ pData

3.15.1.7 unsigned int rows

3.15.1.8 unsigned int stride

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.16 fc2ImageMetadata Struct Reference

Data Fields

- unsigned int embeddedTimeStamp
- unsigned int embeddedGain
- unsigned int embeddedShutter
- unsigned int embeddedBrightness
- unsigned int embeddedExposure
- unsigned int embeddedWhiteBalance
- unsigned int embeddedFrameCounter
- unsigned int embeddedStrobePattern
- unsigned int embeddedGPIOPinState
- unsigned int embeddedROIPosition
- unsigned int reserved [31]

3.16.1 Field Documentation

3.16.1.1 unsigned int embeddedBrightness

3.16.1.2 unsigned int embeddedExposure

3.16.1.3 unsigned int embeddedFrameCounter

3.16.1.4 unsigned int embeddedGain

3.16.1.5 unsigned int embeddedGPIOPinState

3.16.1.6 unsigned int embeddedROIPosition

3.16.1.7 unsigned int embeddedShutter

3.16.1.8 unsigned int embeddedStrobePattern

3.16.1.9 unsigned int embeddedTimeStamp

3.16.1.10 unsigned int embeddedWhiteBalance

3.16.1.11 unsigned int reserved[31]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.17 fc2InternalContext Struct Reference

Data Fields

- FlyCapture2::BusManager ∗ pBusMgr
- FlyCapture2::CameraBase ∗ pCamera

3.17.1 Field Documentation

3.17.1.1 FlyCapture2::BusManager∗ pBusMgr

3.17.1.2 FlyCapture2::CameraBase∗ pCamera

The documentation for this struct was generated from the following file:

- FlyCapture2Internal_C.h

3.18 fc2InternalGuiContext Struct Reference

Data Fields

- FlyCapture2::CameraSelectionDlg ∗ pCameraSelectionDlg
- FlyCapture2::CameraControlDlg ∗ pCameraControlDlg

3.18.1 Field Documentation

3.18.1.1 FlyCapture2::CameraControlDlg∗ pCameraControlDlg

3.18.1.2 FlyCapture2::CameraSelectionDlg∗ pCameraSelectionDlg

The documentation for this struct was generated from the following file:

- FlyCapture2Internal_C.h

3.19 fc2InternalImageCallback Struct Reference

Data Fields

- fc2ImageEventCallback pCallback
- void * pCallbackData

3.19.1 Field Documentation

3.19.1.1 fc2ImageEventCallback pCallback

3.19.1.2 void* pCallbackData

The documentation for this struct was generated from the following file:

- FlyCapture2Internal_C.h

3.20 fc2IPAddress Struct Reference

Data Fields

- unsigned char octets [4]

3.20.1 Field Documentation

3.20.1.1 unsigned char octets[4]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.21 fc2JPEGOption Struct Reference

Data Fields

- BOOL progressive
- unsigned int quality
- unsigned int reserved [16]

3.21.1 Field Documentation

3.21.1.1 BOOL progressive

3.21.1.2 unsigned int quality

3.21.1.3 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.22 fc2JPG2Option Struct Reference

Data Fields

- unsigned int quality
- unsigned int reserved [16]

3.22.1 Field Documentation

3.22.1.1 unsigned int quality

3.22.1.2 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.23 fc2LUTData Struct Reference

Data Fields

- BOOL supported
- BOOL enabled
- unsigned int numBanks
- unsigned int numChannels
- unsigned int inputBitDepth
- unsigned int outputBitDepth
- unsigned int numEntries
- unsigned int reserved [8]

3.23.1 Field Documentation

3.23.1.1 BOOL enabled

3.23.1.2 unsigned int inputBitDepth

3.23.1.3 unsigned int numBanks

3.23.1.4 unsigned int numChannels

3.23.1.5 unsigned int numEntries

3.23.1.6 unsigned int outputBitDepth

3.23.1.7 unsigned int reserved[8]

3.23.1.8 BOOL supported

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.24 fc2MACAddress Struct Reference

Data Fields

- unsigned char octets [6]

3.24.1 Field Documentation

3.24.1.1 unsigned char octets[6]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.25 fc2PGMOption Struct Reference

Data Fields

- BOOL binaryFile
- unsigned int reserved [16]

3.25.1 Field Documentation

3.25.1.1 BOOL binaryFile

3.25.1.2 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.26 fc2PGRGuid Struct Reference

A GUID to the camera.

Data Fields

- unsigned int value [4]

3.26.1 Detailed Description

A GUID to the camera.

It is used to uniquely identify a camera.

3.26.2 Field Documentation

3.26.2.1 unsigned int value[4]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.27 fc2PNGOption Struct Reference

Data Fields

- BOOL interlaced
- unsigned int compressionLevel
- unsigned int reserved [16]

3.27.1 Field Documentation

3.27.1.1 unsigned int compressionLevel

3.27.1.2 BOOL interlaced

3.27.1.3 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.28 fc2PPMOption Struct Reference

Data Fields

- BOOL binaryFile
- unsigned int reserved [16]

3.28.1 Field Documentation

3.28.1.1 BOOL binaryFile

3.28.1.2 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.29 fc2StrobeControl Struct Reference

Data Fields

- unsigned int source
- BOOL onOff
- unsigned int polarity
- float delay
- float duration
- unsigned int reserved [8]

3.29.1 Field Documentation

3.29.1.1 float delay

3.29.1.2 float duration

3.29.1.3 BOOL onOff

3.29.1.4 unsigned int polarity

3.29.1.5 unsigned int reserved[8]

3.29.1.6 unsigned int source

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.30 fc2StrobeInfo Struct Reference

Data Fields

- unsigned int source
- BOOL present
- BOOL readOutSupported
- BOOL onOffSupported
- BOOL polaritySupported
- float minValue
- float maxValue
- unsigned int reserved [8]

3.30.1 Field Documentation

3.30.1.1 float maxValue

3.30.1.2 float minValue

3.30.1.3 BOOL onOffSupported

3.30.1.4 BOOL polaritySupported

3.30.1.5 BOOL present

3.30.1.6 BOOL readOutSupported

3.30.1.7 unsigned int reserved[8]

3.30.1.8 unsigned int source

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.31 fc2SystemInfo Struct Reference

Data Fields

- fc2OSType osType
- char osDescription [MAX_STRING_LENGTH]
- fc2ByteOrder byteOrder
- size_t sysMemSize
- char cpuDescription [MAX_STRING_LENGTH]
- size_t numCpuCores
- char driverList [MAX_STRING_LENGTH]
- char libraryList [MAX_STRING_LENGTH]
- char gpuDescription [MAX_STRING_LENGTH]
- size_t screenWidth
- size_t screenHeight
- unsigned int reserved [16]

3.31.1 Field Documentation

3.31.1.1 fc2ByteOrder byteOrder

3.31.1.2 char cpuDescription[MAX_STRING_LENGTH]

3.31.1.3 char driverList[MAX_STRING_LENGTH]

3.31.1.4 char gpuDescription[MAX_STRING_LENGTH]

3.31.1.5 char libraryList[MAX_STRING_LENGTH]

3.31.1.6 size_t numCpuCores

3.31.1.7 char osDescription[MAX_STRING_LENGTH]

3.31.1.8 fc2OSType osType

3.31.1.9 unsigned int reserved[16]

3.31.1.10 size_t screenHeight

3.31.1.11 size_t screenWidth

3.31.1.12 size_t sysMemSize

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.32 fc2TIFFOption Struct Reference

Data Fields

- fc2TIFFCompressionMethod compression
- unsigned int reserved [16]

3.32.1 Field Documentation

3.32.1.1 fc2TIFFCompressionMethod compression

3.32.1.2 unsigned int reserved[16]

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.33 fc2TimeStamp Struct Reference

Data Fields

- long long seconds
- unsigned int microSeconds
- unsigned int cycleSeconds
- unsigned int cycleCount
- unsigned int cycleOffset
- unsigned int reserved [8]

3.33.1 Field Documentation

3.33.1.1 unsigned int cycleCount

3.33.1.2 unsigned int cycleOffset

3.33.1.3 unsigned int cycleSeconds

3.33.1.4 unsigned int microSeconds

3.33.1.5 unsigned int reserved[8]

3.33.1.6 long long seconds

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.34 fc2TriggerDelay Struct Reference

Data Fields

- fc2PropertyType type
- BOOL present
- BOOL absControl
- BOOL onePush
- BOOL onOff
- BOOL autoManualMode
- unsigned int valueA
- unsigned int valueB
- float absValue
- unsigned int reserved [8]

3.34.1 Field Documentation

3.34.1.1 BOOL absControl

3.34.1.2 float absValue

3.34.1.3 BOOL autoManualMode

3.34.1.4 BOOL onePush

3.34.1.5 BOOL onOff

3.34.1.6 BOOL present

3.34.1.7 unsigned int reserved[8]

3.34.1.8 fc2PropertyType type

3.34.1.9 unsigned int valueA

3.34.1.10 unsigned int valueB

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.35 fc2TriggerDelayInfo Struct Reference

Data Fields

- fc2PropertyType type
- BOOL present
- BOOL autoSupported
- BOOL manualSupported
- BOOL onOffSupported
- BOOL onePushSupported
- BOOL absValSupported
- BOOL readOutSupported
- unsigned int min
- unsigned int max
- float absMin
- float absMax
- char pUnits [MAX_STRING_LENGTH]
- char pUnitAbbr [MAX_STRING_LENGTH]
- unsigned int reserved [8]

3.35.1 Field Documentation

3.35.1.1 float absMax

3.35.1.2 float absMin

3.35.1.3 BOOL absValSupported

3.35.1.4 BOOL autoSupported

3.35.1.5 BOOL manualSupported

3.35.1.6 unsigned int max

3.35.1.7 unsigned int min

3.35.1.8 BOOL onePushSupported

3.35.1.9 BOOL onOffSupported

3.35.1.10 BOOL present

3.35.1.11 char pUnitAbbr[MAX_STRING_LENGTH]

3.35.1.12 char pUnits[MAX_STRING_LENGTH]

3.35.1.13 BOOL readOutSupported

3.35.1.14 unsigned int reserved[8]

3.35.1.15 fc2PropertyType type

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.36 fc2TriggerMode Struct Reference

Data Fields

- BOOL onOff
- unsigned int polarity
- unsigned int source
- unsigned int mode
- unsigned int parameter
- unsigned int reserved [8]

3.36.1 Field Documentation

3.36.1.1 unsigned int mode

3.36.1.2 BOOL onOff

3.36.1.3 unsigned int parameter

3.36.1.4 unsigned int polarity

3.36.1.5 unsigned int reserved[8]

3.36.1.6 unsigned int source

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.37 fc2TriggerModeInfo Struct Reference

Data Fields

- BOOL present
- BOOL readOutSupported
- BOOL onOffSupported
- BOOL polaritySupported
- BOOL valueReadable
- unsigned int sourceMask
- BOOL softwareTriggerSupported
- unsigned int modeMask
- unsigned int reserved [8]

3.37.1 Field Documentation

3.37.1.1 unsigned int modeMask

3.37.1.2 BOOL onOffSupported

3.37.1.3 BOOL polaritySupported

3.37.1.4 BOOL present

3.37.1.5 BOOL readOutSupported

3.37.1.6 unsigned int reserved[8]

3.37.1.7 BOOL softwareTriggerSupported

3.37.1.8 unsigned int sourceMask

3.37.1.9 BOOL valueReadable

The documentation for this struct was generated from the following file:

- FlyCapture2Defs_C.h

3.38 fc2Version Struct Reference

Data Fields

- unsigned int major
- unsigned int minor
- unsigned int type
- unsigned int build

3.38.1 Field Documentation

3.38.1.1 unsigned int build

3.38.1.2 unsigned int major

3.38.1.3 unsigned int minor

3.38.1.4 unsigned int type

The documentation for this struct was generated from the following file:

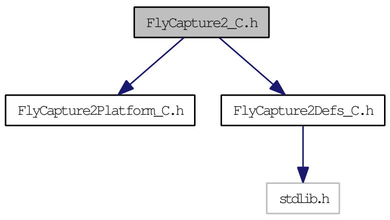
- FlyCapture2Defs_C.h

Chapter 4

File Documentation

4.1 FlyCapture2_C.h File Reference

Include dependency graph for FlyCapture2_C.h:



Functions

- FLYCAPTURE2_C_API fc2Error fc2CreateContext (fc2Context ∗pContext)

  Create a FC2 context for IIDC camaera.

- FLYCAPTURE2_C_API fc2Error fc2CreateGigEContext (fc2Context ∗pContext)

  Create a FC2 context for a GigE Vision camera.

- FLYCAPTURE2_C_API fc2Error fc2DestroyContext (fc2Context context)

  Destroy the FC2 context.

- FLYCAPTURE2_C_API fc2Error fc2FireBusReset (fc2Context context, fc2PGRGuid ∗pGuid)

  Fire a bus reset.

- FLYCAPTURE2_C_API fc2Error fc2GetNumOfCameras (fc2Context context, unsigned int ∗pNumCameras)

  Gets the number of cameras attached to the PC.

- FLYCAPTURE2_C_API fc2Error fc2GetCameraFromIndex (fc2Context context, unsigned int index, fc2PGRGuid ∗pGuid)

  Gets the PGRGuid for a camera on the PC.

- FLYCAPTURE2_C_API fc2Error fc2GetCameraFromSerialNumber (fc2Context context, unsigned int serialNumber, fc2PGRGuid ∗pGuid)

  Gets the PGRGuid for a camera on the PC.

- FLYCAPTURE2_C_API fc2Error fc2GetCameraSerialNumberFromIndex (fc2Context context, unsigned int index, unsigned int ∗pSerialNumber)

  Gets the serial number of the camera with the specified index.

- FLYCAPTURE2_C_API fc2Error fc2GetInterfaceTypeFromGuid (fc2Context context, fc2PGRGuid ∗pGuid, fc2InterfaceType ∗pInterfaceType)

  Gets the interface type associated with a PGRGuid.

- FLYCAPTURE2_C_API fc2Error fc2GetNumOfDevices (fc2Context context, unsigned int ∗pNumDevices)

  Gets the number of devices.

- FLYCAPTURE2_C_API fc2Error fc2GetDeviceFromIndex (fc2Context context, unsigned int index, fc2PGRGuid ∗pGuid)

  Gets the PGRGuid for a device.

- FLYCAPTURE2_C_API fc2Error fc2RegisterCallback (fc2Context context, fc2BusEventCallback enumCallback, fc2BusCallbackType callbackType, void ∗pParameter, fc2CallbackHandle ∗pCallbackHandle)

  Register a callback function that will be called when the specified callback event occurs.

- FLYCAPTURE2_C_API fc2Error fc2UnregisterCallback (fc2Context context, fc2CallbackHandle callbackHandle)

  Unregister a callback function.

- FLYCAPTURE2_C_API fc2Error fc2RescanBus (fc2Context context)

  Force a rescan of the buses.

- FLYCAPTURE2_C_API fc2Error fc2ForceIPAddressToCamera (fc2Context context, fc2MACAddress macAddress, fc2IPAddress ipAddress, fc2IPAddress subnetMask, fc2IPAddress defaultGateway)

  Force the camera with the specific MAC address to the specified IP address, subnet mask and default gateway.

- FLYCAPTURE2_C_API fc2Error fc2DiscoverGigECameras (fc2Context context, fc2CameraInfo ∗gigECameras, unsigned int ∗arraySize)

  Discover all cameras connected to the network even if they reside on a different subnet.

- FLYCAPTURE2_C_API fc2Error fc2WriteRegister (fc2Context context, unsigned int address, unsigned int value)

  Write to the specified register on the camera.

- FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBroadcast (fc2Context context, unsigned int address, unsigned int value)

  Write to the specified register on the camera with broadcast.

- FLYCAPTURE2_C_API fc2Error fc2ReadRegister (fc2Context context, unsigned int address, unsigned int ∗pValue)

  Read the specified register from the camera.

- FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBlock (fc2Context context, unsigned short addressHigh, unsigned int addressLow, const unsigned int ∗pBuffer, unsigned int length)

  Write to the specified register block on the camera.

- FLYCAPTURE2_C_API fc2Error fc2ReadRegisterBlock (fc2Context context, unsigned short addressHigh, unsigned int addressLow, unsigned int ∗pBuffer, unsigned int length)

  Write to the specified register block on the camera.

- FLYCAPTURE2_C_API fc2Error fc2Connect (fc2Context context, fc2PGRGuid ∗guid)

  Connects the camera object to the camera specified by the GUID.

- FLYCAPTURE2_C_API fc2Error fc2Disconnect (fc2Context context)

  Disconnects the fc2Context from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetCallback (fc2Context context, fc2ImageEventCallback pCallbackFn, void ∗pCallbackData)

  Sets the callback data to be used on completion of image transfer.

- FLYCAPTURE2_C_API fc2Error fc2StartCapture (fc2Context context)

  Starts isochronous image capture.

- FLYCAPTURE2_C_API fc2Error fc2StartCaptureCallback (fc2Context context, fc2ImageEventCallback pCallbackFn, void ∗pCallbackData)

  Starts isochronous image capture.

- FLYCAPTURE2_C_API fc2Error fc2StartSyncCapture (unsigned int numCameras, fc2Context ∗pContexts)

  Starts synchronized isochronous image capture on multiple cameras.

- FLYCAPTURE2_C_API fc2Error fc2StartSyncCaptureCallback (unsigned int numCameras, fc2Context ∗pContexts, fc2ImageEventCallback ∗pCallbackFns, void ∗∗pCallbackDataArray)

  Starts synchronized isochronous image capture on multiple cameras.

- FLYCAPTURE2_C_API fc2Error fc2RetrieveBuffer (fc2Context context, fc2Image ∗pImage)

  Retrieves the the next image object containing the next image.

- FLYCAPTURE2_C_API fc2Error fc2StopCapture (fc2Context context)

  Stops isochronous image transfer and cleans up all associated resources.

- FLYCAPTURE2_C_API fc2Error fc2SetUserBuffers (fc2Context context, unsigned char ∗const ppMemBuffers, int size, int nNumBuffers)

  Specify user allocated buffers to use as image data buffers.

- FLYCAPTURE2_C_API fc2Error fc2GetConfiguration (fc2Context context, fc2Config ∗config)

  Get the configuration associated with the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetConfiguration (fc2Context context, fc2Config ∗config)

  Set the configuration associated with the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetCameraInfo (fc2Context context, fc2CameraInfo ∗pCameraInfo)

  Retrieves information from the camera such as serial number, model name and other camera information.

- FLYCAPTURE2_C_API fc2Error fc2GetPropertyInfo (fc2Context context, fc2PropertyInfo ∗propInfo)

  Retrieves information about the specified camera property.

- FLYCAPTURE2_C_API fc2Error fc2GetProperty (fc2Context context, fc2Property ∗prop)

  Reads the settings for the specified property from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetProperty (fc2Context context, fc2Property ∗prop)

  Writes the settings for the specified property to the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetPropertyBroadcast (fc2Context context, fc2Property ∗prop)

  Writes the settings for the specified property to the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetGPIOPinDirection (fc2Context context, unsigned int pin, unsigned int ∗pDirection)

  Get the GPIO pin direction for the specified pin.

- FLYCAPTURE2_C_API fc2Error fc2SetGPIOPinDirection (fc2Context context, unsigned int pin, unsigned int direction)

  Set the GPIO pin direction for the specified pin.

- FLYCAPTURE2_C_API fc2Error fc2SetGPIOPinDirectionBroadcast (fc2Context context, unsigned int pin, unsigned int direction)

  Set the GPIO pin direction for the specified pin.

- FLYCAPTURE2_C_API fc2Error fc2GetTriggerModeInfo (fc2Context context, fc2TriggerModeInfo ∗triggerModeInfo)

  Retrieve trigger information from the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetTriggerMode (fc2Context context, fc2TriggerMode ∗triggerMode)

  Retrieve current trigger settings from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetTriggerMode (fc2Context context, fc2TriggerMode ∗triggerMode)

  Set the specified trigger settings to the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetTriggerModeBroadcast (fc2Context context, fc2TriggerMode ∗triggerMode)

  Set the specified trigger settings to the camera.

- FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTrigger (fc2Context context)
- FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTriggerBroadcast (fc2Context context)

Fire the software trigger according to the DCAM specifications.

- FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelayInfo (fc2Context context, fc2TriggerDelayInfo ∗triggerDelayInfo)

  Retrieve trigger delay information from the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelay (fc2Context context, fc2TriggerDelay ∗triggerDelay)

  Retrieve current trigger delay settings from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelay (fc2Context context, fc2TriggerDelay ∗triggerDelay)

  Set the specified trigger delay settings to the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelayBroadcast (fc2Context context, fc2TriggerDelay ∗triggerDelay)

  Set the specified trigger delay settings to the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetStrobeInfo (fc2Context context, fc2StrobeInfo ∗strobeInfo)

  Retrieve strobe information from the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetStrobe (fc2Context context, fc2StrobeControl ∗strobeControl)

  Retrieve current strobe settings from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetStrobe (fc2Context context, fc2StrobeControl ∗strobeControl)

  Set current strobe settings to the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetStrobeBroadcast (fc2Context context, fc2StrobeControl ∗strobeControl)

  Set current strobe settings to the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRateInfo (fc2Context context, fc2VideoMode videoMode, fc2FrameRate frameRate, BOOL ∗pSupported)

  Query the camera to determine if the specified video mode and frame rate is supported.

- FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRate (fc2Context context, fc2VideoMode ∗videoMode, fc2FrameRate ∗frameRate)

  Get the current video mode and frame rate from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetVideoModeAndFrameRate (fc2Context context, fc2VideoMode videoMode, fc2FrameRate frameRate)

  Set the specified video mode and frame rate to the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetFormat7Info (fc2Context context, fc2Format7Info ∗info, BOOL ∗pSupported)

  Retrieve the availability of Format7 custom image mode and the camera capabilities for the specified Format7 mode.

- FLYCAPTURE2_C_API fc2Error fc2ValidateFormat7Settings (fc2Context context, fc2Format7ImageSettings ∗imageSettings, BOOL ∗settingsAreValid, fc2Format7PacketInfo ∗packetInfo)

  Validates Format7ImageSettings structure and returns valid packet size information if the image settings are valid.

- FLYCAPTURE2_C_API fc2Error fc2GetFormat7Configuration (fc2Context context, fc2Format7ImageSettings ∗imageSettings, unsigned int ∗packetSize, float ∗percentage)

  Get the current Format7 configuration from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetFormat7ConfigurationPacket (fc2Context context, fc2Format7ImageSettings ∗imageSettings, unsigned int packetSize)

  Set the current Format7 configuration to the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetFormat7Configuration (fc2Context context, fc2Format7ImageSettings ∗imageSettings, float percentSpeed)

  Set the current Format7 configuration to the camera.

- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegister (fc2Context context, unsigned int address, unsigned int value)

  Write a GVCP register.

- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBroadcast (fc2Context context, unsigned int address, unsigned int value)

  Write a GVCP register with broadcast.

- FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegister (fc2Context context, unsigned int address, unsigned int ∗pValue)

  Read a GVCP register.

- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBlock (fc2Context context, unsigned int address, const unsigned int ∗pBuffer, unsigned int length)

  Write a GVCP register block.

- FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegisterBlock (fc2Context context, unsigned int address, unsigned int ∗pBuffer, unsigned int length)

  Read a GVCP register block.

- FLYCAPTURE2_C_API fc2Error fc2WriteGVCPMemory (fc2Context context, unsigned int address, const unsigned char ∗pBuffer, unsigned int length)

  Write a GVCP memory block.

- FLYCAPTURE2_C_API fc2Error fc2ReadGVCPMemory (fc2Context context, unsigned int address, unsigned char ∗pBuffer, unsigned int length)

  Read a GVCP memory block.

- FLYCAPTURE2_C_API fc2Error fc2GetGigEProperty (fc2Context context, fc2GigEProperty ∗pGigEProp)

  Get the specified GigEProperty.

- FLYCAPTURE2_C_API fc2Error fc2SetGigEProperty (fc2Context context, const fc2GigEProperty ∗pGigEProp)

  Set the specified GigEProperty.

- FLYCAPTURE2_C_API fc2Error fc2QueryGigEImagingMode (fc2Context context, fc2Mode mode, BOOL ∗isSupported)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEImagingMode (fc2Context context, fc2Mode ∗mode)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEImagingMode (fc2Context context, fc2Mode mode)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettingsInfo (fc2Context context, fc2GigEImageSettingsInfo ∗pInfo)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettings (fc2Context context, fc2GigEImageSettings ∗pImageSettings)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEImageSettings (fc2Context context, const fc2GigEImageSettings ∗pImageSettings)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEImageBinningSettings (fc2Context context, unsigned int ∗horzBinnningValue, unsigned int ∗vertBinnningValue)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEImageBinningSettings (fc2Context context, unsigned int horzBinnningValue, unsigned int vertBinnningValue)
- FLYCAPTURE2_C_API fc2Error fc2GetNumStreamChannels (fc2Context context, unsigned int ∗numChannels)
- FLYCAPTURE2_C_API fc2Error fc2GetGigEStreamChannelInfo (fc2Context context, unsigned int channel, fc2GigEStreamChannel ∗pChannel)
- FLYCAPTURE2_C_API fc2Error fc2SetGigEStreamChannelInfo (fc2Context context, unsigned int channel, fc2GigEStreamChannel ∗pChannel)
- FLYCAPTURE2_C_API fc2Error fc2GetLUTInfo (fc2Context context, fc2LUTData ∗pData)

  Query if LUT support is available on the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetLUTBankInfo (fc2Context context, unsigned int bank, BOOL ∗pReadSupported, BOOL ∗pWriteSupported)

  Query the read/write status of a single LUT bank.

- FLYCAPTURE2_C_API fc2Error fc2GetActiveLUTBank (fc2Context context, unsigned int ∗pActiveBank)

  Get the LUT bank that is currently being used.

- FLYCAPTURE2_C_API fc2Error fc2SetActiveLUTBank (fc2Context context, unsigned int activeBank)

  Set the LUT bank that will be used.

- FLYCAPTURE2_C_API fc2Error fc2EnableLUT (fc2Context context, BOOL on)

  Enable or disable LUT functionality on the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetLUTChannel (fc2Context context, unsigned int bank, unsigned int channel, unsigned int sizeEntries, unsigned int ∗pEntries)

  Get the LUT channel settings from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SetLUTChannel (fc2Context context, unsigned int bank, unsigned int channel, unsigned int sizeEntries, unsigned int ∗pEntries)

  Set the LUT channel settings to the camera.

- FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannel (fc2Context context, unsigned int ∗pCurrentChannel)

  Retrieve the current memory channel from the camera.

- FLYCAPTURE2_C_API fc2Error fc2SaveToMemoryChannel (fc2Context context, unsigned int channel)

  Save the current settings to the specfied current memory channel.

- FLYCAPTURE2_C_API fc2Error fc2RestoreFromMemoryChannel (fc2Context context, unsigned int channel)

  Restore the specfied current memory channel.

- FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannelInfo (fc2Context context, unsigned int ∗pNumChannels)

  Query the camera for memory channel support.

- FLYCAPTURE2_C_API fc2Error fc2GetEmbeddedImageInfo (fc2Context context, fc2EmbeddedImageInfo ∗pInfo)

  Get the current status of the embedded image information register, as well as the availability of each embedded property.

- FLYCAPTURE2_C_API fc2Error fc2SetEmbeddedImageInfo (fc2Context context, fc2EmbeddedImageInfo ∗pInfo)

  Sets the on/off values of the embedded image information structure to the camera.

- FLYCAPTURE2_C_API const char ∗ fc2GetRegisterString (unsigned int registerVal)

  Returns a text representation of the register value.

- FLYCAPTURE2_C_API fc2Error fc2CreateImage (fc2Image ∗pImage)

  Create a fc2Image.

- FLYCAPTURE2_C_API fc2Error fc2DestroyImage (fc2Image ∗image)

  Destroy the fc2Image.

- FLYCAPTURE2_C_API fc2Error fc2SetDefaultColorProcessing (fc2ColorProcessingAlgorithm defaultMethod)

  Set the default color processing algorithm.

- FLYCAPTURE2_C_API fc2Error fc2GetDefaultColorProcessing (fc2ColorProcessingAlgorithm ∗pDefaultMethod)

  Get the default color processing algorithm.

- FLYCAPTURE2_C_API fc2Error fc2SetDefaultOutputFormat (fc2PixelFormat format)

  Set the default output pixel format.

- FLYCAPTURE2_C_API fc2Error fc2GetDefaultOutputFormat (fc2PixelFormat ∗pFormat)

  Get the default output pixel format.

- FLYCAPTURE2_C_API fc2Error fc2DetermineBitsPerPixel (fc2PixelFormat format, unsigned int ∗pBitsPerPixel)

  Calculate the bits per pixel for the specified pixel format.

- FLYCAPTURE2_C_API fc2Error fc2SaveImage (fc2Image ∗pImage, const char ∗pFilename, fc2ImageFileFormat format)

  Save the image to the specified file name with the file format specified.

- FLYCAPTURE2_C_API fc2Error fc2SaveImageWithOption (fc2Image ∗pImage, const char ∗pFilename, fc2ImageFileFormat format, void ∗pOption)

  Save the image to the specified file name with the file format specified.

- FLYCAPTURE2_C_API fc2Error fc2ConvertImage (fc2Image ∗pImageIn, fc2Image ∗pImageOut)
- FLYCAPTURE2_C_API fc2Error fc2ConvertImageTo (fc2PixelFormat format, fc2Image ∗pImageIn, fc2Image ∗pImageOut)

  Converts the current image buffer to the specified output format and stores the result in the specified image.

- FLYCAPTURE2_C_API fc2Error fc2GetImageData (fc2Image ∗pImage, unsigned char ∗∗ppData)

  Get a pointer to the data associated with the image.

- FLYCAPTURE2_C_API fc2Error fc2SetImageData (fc2Image ∗pImage, const unsigned char ∗pData, unsigned int dataSize)

  Set the data of the Image object.

- FLYCAPTURE2_C_API fc2Error fc2SetImageDimensions (fc2Image ∗pImage, unsigned int rows, unsigned int cols, unsigned int stride, fc2PixelFormat pixelFormat, fc2BayerTileFormat bayerFormat)

  Sets the dimensions of the image object.

- FLYCAPTURE2_C_API fc2TimeStamp fc2GetImageTimeStamp (fc2Image ∗pImage)

  Get the timestamp data associated with the image.

- FLYCAPTURE2_C_API fc2Error fc2CalculateImageStatistics (fc2Image ∗pImage, fc2ImageStatisticsContext ∗pImageStatisticsContext)

  Calculate statistics associated with the image.

- FLYCAPTURE2_C_API fc2Error fc2CreateImageStatistics (fc2ImageStatisticsContext ∗pImageStatisticsContext)

  Create a statistics context.

- FLYCAPTURE2_C_API fc2Error fc2DestroyImageStatistics (fc2ImageStatisticsContext imageStatisticsContext)

  Destroy a statistics context.

- FLYCAPTURE2_C_API fc2Error fc2GetImageStatistics (fc2ImageStatisticsContext imageStatisticsContext, fc2StatisticsChannel channel, unsigned int ∗pRangeMin, unsigned int ∗pRangeMax, unsigned int ∗pPixelValueMin, unsigned int ∗pPixelValueMax, unsigned int ∗pNumPixelValues, float ∗pPixelValueMean, int ∗∗ppHistogram)

  Get all statistics for the image.

- FLYCAPTURE2_C_API fc2Error fc2CreateAVI (fc2AVIContext ∗pAVIContext)

  Create a AVI context.

- FLYCAPTURE2_C_API fc2Error fc2AVIOpen (fc2AVIContext AVIContext, const char ∗pFileName, fc2AVIOption ∗pOption)

  Open an AVI file in preparation for writing Images to disk.

- FLYCAPTURE2_C_API fc2Error fc2AVIAppend (fc2AVIContext AVIContext, fc2Image ∗pImage)

  Append an image to the AVI file.

- FLYCAPTURE2_C_API fc2Error fc2AVIClose (fc2AVIContext AVIContext)

  Close the AVI file.

- FLYCAPTURE2_C_API fc2Error fc2DestroyAVI (fc2AVIContext AVIContext)

  Destroy a AVI context.

- FLYCAPTURE2_C_API fc2Error fc2GetSystemInfo (fc2SystemInfo ∗pSystemInfo)

  Get system information.

- FLYCAPTURE2_C_API fc2Error fc2GetLibraryVersion (fc2Version ∗pVersion)

  Get library version.

- FLYCAPTURE2_C_API fc2Error fc2LaunchBrowser (const char ∗pAddress)

  Launch a URL in the system default browser.

- FLYCAPTURE2_C_API fc2Error fc2LaunchHelp (const char ∗pFileName)

  Open a CHM file in the system default CHM viewer.

- FLYCAPTURE2_C_API fc2Error fc2LaunchCommand (const char ∗pCommand)

  Execute a command in the terminal.

- FLYCAPTURE2_C_API fc2Error fc2LaunchCommandAsync (const char ∗pCommand, fc2AsyncCommandCallback pCallback, void ∗pUserData)

  Execute a command in the terminal.

- FLYCAPTURE2_C_API const char ∗ fc2ErrorToDescription (fc2Error error)

  Get a string representation of an error.

4.1.1 Function Documentation

4.1.1.1 FLYCAPTURE2_C_API fc2Error fc2AVIAppend (fc2AVIContext *AVIContext*, fc2Image ∗ *pImage*)

Append an image to the AVI file.

Parameters:

AVIContext The AVI context to use.

pImage The image to append.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.2 FLYCAPTURE2_C_API fc2Error fc2AVIClose (fc2AVIContext *AVIContext*)

Close the AVI file.

Parameters:

AVIContext The AVI context to use.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.3 FLYCAPTURE2_C_API fc2Error fc2AVIOpen (fc2AVIContext *AVIContext*, const char ∗ *pFileName*, fc2AVIOption ∗ *pOption*)

Open an AVI file in preparation for writing Images to disk.

The size of AVI files is limited to 2GB. The filenames are automatically generated using the filename specified.

Parameters:

AVIContext The AVI context to use.

pFileName The filename of the AVI file.

pOption Options to apply to the AVI file.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.4 FLYCAPTURE2_C_API fc2Error fc2CalculateImageStatistics (fc2Image ∗ *pImage*, fc2ImageStatisticsContext ∗ *pImageStatisticsContext*)

Calculate statistics associated with the image.

In order to collect statistics for a particular channel, the enabled flag for the channel must be set to true. Statistics can only be collected for images in Mono8, Mono16, RGB, RGBU, BGR and BGRU.

Parameters:

pImage The fc2Image to be used.

pImageStatisticsContext The fc2ImageStatisticsContext to hold the statistics.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.5 FLYCAPTURE2_C_API fc2Error fc2Connect (fc2Context *context*, fc2PGRGuid ∗ *guid*)

Connects the camera object to the camera specified by the GUID.

Parameters:

context The fc2Context to be used.

guid The unique identifier for a specific camera on the PC.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.6 FLYCAPTURE2_C_API fc2Error fc2ConvertImage (fc2Image ∗ *pImageIn*, fc2Image ∗ *pImageOut*)

Parameters:

pImageIn

pImageOut

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.7 FLYCAPTURE2_C_API fc2Error fc2ConvertImageTo (fc2PixelFormat *format*, fc2Image ∗ *pImageIn*, fc2Image ∗ *pImageOut*)

Converts the current image buffer to the specified output format and stores the result in the specified image. The destination image does not need to be configured in any way before the call is made.

Parameters:

format Output format of the converted image.

pImageIn Input image.

pImageOut Output image.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.8 FLYCAPTURE2_C_API fc2Error fc2CreateAVI (fc2AVIContext ∗ *pAVIContext*)

Create a AVI context.

Parameters:

pAVIContext A AVI context.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.9 FLYCAPTURE2_C_API fc2Error fc2CreateContext (fc2Context $*$ *pContext*)

Create a FC2 context for IIDC camaera.

This call must be made before any other calls that use a context will succeed.

Parameters:

pContext A pointer to the fc2Context to be created.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.10 FLYCAPTURE2_C_API fc2Error fc2CreateGigEContext (fc2Context $*$ *pContext*)

Create a FC2 context for a GigE Vision camera.

This call must be made before any other calls that use a context will succeed.

Parameters:

pContext A pointer to the fc2Context to be created.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.11 FLYCAPTURE2_C_API fc2Error fc2CreateImage (fc2Image $*$ *pImage*)

Create a fc2Image.

If externally allocated memory is to be used for the converted image, simply assigning the pData member of the fc2Image structure is insufficient. fc2SetImageData() should be called in order to populate the fc2Image structure correctly.

Parameters:

pImage Pointer to image to be created.

See also:

fc2SetImageData()

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.12 FLYCAPTURE2_C_API fc2Error fc2CreateImageStatistics (fc2ImageStatisticsContext $*$ *pImageStatisticsContext*)

Create a statistics context.

Parameters:

pImageStatisticsContext A statistics context.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.13 FLYCAPTURE2_C_API fc2Error fc2DestroyAVI (fc2AVIContext *AVIContext*)

Destroy a AVI context.

Parameters:

AVIContext A AVI context.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.14 FLYCAPTURE2_C_API fc2Error fc2DestroyContext (fc2Context *context*)

Destroy the FC2 context.

This must be called when the user is finished with the context in order to prevent memory leaks.

Parameters:

context The context to be destroyed.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.15 FLYCAPTURE2_C_API fc2Error fc2DestroyImage (fc2Image $*$ *image*)

Destroy the fc2Image.

Parameters:

image Pointer to image to be destroyed.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.16 FLYCAPTURE2_C_API fc2Error fc2DestroyImageStatistics (fc2ImageStatisticsContext *imageStatisticsContext*)

Destroy a statistics context.

Parameters:

imageStatisticsContext A statistics context.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.17 FLYCAPTURE2_C_API fc2Error fc2DetermineBitsPerPixel (fc2PixelFormat *format*, unsigned int ∗ *pBitsPerPixel*)

Calculate the bits per pixel for the specified pixel format.

Parameters:

format The pixel format.

pBitsPerPixel The bits per pixel.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.18 FLYCAPTURE2_C_API fc2Error fc2Disconnect (fc2Context *context*)

Disconnects the fc2Context from the camera.

Parameters:

context The fc2Context to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.19 FLYCAPTURE2_C_API fc2Error fc2DiscoverGigECameras (fc2Context *context*, fc2CameraInfo ∗ *gigECameras*, unsigned int ∗ *arraySize*)

Discover all cameras connected to the network even if they reside on a different subnet.

This is useful in situations where a GigE camera is using Persistent IP and the application's subnet is different from the device subnet. After discovering the camera, it is easy to use ForceIPAddressToCamera() to set a different IP configuration.

Parameters:

context The fc2Context to be used.

gigECameras Pointer to an array of CameraInfo structures.

arraySize Size of the array. Number of discovered cameras is returned in the same value.

Returns:

An Error indicating the success or failure of the function. If the error is PGRERROR_BUFFER_-TOO_SMALL then arraySize will contain the minimum size needed for gigECameras array.

4.1.1.20 FLYCAPTURE2_C_API fc2Error fc2EnableLUT (fc2Context *context*, BOOL *on*)

Enable or disable LUT functionality on the camera.

Parameters:

context The fc2Context to be used.

on Whether to enable or disable LUT.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.21 FLYCAPTURE2_C_API const char* fc2ErrorToDescription (fc2Error *error*)

Get a string representation of an error.

Parameters:

error Error to be parsed.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.22 FLYCAPTURE2_C_API fc2Error fc2FireBusReset (fc2Context *context*, fc2PGRGuid * *pGuid*)

Fire a bus reset.

The actual bus reset is only fired for the specified 1394 bus, but it will effectively cause a global bus reset for the library.

Parameters:

context The fc2Context to be used.

pGuid PGRGuid of the camera or the device to cause bus reset.

Returns:

An Error indicating the success or failure of the function.

4.1.1.23 FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTrigger (fc2Context *context*)

Parameters:

context The fc2Context to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.24 FLYCAPTURE2_C_API fc2Error fc2FireSoftwareTriggerBroadcast (fc2Context *context*)

Fire the software trigger according to the DCAM specifications.

Parameters:

context The fc2Context to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.25 FLYCAPTURE2_C_API fc2Error fc2ForceIPAddressToCamera (fc2Context *context*, fc2MACAddress *macAddress*, fc2IPAddress *ipAddress*, fc2IPAddress *subnetMask*, fc2IPAddress *defaultGateway*)

Force the camera with the specific MAC address to the specified IP address, subnet mask and default gateway.

This is useful in situations where a GigE Vision camera is using Persistent IP and the application's subnet is different from the device subnet.

Parameters:

context The fc2Context to be used.

macAddress MAC address of the camera.

ipAddress IP address to set on the camera.

subnetMask Subnet mask to set on the camera.

defaultGateway Default gateway to set on the camera.

Returns:

An Error indicating the success or failure of the function.

4.1.1.26 FLYCAPTURE2_C_API fc2Error fc2GetActiveLUTBank (fc2Context *context*, unsigned int ∗ *pActiveBank*)

Get the LUT bank that is currently being used.

For cameras with PGR LUT, the active bank is always 0.

Parameters:

context The fc2Context to be used.

pActiveBank The currently active bank.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.27 FLYCAPTURE2_C_API fc2Error fc2GetCameraFromIndex (fc2Context *context*, unsigned int *index*, fc2PGRGuid ∗ *pGuid*)

Gets the PGRGuid for a camera on the PC.

It uniquely identifies the camera specified by the index and is used to identify the camera during a fc2Connect() call.

Parameters:

context The fc2Context to be used.

index Zero based index of camera.

pGuid Unique PGRGuid for the camera.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.28 FLYCAPTURE2_C_API fc2Error fc2GetCameraFromSerialNumber (fc2Context *context*, unsigned int *serialNumber*, fc2PGRGuid ∗ *pGuid*)

Gets the PGRGuid for a camera on the PC.

It uniquely identifies the camera specified by the serial number and is used to identify the camera during a fc2Connect() call.

Parameters:

context The fc2Context to be used.

serialNumber Serial number of camera.

pGuid Unique PGRGuid for the camera.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.29 FLYCAPTURE2_C_API fc2Error fc2GetCameraInfo (fc2Context *context*, fc2CameraInfo ∗ *pCameraInfo*)

Retrieves information from the camera such as serial number, model name and other camera information.

Parameters:

context The fc2Context to be used.

pCameraInfo Pointer to the camera information structure to be filled.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.30 FLYCAPTURE2_C_API fc2Error fc2GetCameraSerialNumberFromIndex (fc2Context *context*, unsigned int *index*, unsigned int ∗ *pSerialNumber*)

Gets the serial number of the camera with the specified index.

Parameters:

context The fc2Context to be used.
index Zero based index of desired camera.
pSerialNumber Serial number of camera.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.31 FLYCAPTURE2_C_API fc2Error fc2GetConfiguration (fc2Context *context*, fc2Config ∗ *config*)

Get the configuration associated with the camera.

Parameters:

context The fc2Context to be used.
config Pointer to the configuration structure to be filled.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.32 FLYCAPTURE2_C_API fc2Error fc2GetDefaultColorProcessing (fc2ColorProcessingAlgorithm ∗ *pDefaultMethod*)

Get the default color processing algorithm.

Parameters:

pDefaultMethod The default color processing algorithm.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.33 FLYCAPTURE2_C_API fc2Error fc2GetDefaultOutputFormat (fc2PixelFormat ∗ *pFormat*)

Get the default output pixel format.

Parameters:

pFormat The default pixel format.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.34 FLYCAPTURE2_C_API fc2Error fc2GetDeviceFromIndex (fc2Context *context*, unsigned int *index*, fc2PGRGuid ∗ *pGuid*)

Gets the PGRGuid for a device.

It uniquely identifies the device specified by the index.

Parameters:

context The fc2Context to be used.

index Zero based index of device.

pGuid Unique PGRGuid for the device.

See also:

fc2GetNumOfDevices()

Returns:

An Error indicating the success or failure of the function.

4.1.1.35 FLYCAPTURE2_C_API fc2Error fc2GetEmbeddedImageInfo (fc2Context *context*, fc2EmbeddedImageInfo ∗ *pInfo*)

Get the current status of the embedded image information register, as well as the availability of each embedded property.

Parameters:

context The fc2Context to be used.

pInfo Structure to be filled.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.36 FLYCAPTURE2_C_API fc2Error fc2GetFormat7Configuration (fc2Context *context*, fc2Format7ImageSettings ∗ *imageSettings*, unsigned int ∗ *packetSize*, float ∗ *percentage*)

Get the current Format7 configuration from the camera.

This call will only succeed if the camera is already in Format7.

Parameters:

context The fc2Context to be used.

imageSettings Current image settings.

packetSize Current packet size.

percentage Current packet size as a percentage.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.37 FLYCAPTURE2_C_API fc2Error fc2GetFormat7Info (fc2Context *context*, fc2Format7Info ∗ *info*, BOOL ∗ *pSupported*)

Retrieve the availability of Format7 custom image mode and the camera capabilities for the specified Format7 mode.

The mode must be specified in the Format7Info structure in order for the function to succeed.

Parameters:

context The fc2Context to be used.

info Structure to be filled with the capabilities of the specified mode and the current state in the specified mode.

pSupported Whether the specified mode is supported.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.38 FLYCAPTURE2_C_API fc2Error fc2GetGigEImageBinningSettings (fc2Context *context*, unsigned int ∗ *horzBinnningValue*, unsigned int ∗ *vertBinnningValue*)

4.1.1.39 FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettings (fc2Context *context*, fc2GigEImageSettings ∗ *pImageSettings*)

4.1.1.40 FLYCAPTURE2_C_API fc2Error fc2GetGigEImageSettingsInfo (fc2Context *context*, fc2GigEImageSettingsInfo ∗ *pInfo*)

4.1.1.41 FLYCAPTURE2_C_API fc2Error fc2GetGigEImagingMode (fc2Context *context*, fc2Mode ∗ *mode*)

4.1.1.42 FLYCAPTURE2_C_API fc2Error fc2GetGigEProperty (fc2Context *context*, fc2GigEProperty ∗ *pGigEProp*)

Get the specified GigEProperty.

The GigEPropertyType field must be set in order for this function to succeed.

Parameters:

context The fc2Context to be used.

pGigEProp The GigE property to get.

Returns:

An Error indicating the success or failure of the function.

4.1.1.43 FLYCAPTURE2_C_API fc2Error fc2GetGigEStreamChannelInfo (fc2Context *context*, unsigned int *channel*, fc2GigEStreamChannel ∗ *pChannel*)

4.1.1.44 FLYCAPTURE2_C_API fc2Error fc2GetGPIOPinDirection (fc2Context *context*, unsigned int *pin*, unsigned int ∗ *pDirection*)

Get the GPIO pin direction for the specified pin.

This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

Parameters:

context The fc2Context to be used.

pin Pin to get the direction for.

pDirection Direction of the pin. 0 for input, 1 for output.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.45 FLYCAPTURE2_C_API fc2Error fc2GetImageData (fc2Image ∗ *pImage*, unsigned char ∗∗ *ppData*)

Get a pointer to the data associated with the image.

This function is considered unsafe. The pointer returned could be invalidated if the buffer is resized or released. The pointer may also be invalidated if the Image object is passed to fc2RetrieveBuffer().

Parameters:

pImage The fc2Image to be used.

ppData A pointer to the image data.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.46 FLYCAPTURE2_C_API fc2Error fc2GetImageStatistics (fc2ImageStatisticsContext *imageStatisticsContext*, fc2StatisticsChannel *channel*, unsigned int ∗ *pRangeMin*, unsigned int ∗ *pRangeMax*, unsigned int ∗ *pPixelValueMin*, unsigned int ∗ *pPixelValueMax*, unsigned int ∗ *pNumPixelValues*, float ∗ *pPixelValueMean*, int ∗∗ *ppHistogram*)

Get all statistics for the image.

Parameters:

imageStatisticsContext The statistics context.

channel The statistics channel.

pRangeMin The minimum possible value.

pRangeMax The maximum possible value.

pPixelValueMin The minimum pixel value.

pPixelValueMax The maximum pixel value.

pNumPixelValues The number of unique pixel values.

pPixelValueMean The mean of the image.

ppHistogram Pointer to an array containing the histogram.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.47 FLYCAPTURE2_C_API fc2TimeStamp fc2GetImageTimeStamp (fc2Image ∗ *pImage*)

Get the timestamp data associated with the image.

Parameters:

pImage The fc2Image to be used.

Returns:

Timestamp data associated with the image.

4.1.1.48 FLYCAPTURE2_C_API fc2Error fc2GetInterfaceTypeFromGuid (fc2Context *context*, fc2PGRGuid ∗ *pGuid*, fc2InterfaceType ∗ *pInterfaceType*)

Gets the interface type associated with a PGRGuid.

This is useful in situations where there is a need to enumerate all cameras for a particular interface.

Parameters:

context The fc2Context to be used.

pGuid The PGRGuid to get the interface for.

pInterfaceType The interface type of the PGRGuid.

Returns:

4.1.1.49 FLYCAPTURE2_C_API fc2Error fc2GetLibraryVersion (fc2Version ∗ *pVersion*)

Get library version.

Parameters:

pVersion Structure to receive the library version.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.50 FLYCAPTURE2_C_API fc2Error fc2GetLUTBankInfo (fc2Context *context*, unsigned int *bank*, BOOL ∗ *pReadSupported*, BOOL ∗ *pWriteSupported*)

Query the read/write status of a single LUT bank.

Parameters:

context The fc2Context to be used.

bank The bank to query.

pReadSupported Whether reading from the bank is supported.

pWriteSupported Whether writing to the bank is supported.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.51 FLYCAPTURE2_C_API fc2Error fc2GetLUTChannel (fc2Context *context*, unsigned int *bank*, unsigned int *channel*, unsigned int *sizeEntries*, unsigned int * *pEntries*)

Get the LUT channel settings from the camera.

Parameters:

context The fc2Context to be used.

bank Bank to retrieve.

channel Channel to retrieve.

sizeEntries Number of entries in LUT table to read.

pEntries Array to store LUT entries.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.52 FLYCAPTURE2_C_API fc2Error fc2GetLUTInfo (fc2Context *context*, fc2LUTData * *pData*)

Query if LUT support is available on the camera.

Parameters:

context The fc2Context to be used.

pData The LUT structure to be filled.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.53 FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannel (fc2Context *context*, unsigned int * *pCurrentChannel*)

Retrieve the current memory channel from the camera.

Parameters:

context The fc2Context to be used.

pCurrentChannel Current memory channel.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.54 FLYCAPTURE2_C_API fc2Error fc2GetMemoryChannelInfo (fc2Context *context*, unsigned int ∗ *pNumChannels*)

Query the camera for memory channel support.

If the number of channels is 0, then memory channel support is not available.

Parameters:

context The fc2Context to be used.

pNumChannels Number of memory channels supported.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.55 FLYCAPTURE2_C_API fc2Error fc2GetNumOfCameras (fc2Context *context*, unsigned int ∗ *pNumCameras*)

Gets the number of cameras attached to the PC.

Parameters:

context The fc2Context to be used.

pNumCameras Number of cameras detected.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.56 FLYCAPTURE2_C_API fc2Error fc2GetNumOfDevices (fc2Context *context*, unsigned int ∗ *pNumDevices*)

Gets the number of devices.

This may include hubs, host controllers and other hardware devices (including cameras).

Parameters:

context The fc2Context to be used.

pNumDevices The number of devices found.

Returns:

An Error indicating the success or failure of the function.

4.1.1.57 FLYCAPTURE2_C_API fc2Error fc2GetNumStreamChannels (fc2Context *context*, unsigned int ∗ *numChannels*)

4.1.1.58 FLYCAPTURE2_C_API fc2Error fc2GetProperty (fc2Context *context*, fc2Property ∗ *prop*)

Reads the settings for the specified property from the camera.

The property type must be specified in the fc2Property structure passed into the function in order for the function to succeed. If auto is on, the integer and abs values returned may not be consistent with each other.

Parameters:

context The fc2Context to be used.
prop Pointer to the Property structure to be filled.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.59 FLYCAPTURE2_C_API fc2Error fc2GetPropertyInfo (fc2Context *context*, fc2PropertyInfo ∗ *propInfo*)

Retrieves information about the specified camera property.

The property type must be specified in the fc2PropertyInfo structure passed into the function in order for the function to succeed.

Parameters:

context The fc2Context to be used.
propInfo Pointer to the PropertyInfo structure to be filled.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.60 FLYCAPTURE2_C_API const char∗ fc2GetRegisterString (unsigned int *registerVal*)

Returns a text representation of the register value.

Parameters:

registerVal The register value to query.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.61 FLYCAPTURE2_C_API fc2Error fc2GetStrobe (fc2Context *context*, fc2StrobeControl ∗ *strobeControl*)

Retrieve current strobe settings from the camera.

The strobe pin must be specified in the structure before being passed in to the function.

Parameters:

context The fc2Context to be used.
strobeControl Structure to receive strobe settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.62 FLYCAPTURE2_C_API fc2Error fc2GetStrobeInfo (fc2Context *context*, fc2StrobeInfo ∗ *strobeInfo*)

Retrieve strobe information from the camera.

Parameters:

context The fc2Context to be used.

strobeInfo Structure to receive strobe information.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.63 FLYCAPTURE2_C_API fc2Error fc2GetSystemInfo (fc2SystemInfo ∗ *pSystemInfo*)

Get system information.

Parameters:

pSystemInfo Structure to receive system information.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.64 FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelay (fc2Context *context*, fc2TriggerDelay ∗ *triggerDelay*)

Retrieve current trigger delay settings from the camera.

Parameters:

context The fc2Context to be used.

triggerDelay Structure to receive trigger delay settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.65 FLYCAPTURE2_C_API fc2Error fc2GetTriggerDelayInfo (fc2Context *context*, fc2TriggerDelayInfo ∗ *triggerDelayInfo*)

Retrieve trigger delay information from the camera.

Parameters:

context The fc2Context to be used.

triggerDelayInfo Structure to receive trigger delay information.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.66 FLYCAPTURE2_C_API fc2Error fc2GetTriggerMode (fc2Context *context*, fc2TriggerMode ∗ *triggerMode*)

Retrieve current trigger settings from the camera.

Parameters:

context The fc2Context to be used.

triggerMode Structure to receive trigger mode settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.67 FLYCAPTURE2_C_API fc2Error fc2GetTriggerModeInfo (fc2Context *context*, fc2TriggerModeInfo ∗ *triggerModeInfo*)

Retrieve trigger information from the camera.

Parameters:

context The fc2Context to be used.

triggerModeInfo Structure to receive trigger information.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.68 FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRate (fc2Context *context*, fc2VideoMode ∗ *videoMode*, fc2FrameRate ∗ *frameRate*)

Get the current video mode and frame rate from the camera.

If the camera is in Format7, the video mode will be VIDEOMODE_FORMAT7 and the frame rate will be FRAMERATE_FORMAT7.

Parameters:

context The fc2Context to be used.

videoMode Current video mode.

frameRate Current frame rate.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.69 FLYCAPTURE2_C_API fc2Error fc2GetVideoModeAndFrameRateInfo (fc2Context *context*, fc2VideoMode *videoMode*, fc2FrameRate *frameRate*, BOOL ∗ *pSupported*)

Query the camera to determine if the specified video mode and frame rate is supported.

Parameters:

context The fc2Context to be used.

videoMode Video mode to check.

frameRate Frame rate to check.

pSupported Whether the video mode and frame rate is supported.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.70 FLYCAPTURE2_C_API fc2Error fc2LaunchBrowser (const char ∗ *pAddress*)

Launch a URL in the system default browser.

Parameters:

pAddress URL to open in browser.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.71 FLYCAPTURE2_C_API fc2Error fc2LaunchCommand (const char ∗ *pCommand*)

Execute a command in the terminal.

This is a blocking call that will return when the command completes.

Parameters:

pCommand Command to execute.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.72 FLYCAPTURE2_C_API fc2Error fc2LaunchCommandAsync (const char ∗ *pCommand*, fc2AsyncCommandCallback *pCallback*, void ∗ *pUserData*)

Execute a command in the terminal.

This is a non-blocking call that will return immediately. The return value of the command can be retrieved in the callback.

Parameters:

pCommand Command to execute.

pCallback Callback to fire when command is complete.

pUserData Data pointer to pass to callback.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.73 FLYCAPTURE2_C_API fc2Error fc2LaunchHelp (const char ∗ *pFileName*)

Open a CHM file in the system default CHM viewer.

Parameters:

pFileName Filename of CHM file to open.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.74 FLYCAPTURE2_C_API fc2Error fc2QueryGigEImagingMode (fc2Context *context*, fc2Mode *mode*, BOOL ∗ *isSupported*)

4.1.1.75 FLYCAPTURE2_C_API fc2Error fc2ReadGVCPMemory (fc2Context *context*, unsigned int *address*, unsigned char ∗ *pBuffer*, unsigned int *length*)

Read a GVCP memory block.

Parameters:

context The fc2Context to be used.

address GVCP address to be read from.

pBuffer Array containing data to be written.

length Size of array, in quadlets.

Returns:

An Error indicating the success or failure of the function.

4.1.1.76 FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegister (fc2Context *context*, unsigned int *address*, unsigned int ∗ *pValue*)

Read a GVCP register.

Parameters:

context The fc2Context to be used.

address GVCP address to be read from.

pValue The value that is read.

Returns:

An Error indicating the success or failure of the function.

4.1.1.77 FLYCAPTURE2_C_API fc2Error fc2ReadGVCPRegisterBlock (fc2Context *context*, unsigned int *address*, unsigned int ∗ *pBuffer*, unsigned int *length*)

Read a GVCP register block.

Parameters:

context The fc2Context to be used.

address GVCP address to be read from.

pBuffer Array containing data to be written.

length Size of array, in quadlets.

Returns:

An Error indicating the success or failure of the function.

4.1.1.78 FLYCAPTURE2_C_API fc2Error fc2ReadRegister (fc2Context *context*, unsigned int *address*, unsigned int ∗ *pValue*)

Read the specified register from the camera.

Parameters:

context The fc2Context to be used.

address DCAM address to be read from.

pValue The value that is read.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.79 FLYCAPTURE2_C_API fc2Error fc2ReadRegisterBlock (fc2Context *context*, unsigned short *addressHigh*, unsigned int *addressLow*, unsigned int ∗ *pBuffer*, unsigned int *length*)

Write to the specified register block on the camera.

Parameters:

context The fc2Context to be used.

addressHigh Top 16 bits of the 48 bit absolute address to read from.

addressLow Bottom 32 bits of the 48 bits absolute address to read from.

pBuffer Array to store read data.

length Size of array, in quadlets.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.80 FLYCAPTURE2_C_API fc2Error fc2RegisterCallback (fc2Context *context*, fc2BusEventCallback *enumCallback*, fc2BusCallbackType *callbackType*, void ∗ *pParameter*, fc2CallbackHandle ∗ *pCallbackHandle*)

Register a callback function that will be called when the specified callback event occurs.

Parameters:

context The fc2Context to be used.
enumCallback Pointer to function that will receive the callback.
callbackType Type of callback to register for.
pParameter Callback parameter to be passed to callback.
pCallbackHandle Unique callback handle used for unregistering callback.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.81 FLYCAPTURE2_C_API fc2Error fc2RescanBus (fc2Context *context*)

Force a rescan of the buses.

This does not trigger a bus reset. However, any current connections to a Camera object will be invalidated.

Returns:

An Error indicating the success or failure of the function.

4.1.1.82 FLYCAPTURE2_C_API fc2Error fc2RestoreFromMemoryChannel (fc2Context *context*, unsigned int *channel*)

Restore the specfied current memory channel.

Parameters:

context The fc2Context to be used.
channel Memory channel to restore from.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.83 FLYCAPTURE2_C_API fc2Error fc2RetrieveBuffer (fc2Context *context*, fc2Image ∗ *pImage*)

Retrieves the the next image object containing the next image.

Parameters:

context The fc2Context to be used.
pImage Pointer to fc2Image to store image data.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.84 FLYCAPTURE2_C_API fc2Error fc2SaveImage (fc2Image ∗ *pImage*, const char ∗ *pFilename*, fc2ImageFileFormat *format*)

Save the image to the specified file name with the file format specified.

Parameters:

pImage The fc2Image to be used.

pFilename Filename to save image with.

format File format to save in.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.85 FLYCAPTURE2_C_API fc2Error fc2SaveImageWithOption (fc2Image ∗ *pImage*, const char ∗ *pFilename*, fc2ImageFileFormat *format*, void ∗ *pOption*)

Save the image to the specified file name with the file format specified.

Parameters:

pImage The fc2Image to be used.

pFilename Filename to save image with.

format File format to save in.

pOption Options for saving image.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.86 FLYCAPTURE2_C_API fc2Error fc2SaveToMemoryChannel (fc2Context *context*, unsigned int *channel*)

Save the current settings to the specfied current memory channel.

Parameters:

context The fc2Context to be used.

channel Memory channel to save to.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.87 FLYCAPTURE2_C_API fc2Error fc2SetActiveLUTBank (fc2Context *context*, unsigned int *activeBank*)

Set the LUT bank that will be used.

Parameters:

context The fc2Context to be used.

activeBank The bank to be set as active.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.88 FLYCAPTURE2_C_API fc2Error fc2SetCallback (fc2Context *context*, fc2ImageEventCallback *pCallbackFn*, void ∗ *pCallbackData*)

Sets the callback data to be used on completion of image transfer.

To clear the current stored callback data, pass in NULL for both callback arguments.

Parameters:

context The fc2Context to be used.

pCallbackFn A function to be called when a new image is received.

pCallbackData A pointer to data that can be passed to the callback function.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.89 FLYCAPTURE2_C_API fc2Error fc2SetConfiguration (fc2Context *context*, fc2Config ∗ *config*)

Set the configuration associated with the camera.

Parameters:

context The fc2Context to be used.

config Pointer to the configuration structure to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.90 FLYCAPTURE2_C_API fc2Error fc2SetDefaultColorProcessing (fc2ColorProcessingAlgorithm *defaultMethod*)

Set the default color processing algorithm.

This method will be used for any image with the DEFAULT algorithm set. The method used is determined at the time of the Convert() call, therefore the most recent execution of this function will take precedence. The default setting is shared within the current process.

Parameters:

defaultMethod The color processing algorithm to set.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.91 FLYCAPTURE2_C_API fc2Error fc2SetDefaultOutputFormat (fc2PixelFormat *format*)

Set the default output pixel format.

This format will be used for any call to Convert() that does not specify an output format. The format used will be determined at the time of the Convert() call, therefore the most recent execution of this function will take precedence. The default is shared within the current process.

Parameters:

format The output pixel format to set.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.92 FLYCAPTURE2_C_API fc2Error fc2SetEmbeddedImageInfo (fc2Context *context*, fc2EmbeddedImageInfo * *pInfo*)

Sets the on/off values of the embedded image information structure to the camera.

Parameters:

context The fc2Context to be used.

pInfo Structure to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.93 FLYCAPTURE2_C_API fc2Error fc2SetFormat7Configuration (fc2Context *context*, fc2Format7ImageSettings * *imageSettings*, float *percentSpeed*)

Set the current Format7 configuration to the camera.

Parameters:

context The fc2Context to be used.

imageSettings Image settings to be written to the camera.

percentSpeed Packet size as a percentage to be written to the camera.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.94 FLYCAPTURE2_C_API fc2Error fc2SetFormat7ConfigurationPacket (fc2Context *context*, fc2Format7ImageSettings * *imageSettings*, unsigned int *packetSize*)

Set the current Format7 configuration to the camera.

Parameters:

context The fc2Context to be used.

imageSettings Image settings to be written to the camera.

packetSize Packet size to be written to the camera.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.95 FLYCAPTURE2_C_API fc2Error fc2SetGigEImageBinningSettings (fc2Context *context*, unsigned int *horzBinnningValue*, unsigned int *vertBinnningValue*)

4.1.1.96 FLYCAPTURE2_C_API fc2Error fc2SetGigEImageSettings (fc2Context *context*, const fc2GigEImageSettings ∗ *pImageSettings*)

4.1.1.97 FLYCAPTURE2_C_API fc2Error fc2SetGigEImagingMode (fc2Context *context*, fc2Mode *mode*)

4.1.1.98 FLYCAPTURE2_C_API fc2Error fc2SetGigEProperty (fc2Context *context*, const fc2GigEProperty ∗ *pGigEProp*)

Set the specified GigEProperty.

The GigEPropertyType field must be set in order for this function to succeed.

Parameters:

context The fc2Context to be used.

pGigEProp The GigE property to set.

Returns:

An Error indicating the success or failure of the function.

4.1.1.99 FLYCAPTURE2_C_API fc2Error fc2SetGigEStreamChannelInfo (fc2Context *context*, unsigned int *channel*, fc2GigEStreamChannel ∗ *pChannel*)

4.1.1.100 FLYCAPTURE2_C_API fc2Error fc2SetGPIOPinDirection (fc2Context *context*, unsigned int *pin*, unsigned int *direction*)

Set the GPIO pin direction for the specified pin.

This is useful if there is a need to set the pin into an input pin (i.e. to read the voltage) off the pin without setting it as a trigger source. This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

Parameters:

context The fc2Context to be used.

pin Pin to get the direction for.

direction Direction of the pin. 0 for input, 1 for output.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.101 FLYCAPTURE2_C_API fc2Error fc2SetGPIOPinDirectionBroadcast (fc2Context *context*, unsigned int *pin*, unsigned int *direction*)

Set the GPIO pin direction for the specified pin.

This is useful if there is a need to set the pin into an input pin (i.e. to read the voltage) off the pin without setting it as a trigger source. This is not a required call when using the trigger or strobe functions as the pin direction is set automatically internally.

Parameters:

context The fc2Context to be used.

pin Pin to get the direction for.

direction Direction of the pin. 0 for input, 1 for output.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.102 FLYCAPTURE2_C_API fc2Error fc2SetImageData (fc2Image ∗ *pImage*, const unsigned char ∗ *pData*, unsigned int *dataSize*)

Set the data of the Image object.

Ownership of the image buffer is not transferred to the Image object. It is the user's responsibility to delete the buffer when it is no longer in use.

Parameters:

pImage The fc2Image to be used.

pData Pointer to the image buffer.

dataSize Size of the image buffer.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.103 FLYCAPTURE2_C_API fc2Error fc2SetImageDimensions (fc2Image ∗ *pImage*, unsigned int *rows*, unsigned int *cols*, unsigned int *stride*, fc2PixelFormat *pixelFormat*, fc2BayerTileFormat *bayerFormat*)

Sets the dimensions of the image object.

Parameters:

pImage The fc2Image to be used.

rows Number of rows to set.

cols Number of cols to set.

stride Stride to set.

pixelFormat Pixel format to set.

bayerFormat Bayer tile format to set.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.104 FLYCAPTURE2_C_API fc2Error fc2SetLUTChannel (fc2Context *context*, unsigned int *bank*, unsigned int *channel*, unsigned int *sizeEntries*, unsigned int ∗ *pEntries*)

Set the LUT channel settings to the camera.

Parameters:

context The fc2Context to be used.

bank Bank to set.

channel Channel to set.

sizeEntries Number of entries in LUT table to write.

pEntries Array containing LUT entries to write.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.105 FLYCAPTURE2_C_API fc2Error fc2SetProperty (fc2Context *context*, fc2Property ∗ *prop*)

Writes the settings for the specified property to the camera.

The property type must be specified in the Property structure passed into the function in order for the function to succeed. The absControl flag controls whether the absolute or integer value is written to the camera.

Parameters:

context The fc2Context to be used.

prop Pointer to the Property structure to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.106 FLYCAPTURE2_C_API fc2Error fc2SetPropertyBroadcast (fc2Context *context*, fc2Property ∗ *prop*)

Writes the settings for the specified property to the camera.

The property type must be specified in the Property structure passed into the function in order for the function to succeed. The absControl flag controls whether the absolute or integer value is written to the camera.

Parameters:

context The fc2Context to be used.

prop Pointer to the Property structure to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.107 FLYCAPTURE2_C_API fc2Error fc2SetStrobe (fc2Context *context*, fc2StrobeControl ∗ *strobeControl*)

Set current strobe settings to the camera.

The strobe pin must be specified in the structure before being passed in to the function.

Parameters:

context The fc2Context to be used.

strobeControl Structure providing strobe settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.108 FLYCAPTURE2_C_API fc2Error fc2SetStrobeBroadcast (fc2Context *context*, fc2StrobeControl ∗ *strobeControl*)

Set current strobe settings to the camera.

The strobe pin must be specified in the structure before being passed in to the function.

Parameters:

context The fc2Context to be used.

strobeControl Structure providing strobe settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.109 FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelay (fc2Context *context*, fc2TriggerDelay ∗ *triggerDelay*)

Set the specified trigger delay settings to the camera.

Parameters:

context The fc2Context to be used.

triggerDelay Structure providing trigger delay settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.110 FLYCAPTURE2_C_API fc2Error fc2SetTriggerDelayBroadcast (fc2Context *context*, fc2TriggerDelay ∗ *triggerDelay*)

Set the specified trigger delay settings to the camera.

Parameters:

context The fc2Context to be used.

triggerDelay Structure providing trigger delay settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.111 FLYCAPTURE2_C_API fc2Error fc2SetTriggerMode (fc2Context *context*, fc2TriggerMode ∗ *triggerMode*)

Set the specified trigger settings to the camera.

Parameters:

context The fc2Context to be used.

triggerMode Structure providing trigger mode settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.112 FLYCAPTURE2_C_API fc2Error fc2SetTriggerModeBroadcast (fc2Context *context*, fc2TriggerMode ∗ *triggerMode*)

Set the specified trigger settings to the camera.

Parameters:

context The fc2Context to be used.

triggerMode Structure providing trigger mode settings.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.113 FLYCAPTURE2_C_API fc2Error fc2SetUserBuffers (fc2Context *context*, unsigned char ∗const *ppMemBuffers*, int *size*, int *nNumBuffers*)

Specify user allocated buffers to use as image data buffers.

Parameters:

context The fc2Context to be used.

ppMemBuffers Pointer to memory buffers to be written to. The size of the data being should be be equal to (size ∗ numBuffers) or larger.

size The size of each buffer (in bytes).

nNumBuffers Number of buffers in the array.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.114 FLYCAPTURE2_C_API fc2Error fc2SetVideoModeAndFrameRate (fc2Context *context*, fc2VideoMode *videoMode*, fc2FrameRate *frameRate*)

Set the specified video mode and frame rate to the camera.

It is not possible to set the camera to VIDEOMODE_FORMAT7 or FRAMERATE_FORMAT7. Use the Format7 functions to set the camera into Format7.

Parameters:

context The fc2Context to be used.

videoMode Video mode to set to camera.

frameRate Frame rate to set to camera.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.115 FLYCAPTURE2_C_API fc2Error fc2StartCapture (fc2Context *context*)

Starts isochronous image capture.

It will use either the current video mode or the most recently set video mode of the camera.

Parameters:

context The fc2Context to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.116 FLYCAPTURE2_C_API fc2Error fc2StartCaptureCallback (fc2Context *context*, fc2ImageEventCallback *pCallbackFn*, void ∗ *pCallbackData*)

Starts isochronous image capture.

It will use either the current video mode or the most recently set video mode of the camera. The callback function is called when a new image is received from the camera.

Parameters:

context The fc2Context to be used.

pCallbackFn A function to be called when a new image is received.

pCallbackData A pointer to data that can be passed to the callback function. A NULL pointer is acceptable.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.117 FLYCAPTURE2_C_API fc2Error fc2StartSyncCapture (unsigned int *numCameras*, fc2Context ∗ *pContexts*)

Starts synchronized isochronous image capture on multiple cameras.

Parameters:

numCameras Number of fc2Contexts in the ppCameras array.

pContexts Array of fc2Contexts.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.118 FLYCAPTURE2_C_API fc2Error fc2StartSyncCaptureCallback (unsigned int *numCameras*, fc2Context ∗ *pContexts*, fc2ImageEventCallback ∗ *pCallbackFns*, void ∗∗ *pCallbackDataArray*)

Starts synchronized isochronous image capture on multiple cameras.

Parameters:

numCameras Number of fc2Contexts in the ppCameras array.

pContexts Array of fc2Contexts.

pCallbackFns Array of callback functions for each camera.

pCallbackDataArray Array of callback data pointers.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.119 FLYCAPTURE2_C_API fc2Error fc2StopCapture (fc2Context *context*)

Stops isochronous image transfer and cleans up all associated resources.

Parameters:

context The fc2Context to be used.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.120 FLYCAPTURE2_C_API fc2Error fc2UnregisterCallback (fc2Context *context*, fc2CallbackHandle *callbackHandle*)

Unregister a callback function.

Parameters:

context The fc2Context to be used.

callbackHandle Unique callback handle.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.121 FLYCAPTURE2_C_API fc2Error fc2ValidateFormat7Settings (fc2Context *context*, fc2Format7ImageSettings ∗ *imageSettings*, BOOL ∗ *settingsAreValid*, fc2Format7PacketInfo ∗ *packetInfo*)

Validates Format7ImageSettings structure and returns valid packet size information if the image settings are valid.

The current image settings are cached while validation is taking place. The cached settings are restored when validation is complete.

Parameters:

context The fc2Context to be used.

imageSettings Structure containing the image settings.

settingsAreValid Whether the settings are valid.

packetInfo Packet size information that can be used to determine a valid packet size.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.122 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPMemory (fc2Context *context*, unsigned int *address*, const unsigned char ∗ *pBuffer*, unsigned int *length*)

Write a GVCP memory block.

Parameters:

context The fc2Context to be used.

address GVCP address to be write to.

pBuffer Array containing data to be written.

length Size of array, in quadlets.

Returns:

An Error indicating the success or failure of the function.

4.1.1.123 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegister (fc2Context *context*, unsigned int *address*, unsigned int *value*)

Write a GVCP register.

Parameters:

context The fc2Context to be used.

address GVCP address to be written to.

value The value to be written.

Returns:

An Error indicating the success or failure of the function.

4.1.1.124 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBlock (fc2Context *context*, unsigned int *address*, const unsigned int ∗ *pBuffer*, unsigned int *length*)

Write a GVCP register block.

Parameters:

context The fc2Context to be used.

address GVCP address to be write to.

pBuffer Array containing data to be written.

length Size of array, in quadlets.

Returns:

An Error indicating the success or failure of the function.

4.1.1.125 FLYCAPTURE2_C_API fc2Error fc2WriteGVCPRegisterBroadcast (fc2Context *context*, unsigned int *address*, unsigned int *value*)

Write a GVCP register with broadcast.

Parameters:

context The fc2Context to be used.

address GVCP address to be written to.

value The value to be written.

Returns:

An Error indicating the success or failure of the function.

4.1.1.126 FLYCAPTURE2_C_API fc2Error fc2WriteRegister (fc2Context *context*, unsigned int *address*, unsigned int *value*)

Write to the specified register on the camera.

Parameters:

context The fc2Context to be used.

address DCAM address to be written to.

value The value to be written.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.127 FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBlock (fc2Context *context*, unsigned short *addressHigh*, unsigned int *addressLow*, const unsigned int ∗ *pBuffer*, unsigned int *length*)

Write to the specified register block on the camera.

Parameters:

context The fc2Context to be used.

addressHigh Top 16 bits of the 48 bit absolute address to write to.

addressLow Bottom 32 bits of the 48 bits absolute address to write to.

pBuffer Array containing data to be written.

length Size of array, in quadlets.

Returns:

A fc2Error indicating the success or failure of the function.

4.1.1.128 FLYCAPTURE2_C_API fc2Error fc2WriteRegisterBroadcast (fc2Context *context*, unsigned int *address*, unsigned int *value*)

Write to the specified register on the camera with broadcast.

Parameters:

context The fc2Context to be used.

address DCAM address to be written to.

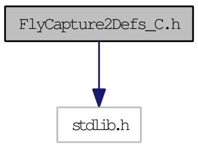
value The value to be written.

Returns:

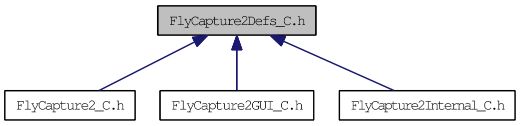
A fc2Error indicating the success or failure of the function.

4.2 FlyCapture2Defs_C.h File Reference

Include dependency graph for FlyCapture2Defs_C.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct fc2PGRGuid

  A GUID to the camera.

- struct fc2Image
- struct fc2SystemInfo
- struct fc2Version
- struct fc2Config
- struct fc2TriggerDelayInfo
- struct fc2TriggerDelay
- struct fc2TriggerModeInfo
- struct fc2TriggerMode
- struct fc2StrobeInfo
- struct fc2StrobeControl
- struct fc2Format7ImageSettings
- struct fc2Format7Info
- struct fc2Format7PacketInfo
- struct fc2IPAddress
- struct fc2MACAddress
- struct fc2GigEProperty
- struct fc2GigEStreamChannel
- struct fc2GigEConfig
- struct fc2GigEImageSettingsInfo
- struct fc2GigEImageSettings
- struct fc2TimeStamp
- struct fc2ConfigROM
- struct fc2CameraInfo
- struct fc2EmbeddedImageInfoProperty
- struct fc2EmbeddedImageInfo
- struct fc2ImageMetadata

- struct fc2LUTData
- struct fc2PNGOption
- struct fc2PPMOption
- struct fc2PGMOption
- struct fc2TIFFOption
- struct fc2JPEGOption
- struct fc2JPG2Option
- struct fc2AVIOption

Defines

- #define FALSE 0
- #define TRUE 1
- #define FULL_32BIT_VALUE 0x7FFFFFFF
- #define MAX_STRING_LENGTH 512

Typedefs

- typedef int BOOL
- typedef void ∗ fc2Context

  A context to the FlyCapture2 C library.

- typedef void ∗ fc2GuiContext

  A context to the FlyCapture2 C GUI library.

- typedef void ∗ fc2ImageImpl

  An internal pointer used in the fc2Image structure.

- typedef void ∗ fc2AVIContext

  A context referring to the AVI recorder object.

- typedef void ∗ fc2ImageStatisticsContext

  A context referring to the ImageStatistics object.

- typedef void ∗ fc2CallbackHandle
- typedef void(∗ fc2BusEventCallback)(void ∗pParameter, unsigned int serialNumber)
- typedef void(∗ fc2ImageEventCallback)(fc2Image ∗image, void ∗pCallbackData)
- typedef void(∗ fc2AsyncCommandCallback)(fc2Error retError, void ∗pUserData)

Enumerations

- enum fc2Error {

  FC2_ERROR_UNDEFINED = -1,

  FC2_ERROR_OK,

  FC2_ERROR_FAILED,

  FC2_ERROR_NOT_IMPLEMENTED,

  FC2_ERROR_FAILED_BUS_MASTER_CONNECTION,

FC2_ERROR_NOT_CONNECTED,
FC2_ERROR_INIT_FAILED,
FC2_ERROR_NOT_INTITIALIZED,
FC2_ERROR_INVALID_PARAMETER,
FC2_ERROR_INVALID_SETTINGS,
FC2_ERROR_INVALID_BUS_MANAGER,
FC2_ERROR_MEMORY_ALLOCATION_FAILED,
FC2_ERROR_LOW_LEVEL_FAILURE,
FC2_ERROR_NOT_FOUND,
FC2_ERROR_FAILED_GUID,
FC2_ERROR_INVALID_PACKET_SIZE,
FC2_ERROR_INVALID_MODE,
FC2_ERROR_NOT_IN_FORMAT7,
FC2_ERROR_NOT_SUPPORTED,
FC2_ERROR_TIMEOUT,
FC2_ERROR_BUS_MASTER_FAILED,
FC2_ERROR_INVALID_GENERATION,
FC2_ERROR_LUT_FAILED,
FC2_ERROR_IIDC_FAILED,
FC2_ERROR_STROBE_FAILED,
FC2_ERROR_TRIGGER_FAILED,
FC2_ERROR_PROPERTY_FAILED,
FC2_ERROR_PROPERTY_NOT_PRESENT,
FC2_ERROR_REGISTER_FAILED,
FC2_ERROR_READ_REGISTER_FAILED,
FC2_ERROR_WRITE_REGISTER_FAILED,
FC2_ERROR_ISOCH_FAILED,
FC2_ERROR_ISOCH_ALREADY_STARTED,
FC2_ERROR_ISOCH_NOT_STARTED,
FC2_ERROR_ISOCH_START_FAILED,
FC2_ERROR_ISOCH_RETRIEVE_BUFFER_FAILED,
FC2_ERROR_ISOCH_STOP_FAILED,
FC2_ERROR_ISOCH_SYNC_FAILED,
FC2_ERROR_ISOCH_BANDWIDTH_EXCEEDED,
FC2_ERROR_IMAGE_CONVERSION_FAILED,
FC2_ERROR_IMAGE_LIBRARY_FAILURE,
FC2_ERROR_BUFFER_TOO_SMALL,
FC2_ERROR_IMAGE_CONSISTENCY_ERROR,
FC2_ERROR_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2BusCallbackType {
  FC2_BUS_RESET,
  FC2_ARRIVAL,
  FC2_REMOVAL,
  FC2_CALLBACK_TYPE_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2GrabMode {
  FC2_DROP_FRAMES,
  FC2_BUFFER_FRAMES,
  FC2_UNSPECIFIED_GRAB_MODE,
  FC2_GRAB_MODE_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2GrabTimeout {
  FC2_TIMEOUT_NONE = 0,
  FC2_TIMEOUT_INFINITE = -1,
  FC2_TIMEOUT_UNSPECIFIED = -2,
  FC2_GRAB_TIMEOUT_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2BandwidthAllocation {
  FC2_BANDWIDTH_ALLOCATION_OFF = 0,
  FC2_BANDWIDTH_ALLOCATION_ON = 1,
  FC2_BANDWIDTH_ALLOCATION_UNSUPPORTED = 2,
  FC2_BANDWIDTH_ALLOCATION_UNSPECIFIED = 3,
  FC2_BANDWIDTH_ALLOCATION_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2InterfaceType {
  FC2_INTERFACE_IEEE1394,
  FC2_INTERFACE_USB_2,
  FC2_INTERFACE_GIGE,
  FC2_INTERFACE_UNKNOWN,
  FC2_INTERFACE_TYPE_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2PropertyType {
  FC2_BRIGHTNESS,
  FC2_AUTO_EXPOSURE,
  FC2_SHARPNESS,
  FC2_WHITE_BALANCE,
  FC2_HUE,
  FC2_SATURATION,
  FC2_GAMMA,
  FC2_IRIS,
  FC2_FOCUS,
  FC2_ZOOM,
  FC2_PAN,
  FC2_TILT,
  FC2_SHUTTER,
  FC2_GAIN,

FC2_TRIGGER_MODE,
FC2_TRIGGER_DELAY,
FC2_FRAME_RATE,
FC2_TEMPERATURE,
FC2_UNSPECIFIED_PROPERTY_TYPE,
FC2_PROPERTY_TYPE_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2FrameRate {
FC2_FRAMERATE_1_875,
FC2_FRAMERATE_3_75,
FC2_FRAMERATE_7_5,
FC2_FRAMERATE_15,
FC2_FRAMERATE_30,
FC2_FRAMERATE_60,
FC2_FRAMERATE_120,
FC2_FRAMERATE_240,
FC2_FRAMERATE_FORMAT7,
FC2_NUM_FRAMERATES,
FC2_FRAMERATE_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2VideoMode {
FC2_VIDEOMODE_160x120YUV444,
FC2_VIDEOMODE_320x240YUV422,
FC2_VIDEOMODE_640x480YUV411,
FC2_VIDEOMODE_640x480YUV422,
FC2_VIDEOMODE_640x480RGB,
FC2_VIDEOMODE_640x480Y8,
FC2_VIDEOMODE_640x480Y16,
FC2_VIDEOMODE_800x600YUV422,
FC2_VIDEOMODE_800x600RGB,
FC2_VIDEOMODE_800x600Y8,
FC2_VIDEOMODE_800x600Y16,
FC2_VIDEOMODE_1024x768YUV422,
FC2_VIDEOMODE_1024x768RGB,
FC2_VIDEOMODE_1024x768Y8,
FC2_VIDEOMODE_1024x768Y16,
FC2_VIDEOMODE_1280x960YUV422,
FC2_VIDEOMODE_1280x960RGB,
FC2_VIDEOMODE_1280x960Y8,
FC2_VIDEOMODE_1280x960Y16,
FC2_VIDEOMODE_1600x1200YUV422,
FC2_VIDEOMODE_1600x1200RGB,
FC2_VIDEOMODE_1600x1200Y8,

FC2_VIDEOMODE_1600x1200Y16,
FC2_VIDEOMODE_FORMAT7,
FC2_NUM_VIDEOMODES,
FC2_VIDEOMODE_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2Mode {
FC2_MODE_0 = 0,
FC2_MODE_1,
FC2_MODE_2,
FC2_MODE_3,
FC2_MODE_4,
FC2_MODE_5,
FC2_MODE_6,
FC2_MODE_7,
FC2_MODE_8,
FC2_MODE_9,
FC2_MODE_10,
FC2_MODE_11,
FC2_MODE_12,
FC2_MODE_13,
FC2_MODE_14,
FC2_MODE_15,
FC2_MODE_16,
FC2_MODE_17,
FC2_MODE_18,
FC2_MODE_19,
FC2_MODE_20,
FC2_MODE_21,
FC2_MODE_22,
FC2_MODE_23,
FC2_MODE_24,
FC2_MODE_25,
FC2_MODE_26,
FC2_MODE_27,
FC2_MODE_28,
FC2_MODE_29,
FC2_MODE_30,
FC2_MODE_31,
FC2_NUM_MODES,
FC2_MODE_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2PixelFormat {
  FC2_PIXEL_FORMAT_MONO8 = 0x80000000,
  FC2_PIXEL_FORMAT_411YUV8 = 0x40000000,
  FC2_PIXEL_FORMAT_422YUV8 = 0x20000000,
  FC2_PIXEL_FORMAT_444YUV8 = 0x10000000,
  FC2_PIXEL_FORMAT_RGB8 = 0x08000000,
  FC2_PIXEL_FORMAT_MONO16 = 0x04000000,
  FC2_PIXEL_FORMAT_RGB16 = 0x02000000,
  FC2_PIXEL_FORMAT_S_MONO16 = 0x01000000,
  FC2_PIXEL_FORMAT_S_RGB16 = 0x00800000,
  FC2_PIXEL_FORMAT_RAW8 = 0x00400000,
  FC2_PIXEL_FORMAT_RAW16 = 0x00200000,
  FC2_PIXEL_FORMAT_MONO12 = 0x00100000,
  FC2_PIXEL_FORMAT_RAW12 = 0x00080000,
  FC2_PIXEL_FORMAT_BGR = 0x80000008,
  FC2_PIXEL_FORMAT_BGRU = 0x40000008,
  FC2_PIXEL_FORMAT_RGB = FC2_PIXEL_FORMAT_RGB8,
  FC2_PIXEL_FORMAT_RGBU = 0x40000002,
  FC2_NUM_PIXEL_FORMATS = 15,
  FC2_UNSPECIFIED_PIXEL_FORMAT = 0 }
- enum fc2BusSpeed {
  FC2_BUSSPEED_S100,
  FC2_BUSSPEED_S200,
  FC2_BUSSPEED_S400,
  FC2_BUSSPEED_S480,
  FC2_BUSSPEED_S800,
  FC2_BUSSPEED_S1600,
  FC2_BUSSPEED_S3200,
  FC2_BUSSPEED_10BASE_T,
  FC2_BUSSPEED_100BASE_T,
  FC2_BUSSPEED_1000BASE_T,
  FC2_BUSSPEED_10000BASE_T,
  FC2_BUSSPEED_S_FASTEST,
  FC2_BUSSPEED_ANY,
  FC2_BUSSPEED_SPEED_UNKNOWN = -1,
  FC2_BUSSPEED_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2ColorProcessingAlgorithm {
  FC2_DEFAULT,
  FC2_NO_COLOR_PROCESSING,
  FC2_NEAREST_NEIGHBOR_FAST,
  FC2_EDGE_SENSING,

FC2_HQ_LINEAR,
FC2_RIGOROUS,
FC2_COLOR_PROCESSING_ALGORITHM_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2BayerTileFormat {
FC2_BT_NONE,
FC2_BT_RGGB,
FC2_BT_GRBG,
FC2_BT_GBRG,
FC2_BT_BGGR,
FC2_BT_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2ImageFileFormat {
FC2_FROM_FILE_EXT = -1,
FC2_PGM,
FC2_PPM,
FC2_BMP,
FC2_JPEG,
FC2_JPEG2000,
FC2_TIFF,
FC2_PNG,
FC2_RAW,
FC2_IMAGE_FILE_FORMAT_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2GigEPropertyType {
FC2_HEARTBEAT,
FC2_HEARTBEAT_TIMEOUT }
- enum fc2StatisticsChannel {
FC2_STATISTICS_GREY,
FC2_STATISTICS_RED,
FC2_STATISTICS_GREEN,
FC2_STATISTICS_BLUE,
FC2_STATISTICS_HUE,
FC2_STATISTICS_SATURATION,
FC2_STATISTICS_LIGHTNESS,
FC2_STATISTICS_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2OSType {
FC2_WINDOWS_X86,
FC2_WINDOWS_X64,
FC2_LINUX_X86,
FC2_LINUX_X64,
FC2_MAC,
FC2_UNKNOWN_OS,
FC2_OSTYPE_FORCE_32BITS = FULL_32BIT_VALUE }

- enum fc2ByteOrder {
  FC2_BYTE_ORDER_LITTLE_ENDIAN,
  FC2_BYTE_ORDER_BIG_ENDIAN,
  FC2_BYTE_ORDER_FORCE_32BITS = FULL_32BIT_VALUE }
- enum fc2TIFFCompressionMethod {
  FC2_TIFF_NONE = 1,
  FC2_TIFF_PACKBITS,
  FC2_TIFF_DEFLATE,
  FC2_TIFF_ADOBE_DEFLATE,
  FC2_TIFF_CCITTFAX3,
  FC2_TIFF_CCITTFAX4,
  FC2_TIFF_LZW,
  FC2_TIFF_JPEG }

4.2.1 Define Documentation

4.2.1.1 #define FALSE 0

4.2.1.2 #define FULL_32BIT_VALUE 0x7FFFFFFF

4.2.1.3 #define MAX_STRING_LENGTH 512

4.2.1.4 #define TRUE 1

4.2.2 Typedef Documentation

4.2.2.1 typedef int BOOL

4.2.2.2 typedef void(∗ fc2AsyncCommandCallback)(fc2Error retError, void ∗pUserData)

4.2.2.3 typedef void∗ fc2AVIContext

A context referring to the AVI recorder object.

4.2.2.4 typedef void(∗ fc2BusEventCallback)(void ∗pParameter, unsigned int serialNumber)

4.2.2.5 typedef void∗ fc2CallbackHandle

4.2.2.6 typedef void∗ fc2Context

A context to the FlyCapture2 C library.

It must be created before performing any calls to the library.

4.2.2.7 typedef void∗ fc2GuiContext

A context to the FlyCapture2 C GUI library.

It must be created before performing any calls to the library.

4.2.2.8 typedef void(∗ fc2ImageEventCallback)(fc2Image ∗image, void ∗pCallbackData)

4.2.2.9 typedef void∗ fc2ImageImpl

An internal pointer used in the fc2Image structure.

4.2.2.10 typedef void∗ fc2ImageStatisticsContext

A context referring to the ImageStatistics object.

4.2.3 Enumeration Type Documentation

4.2.3.1 enum fc2BandwidthAllocation

Enumerator:

FC2_BANDWIDTH_ALLOCATION_OFF

FC2_BANDWIDTH_ALLOCATION_ON

FC2_BANDWIDTH_ALLOCATION_UNSUPPORTED

FC2_BANDWIDTH_ALLOCATION_UNSPECIFIED

FC2_BANDWIDTH_ALLOCATION_FORCE_32BITS

4.2.3.2 enum fc2BayerTileFormat

Enumerator:

FC2_BT_NONE	No bayer tile format.
FC2_BT_RGGB	Red-Green-Green-Blue.
FC2_BT_GRBG	Green-Red-Blue-Green.
FC2_BT_GBRG	Green-Blue-Red-Green.
FC2_BT_BGGR	Blue-Green-Green-Red.
FC2_BT_FORCE_32BITS	

4.2.3.3 enum fc2BusCallbackType

Enumerator:

FC2_BUS_RESET

FC2_ARRIVAL

FC2_REMOVAL

FC2_CALLBACK_TYPE_FORCE_32BITS

4.2.3.4 enum fc2BusSpeed

Enumerator:

FC2_BUSSPEED_S100 100Mbits/sec.

FC2_BUSSPEED_S200 200Mbits/sec.

FC2_BUSSPEED_S400 400Mbits/sec.

FC2_BUSSPEED_S480 480Mbits/sec.
Only for USB cameras.

FC2_BUSSPEED_S800 800Mbits/sec.

FC2_BUSSPEED_S1600 1600Mbits/sec.

FC2_BUSSPEED_S3200 3200Mbits/sec.

FC2_BUSSPEED_10BASE_T 10Base-T.
Only for GigE cameras.

FC2_BUSSPEED_100BASE_T 100Base-T.
Only for GigE cameras.

FC2_BUSSPEED_1000BASE_T 1000Base-T (Gigabit Ethernet).
Only for GigE cameras.

FC2_BUSSPEED_10000BASE_T 10000Base-T.
Only for GigE cameras.

FC2_BUSSPEED_S_FASTEST The fastest speed available.

FC2_BUSSPEED_ANY Any speed that is available.

FC2_BUSSPEED_SPEED_UNKNOWN Unknown bus speed.

FC2_BUSSPEED_FORCE_32BITS

4.2.3.5 enum fc2ByteOrder

Enumerator:

FC2_BYTE_ORDER_LITTLE_ENDIAN

FC2_BYTE_ORDER_BIG_ENDIAN

FC2_BYTE_ORDER_FORCE_32BITS

4.2.3.6 enum fc2ColorProcessingAlgorithm

Enumerator:

FC2_DEFAULT

FC2_NO_COLOR_PROCESSING

FC2_NEAREST_NEIGHBOR_FAST

FC2_EDGE_SENSING

FC2_HQ_LINEAR

FC2_RIGOROUS

FC2_COLOR_PROCESSING_ALGORITHM_FORCE_32BITS

4.2.3.7 enum fc2Error

Enumerator:

FC2_ERROR_UNDEFINED Undefined.

FC2_ERROR_OK Function returned with no errors.

FC2_ERROR_FAILED General failure.

FC2_ERROR_NOT_IMPLEMENTED Function has not been implemented.

FC2_ERROR_FAILED_BUS_MASTER_CONNECTION Could not connect to Bus Master.

FC2_ERROR_NOT_CONNECTED Camera has not been connected.

FC2_ERROR_INIT_FAILED Initialization failed.

FC2_ERROR_NOT_INTITIALIZED Camera has not been initialized.

FC2_ERROR_INVALID_PARAMETER Invalid parameter passed to function.

FC2_ERROR_INVALID_SETTINGS Setting set to camera is invalid.

FC2_ERROR_INVALID_BUS_MANAGER Invalid Bus Manager object.

FC2_ERROR_MEMORY_ALLOCATION_FAILED Could not allocate memory.

FC2_ERROR_LOW_LEVEL_FAILURE Low level error.

FC2_ERROR_NOT_FOUND Device not found.

FC2_ERROR_FAILED_GUID GUID failure.

FC2_ERROR_INVALID_PACKET_SIZE Packet size set to camera is invalid.

FC2_ERROR_INVALID_MODE Invalid mode has been passed to function.

FC2_ERROR_NOT_IN_FORMAT7 Error due to not being in Format7.

FC2_ERROR_NOT_SUPPORTED This feature is unsupported.

FC2_ERROR_TIMEOUT Timeout error.

FC2_ERROR_BUS_MASTER_FAILED Bus Master Failure.

FC2_ERROR_INVALID_GENERATION Generation Count Mismatch.

FC2_ERROR_LUT_FAILED Look Up Table failure.

FC2_ERROR_IIDC_FAILED IIDC failure.

FC2_ERROR_STROBE_FAILED Strobe failure.

FC2_ERROR_TRIGGER_FAILED Trigger failure.

FC2_ERROR_PROPERTY_FAILED Property failure.

FC2_ERROR_PROPERTY_NOT_PRESENT Property is not present.

FC2_ERROR_REGISTER_FAILED Register access failed.

FC2_ERROR_READ_REGISTER_FAILED Register read failed.

FC2_ERROR_WRITE_REGISTER_FAILED Register write failed.

FC2_ERROR_ISOCH_FAILED Isochronous failure.

FC2_ERROR_ISOCH_ALREADY_STARTED Isochronous transfer has already been started.

FC2_ERROR_ISOCH_NOT_STARTED Isochronous transfer has not been started.

FC2_ERROR_ISOCH_START_FAILED Isochronous start failed.

FC2_ERROR_ISOCH_RETRIEVE_BUFFER_FAILED Isochronous retrieve buffer failed.

FC2_ERROR_ISOCH_STOP_FAILED Isochronous stop failed.

FC2_ERROR_ISOCH_SYNC_FAILED Isochronous image synchronization failed.

FC2_ERROR_ISOCH_BANDWIDTH_EXCEEDED Isochronous bandwidth exceeded.

FC2_ERROR_IMAGE_CONVERSION_FAILED Image conversion failed.

FC2_ERROR_IMAGE_LIBRARY_FAILURE Image library failure.

FC2_ERROR_BUFFER_TOO_SMALL Buffer is too small.

FC2_ERROR_IMAGE_CONSISTENCY_ERROR There is an image consistency error.

FC2_ERROR_FORCE_32BITS

4.2.3.8 enum fc2FrameRate

Enumerator:

FC2_FRAMERATE_1_875 1.875 fps.

FC2_FRAMERATE_3_75 3.75 fps.

FC2_FRAMERATE_7_5 7.5 fps.

FC2_FRAMERATE_15 15 fps.

FC2_FRAMERATE_30 30 fps.

FC2_FRAMERATE_60 60 fps.

FC2_FRAMERATE_120 120 fps.

FC2_FRAMERATE_240 240 fps.

FC2_FRAMERATE_FORMAT7 Custom frame rate for Format7 functionality.

FC2_NUM_FRAMERATES Number of possible camera frame rates.

FC2_FRAMERATE_FORCE_32BITS

4.2.3.9 enum fc2GigEPropertyType

Enumerator:

FC2_HEARTBEAT

FC2_HEARTBEAT_TIMEOUT

4.2.3.10 enum fc2GrabMode

Enumerator:

FC2_DROP_FRAMES

FC2_BUFFER_FRAMES

FC2_UNSPECIFIED_GRAB_MODE

FC2_GRAB_MODE_FORCE_32BITS

4.2.3.11 enum fc2GrabTimeout

Enumerator:

FC2_TIMEOUT_NONE

FC2_TIMEOUT_INFINITE

FC2_TIMEOUT_UNSPECIFIED

FC2_GRAB_TIMEOUT_FORCE_32BITS

4.2.3.12 enum fc2ImageFileFormat

Enumerator:

FC2_FROM_FILE_EXT Determine file format from file extension.

FC2_PGM Portable gray map.

FC2_PPM Portable pixmap.

FC2_BMP Bitmap.

FC2_JPEG JPEG.

FC2_JPEG2000 JPEG 2000.

FC2_TIFF Tagged image file format.

FC2_PNG Portable network graphics.

FC2_RAW Raw data.

FC2_IMAGE_FILE_FORMAT_FORCE_32BITS

4.2.3.13 enum fc2InterfaceType

Enumerator:

FC2_INTERFACE_IEEE1394

FC2_INTERFACE_USB_2

FC2_INTERFACE_GIGE

FC2_INTERFACE_UNKNOWN

FC2_INTERFACE_TYPE_FORCE_32BITS

4.2.3.14 enum fc2Mode

Enumerator:

FC2_MODE_0

FC2_MODE_1

FC2_MODE_2

FC2_MODE_3

FC2_MODE_4

FC2_MODE_5

FC2_MODE_6

FC2_MODE_7

FC2_MODE_8

FC2_MODE_9

FC2_MODE_10

FC2_MODE_11

FC2_MODE_12

FC2_MODE_13

FC2_MODE_14

FC2_MODE_15

FC2_MODE_16

FC2_MODE_17

FC2_MODE_18

FC2_MODE_19

FC2_MODE_20

FC2_MODE_21

FC2_MODE_22

FC2_MODE_23

FC2_MODE_24

FC2_MODE_25

FC2_MODE_26

FC2_MODE_27

FC2_MODE_28

FC2_MODE_29

FC2_MODE_30

FC2_MODE_31

FC2_NUM_MODES Number of modes.

FC2_MODE_FORCE_32BITS

4.2.3.15 enum fc2OSType

Enumerator:

FC2_WINDOWS_X86

FC2_WINDOWS_X64

FC2_LINUX_X86

FC2_LINUX_X64

FC2_MAC

FC2_UNKNOWN_OS

FC2_OSTYPE_FORCE_32BITS

4.2.3.16 enum fc2PixelFormat

Enumerator:

FC2_PIXEL_FORMAT_MONO8 8 bits of mono information.

FC2_PIXEL_FORMAT_411YUV8 YUV 4:1:1.

FC2_PIXEL_FORMAT_422YUV8 YUV 4:2:2.

FC2_PIXEL_FORMAT_444YUV8 YUV 4:4:4.

FC2_PIXEL_FORMAT_RGB8 R = G = B = 8 bits.

FC2_PIXEL_FORMAT_MONO16 16 bits of mono information.

FC2_PIXEL_FORMAT_RGB16 R = G = B = 16 bits.

FC2_PIXEL_FORMAT_S_MONO16 16 bits of signed mono information.

FC2_PIXEL_FORMAT_S_RGB16 R = G = B = 16 bits signed.

FC2_PIXEL_FORMAT_RAW8 8 bit raw data output of sensor.

FC2_PIXEL_FORMAT_RAW16 16 bit raw data output of sensor.

FC2_PIXEL_FORMAT_MONO12 12 bits of mono information.

FC2_PIXEL_FORMAT_RAW12 12 bit raw data output of sensor.

FC2_PIXEL_FORMAT_BGR 24 bit BGR.

FC2_PIXEL_FORMAT_BGRU 32 bit BGRU.

FC2_PIXEL_FORMAT_RGB 24 bit RGB.

FC2_PIXEL_FORMAT_RGBU 32 bit RGBU.

FC2_NUM_PIXEL_FORMATS Number of pixel formats.

FC2_UNSPECIFIED_PIXEL_FORMAT Unspecified pixel format.

4.2.3.17 enum fc2PropertyType

Enumerator:

FC2_BRIGHTNESS

FC2_AUTO_EXPOSURE

FC2_SHARPNESS

FC2_WHITE_BALANCE

FC2_HUE

FC2_SATURATION

FC2_GAMMA

FC2_IRIS

FC2_FOCUS

FC2_ZOOM

FC2_PAN

FC2_TILT

FC2_SHUTTER

FC2_GAIN

FC2_TRIGGER_MODE

FC2_TRIGGER_DELAY

FC2_FRAME_RATE

FC2_TEMPERATURE

FC2_UNSPECIFIED_PROPERTY_TYPE

FC2_PROPERTY_TYPE_FORCE_32BITS

4.2.3.18 enum fc2StatisticsChannel

Enumerator:

FC2_STATISTICS_GREY

FC2_STATISTICS_RED

FC2_STATISTICS_GREEN

FC2_STATISTICS_BLUE

FC2_STATISTICS_HUE

FC2_STATISTICS_SATURATION

FC2_STATISTICS_LIGHTNESS

FC2_STATISTICS_FORCE_32BITS

4.2.3.19 enum fc2TIFFCompressionMethod

Enumerator:

FC2_TIFF_NONE

FC2_TIFF_PACKBITS

FC2_TIFF_DEFLATE

FC2_TIFF_ADOBE_DEFLATE

FC2_TIFF_CCITTFAX3

FC2_TIFF_CCITTFAX4

FC2_TIFF_LZW

FC2_TIFF_JPEG

4.2.3.20 enum fc2VideoMode

Enumerator:

FC2_VIDEOMODE_160x120YUV444	160x120 YUV444.
FC2_VIDEOMODE_320x240YUV422	320x240 YUV422.
FC2_VIDEOMODE_640x480YUV411	640x480 YUV411.
FC2_VIDEOMODE_640x480YUV422	640x480 YUV422.
FC2_VIDEOMODE_640x480RGB	640x480 24-bit RGB.
FC2_VIDEOMODE_640x480Y8	640x480 8-bit.
FC2_VIDEOMODE_640x480Y16	640x480 16-bit.
FC2_VIDEOMODE_800x600YUV422	800x600 YUV422.
FC2_VIDEOMODE_800x600RGB	800x600 RGB.
FC2_VIDEOMODE_800x600Y8	800x600 8-bit.
FC2_VIDEOMODE_800x600Y16	800x600 16-bit.
FC2_VIDEOMODE_1024x768YUV422	1024x768 YUV422.
FC2_VIDEOMODE_1024x768RGB	1024x768 RGB.
FC2_VIDEOMODE_1024x768Y8	1024x768 8-bit.

FC2_VIDEOMODE_1024x768Y16 1024x768 16-bit.

FC2_VIDEOMODE_1280x960YUV422 1280x960 YUV422.

FC2_VIDEOMODE_1280x960RGB 1280x960 RGB.

FC2_VIDEOMODE_1280x960Y8 1280x960 8-bit.

FC2_VIDEOMODE_1280x960Y16 1280x960 16-bit.

FC2_VIDEOMODE_1600x1200YUV422 1600x1200 YUV422.

FC2_VIDEOMODE_1600x1200RGB 1600x1200 RGB.

FC2_VIDEOMODE_1600x1200Y8 1600x1200 8-bit.

FC2_VIDEOMODE_1600x1200Y16 1600x1200 16-bit.

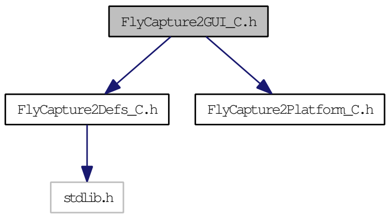
FC2_VIDEOMODE_FORMAT7 Custom video mode for Format7 functionality.

FC2_NUM_VIDEOMODES Number of possible video modes.

FC2_VIDEOMODE_FORCE_32BITS

4.3 FlyCapture2GUI_C.h File Reference

Include dependency graph for FlyCapture2GUI_C.h:



Functions

- FLYCAPTURE2_C_API fc2Error fc2CreateGUIContext (fc2GuiContext ∗pContext)

  Create a GUI context.

- FLYCAPTURE2_C_API fc2Error fc2DestroyGUIContext (fc2GuiContext context)

  Destroy a GUI context.

- FLYCAPTURE2_C_API void fc2GUIConnect (fc2GuiContext context, fc2Context cameraContext)

  Connect GUI context to a camera context.

- FLYCAPTURE2_C_API void fc2Disonnect (fc2GuiContext context)

  Disconnect GUI context from camera.

- FLYCAPTURE2_C_API void fc2Show (fc2GuiContext context)

  Show the GUI.

- FLYCAPTURE2_C_API void fc2Hide (fc2GuiContext context)

  Hide the GUI.

- FLYCAPTURE2_C_API BOOL fc2IsVisible (fc2GuiContext context)

  Check if the GUI is visible.

- FLYCAPTURE2_C_API void fc2ShowModal (fc2GuiContext context, BOOL ∗pOkSelected, fc2PGRGuid ∗guidArray, unsigned int ∗size)

  Show the camera selection dialog.

4.3.1 Function Documentation

4.3.1.1 FLYCAPTURE2_C_API fc2Error fc2CreateGUIContext (fc2GuiContext ∗ *pContext*)

Create a GUI context.

Parameters:

pContext	Pointer to context to be created.

Returns:

An Error indicating the success or failure of the function.

4.3.1.2 FLYCAPTURE2_C_API fc2Error fc2DestroyGUIContext (fc2GuiContext *context*)

Destroy a GUI context.

Parameters:

context Context to be destroyed.

Returns:

An Error indicating the success or failure of the function.

4.3.1.3 FLYCAPTURE2_C_API void fc2Disonnect (fc2GuiContext *context*)

Disconnect GUI context from camera.

Parameters:

context GUI context to disconnect.

Returns:

An Error indicating the success or failure of the function.

4.3.1.4 FLYCAPTURE2_C_API void fc2GUIConnect (fc2GuiContext *context*, fc2Context *cameraContext*)

Connect GUI context to a camera context.

Parameters:

context GUI context to connect.

cameraContext Camera context to connect.

Returns:

An Error indicating the success or failure of the function.

4.3.1.5 FLYCAPTURE2_C_API void fc2Hide (fc2GuiContext *context*)

Hide the GUI.

Parameters:

context Pointer to context to hide.

Returns:

An Error indicating the success or failure of the function.

4.3.1.6 FLYCAPTURE2_C_API BOOL fc2IsVisible (fc2GuiContext *context*)

Check if the GUI is visible.

Parameters:

context Pointer to context to show.

Returns:

Whether the GUI is visible.

4.3.1.7 FLYCAPTURE2_C_API void fc2Show (fc2GuiContext *context*)

Show the GUI.

Parameters:

context Pointer to context to show.

Returns:

An Error indicating the success or failure of the function.

4.3.1.8 FLYCAPTURE2_C_API void fc2ShowModal (fc2GuiContext *context*, BOOL ∗ *pOkSelected*, fc2PGRGuid ∗ *guidArray*, unsigned int ∗ *size*)

Show the camera selection dialog.

Parameters:

context Pointer to context to show.

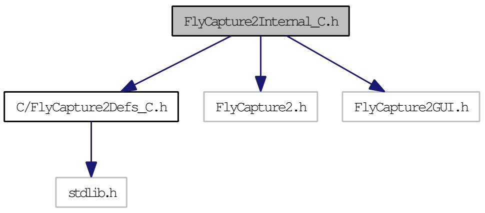
pOkSelected Whether Ok (true) or Cancel (false) was clicked.

guidArray Array of PGRGuids containing the selected cameras.

size Size of PGRGuid array.

4.4 FlyCapture2Internal_C.h File Reference

Include dependency graph for FlyCapture2Internal_C.h:



Data Structures

- struct fc2InternalContext
- struct fc2InternalGuiContext
- struct fc2InternalImageCallback

Functions

- bool IsContextValid (fc2Context context)
- bool IsGuiContextValid (fc2GuiContext context)
- void SyncCppImageToStruct (fc2Image ∗pImage)

4.4.1 Function Documentation

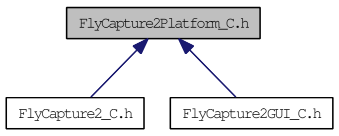
4.4.1.1 bool IsContextValid (fc2Context *context*) `[inline]`

4.4.1.2 bool IsGuiContextValid (fc2GuiContext *context*) `[inline]`

4.4.1.3 void SyncCppImageToStruct (fc2Image ∗ *pImage*) `[inline]`

4.5 FlyCapture2Platform_C.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define FLYCAPTURE2_C_API
- #define FLYCAPTURE2_C_CALL_CONVEN

4.5.1 Define Documentation

4.5.1.1 #define FLYCAPTURE2_C_API

4.5.1.2 #define FLYCAPTURE2_C_CALL_CONVEN

Index

absControl
    fc2TriggerDelay, 40
absMax
    fc2TriggerDelayInfo, 42
absMin
    fc2TriggerDelayInfo, 42
absValSupported
    fc2TriggerDelayInfo, 42
absValue
    fc2TriggerDelay, 40
asyncBusSpeed
    fc2Config, 8
autoManualMode
    fc2TriggerDelay, 40
autoSupported
    fc2TriggerDelayInfo, 42
available
    fc2EmbeddedImageInfoProperty, 12

bandwidthAllocation
    fc2Config, 8
bayerFormat
    fc2Image, 21
bayerTileFormat
    fc2CameraInfo, 7
binaryFile
    fc2PGMOption, 31
    fc2PPMOption, 34
BOOL
    FlyCapture2Defs_C.h, 100
brightness
    fc2EmbeddedImageInfo, 11
build
    fc2Version, 45
byteOrder
    fc2SystemInfo, 37

channels
    fc2GigEConfig, 16
chipIdHi
    fc2ConfigROM, 9
chipIdLo
    fc2ConfigROM, 9
cols
    fc2Image, 21
compression
    fc2TIFFOption, 38
compressionLevel
    fc2PNGOption, 33
configROM
    fc2CameraInfo, 7
cpuDescription
    fc2SystemInfo, 37
cycleCount
    fc2TimeStamp, 39
cycleOffset
    fc2TimeStamp, 39
cycleSeconds
    fc2TimeStamp, 39

dataSize
    fc2Image, 21
defaultGateway
    fc2CameraInfo, 7
delay
    fc2StrobeControl, 35
destinationIpAddress
    fc2GigEStreamChannel, 20
doNotFragment
    fc2GigEStreamChannel, 20
driverList
    fc2SystemInfo, 37
driverName
    fc2CameraInfo, 7
duration
    fc2StrobeControl, 35

embeddedBrightness
    fc2ImageMetadata, 22
embeddedExposure
    fc2ImageMetadata, 22
embeddedFrameCounter
    fc2ImageMetadata, 22
embeddedGain
    fc2ImageMetadata, 22
embeddedGPIOPinState
    fc2ImageMetadata, 22
embeddedROIPosition
    fc2ImageMetadata, 22
embeddedShutter

fc2ImageMetadata, 22
embeddedStrobePattern
fc2ImageMetadata, 22
embeddedTimeStamp
fc2ImageMetadata, 22
embeddedWhiteBalance
fc2ImageMetadata, 22
enabled
fc2LUTData, 29
exposure
fc2EmbeddedImageInfo, 11

FALSE
FlyCapture2Defs_C.h, 100
FC2_ARRIVAL
FlyCapture2Defs_C.h, 101
FC2_AUTO_EXPOSURE
FlyCapture2Defs_C.h, 107
FC2_BANDWIDTH_ALLOCATION_FORCE_-32BITS
FlyCapture2Defs_C.h, 101
FC2_BANDWIDTH_ALLOCATION_OFF
FlyCapture2Defs_C.h, 101
FC2_BANDWIDTH_ALLOCATION_ON
FlyCapture2Defs_C.h, 101
FC2_BANDWIDTH_ALLOCATION_-UNSPECIFIED
FlyCapture2Defs_C.h, 101
FC2_BANDWIDTH_ALLOCATION_-UNSUPPORTED
FlyCapture2Defs_C.h, 101
FC2_BMP
FlyCapture2Defs_C.h, 105
FC2_BRIGHTNESS
FlyCapture2Defs_C.h, 107
FC2_BT_BGGR
FlyCapture2Defs_C.h, 101
FC2_BT_FORCE_32BITS
FlyCapture2Defs_C.h, 101
FC2_BT_GBRG
FlyCapture2Defs_C.h, 101
FC2_BT_GRBG
FlyCapture2Defs_C.h, 101
FC2_BT_NONE
FlyCapture2Defs_C.h, 101
FC2_BT_RGGB
FlyCapture2Defs_C.h, 101
FC2_BUFFER_FRAMES
FlyCapture2Defs_C.h, 104
FC2_BUS_RESET
FlyCapture2Defs_C.h, 101
FC2_BUSSPEED_10000BASE_T
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_1000BASE_T
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_100BASE_T
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_10BASE_T
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_ANY
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_FORCE_32BITS
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S100
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S1600
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S200
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S3200
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S400
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S480
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S800
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_S_FASTEST
FlyCapture2Defs_C.h, 102
FC2_BUSSPEED_SPEED_UNKNOWN
FlyCapture2Defs_C.h, 102
FC2_BYTE_ORDER_BIG_ENDIAN
FlyCapture2Defs_C.h, 102
FC2_BYTE_ORDER_FORCE_32BITS
FlyCapture2Defs_C.h, 102
FC2_BYTE_ORDER_LITTLE_ENDIAN
FlyCapture2Defs_C.h, 102
FC2_CALLBACK_TYPE_FORCE_32BITS
FlyCapture2Defs_C.h, 101
FC2_COLOR_PROCESSING_ALGORITHM_-FORCE_32BITS
FlyCapture2Defs_C.h, 102
FC2_DEFAULT
FlyCapture2Defs_C.h, 102
FC2_DROP_FRAMES
FlyCapture2Defs_C.h, 104
FC2_EDGE_SENSING
FlyCapture2Defs_C.h, 102
FC2_ERROR_BUFFER_TOO_SMALL
FlyCapture2Defs_C.h, 104
FC2_ERROR_BUS_MASTER_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_FAILED_BUS_MASTER_-CONNECTION
FlyCapture2Defs_C.h, 103
FC2_ERROR_FAILED_GUID

FlyCapture2Defs_C.h, 103
FC2_ERROR_FORCE_32BITS
FlyCapture2Defs_C.h, 104
FC2_ERROR_IIDC_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_IMAGE_CONSISTENCY_ERROR
FlyCapture2Defs_C.h, 104
FC2_ERROR_IMAGE_CONVERSION_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_IMAGE_LIBRARY_FAILURE
FlyCapture2Defs_C.h, 104
FC2_ERROR_INIT_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_INVALID_BUS_MANAGER
FlyCapture2Defs_C.h, 103
FC2_ERROR_INVALID_GENERATION
FlyCapture2Defs_C.h, 103
FC2_ERROR_INVALID_MODE
FlyCapture2Defs_C.h, 103
FC2_ERROR_INVALID_PACKET_SIZE
FlyCapture2Defs_C.h, 103
FC2_ERROR_INVALID_PARAMETER
FlyCapture2Defs_C.h, 103
FC2_ERROR_INVALID_SETTINGS
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_ALREADY_STARTED
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_BANDWIDTH_-
EXCEEDED
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_NOT_STARTED
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_RETRIEVE_BUFFER_-
FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_START_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_STOP_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_ISOCH_SYNC_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_LOW_LEVEL_FAILURE
FlyCapture2Defs_C.h, 103
FC2_ERROR_LUT_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_MEMORY_ALLOCATION_-
FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_NOT_CONNECTED
FlyCapture2Defs_C.h, 103
FC2_ERROR_NOT_FOUND
FlyCapture2Defs_C.h, 103
FC2_ERROR_NOT_IMPLEMENTED
FlyCapture2Defs_C.h, 103
FC2_ERROR_NOT_IN_FORMAT7
FlyCapture2Defs_C.h, 103
FC2_ERROR_NOT_INTITIALIZED
FlyCapture2Defs_C.h, 103
FC2_ERROR_NOT_SUPPORTED
FlyCapture2Defs_C.h, 103
FC2_ERROR_OK
FlyCapture2Defs_C.h, 103
FC2_ERROR_PROPERTY_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_PROPERTY_NOT_PRESENT
FlyCapture2Defs_C.h, 103
FC2_ERROR_READ_REGISTER_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_REGISTER_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_STROBE_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_TIMEOUT
FlyCapture2Defs_C.h, 103
FC2_ERROR_TRIGGER_FAILED
FlyCapture2Defs_C.h, 103
FC2_ERROR_UNDEFINED
FlyCapture2Defs_C.h, 103
FC2_ERROR_WRITE_REGISTER_FAILED
FlyCapture2Defs_C.h, 103
FC2_FOCUS
FlyCapture2Defs_C.h, 107
FC2_FRAME_RATE
FlyCapture2Defs_C.h, 107
FC2_FRAMERATE_120
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_15
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_1_875
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_240
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_30
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_3_75
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_60
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_7_5
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_FORCE_32BITS
FlyCapture2Defs_C.h, 104
FC2_FRAMERATE_FORMAT7
FlyCapture2Defs_C.h, 104
FC2_FROM_FILE_EXT
FlyCapture2Defs_C.h, 105

FC2_GAIN
FlyCapture2Defs_C.h, 107
FC2_GAMMA
FlyCapture2Defs_C.h, 107
FC2_GRAB_MODE_FORCE_32BITS
FlyCapture2Defs_C.h, 104
FC2_GRAB_TIMEOUT_FORCE_32BITS
FlyCapture2Defs_C.h, 104
FC2_HEARTBEAT
FlyCapture2Defs_C.h, 104
FC2_HEARTBEAT_TIMEOUT
FlyCapture2Defs_C.h, 104
FC2_HQ_LINEAR
FlyCapture2Defs_C.h, 102
FC2_HUE
FlyCapture2Defs_C.h, 107
FC2_IMAGE_FILE_FORMAT_FORCE_32BITS
FlyCapture2Defs_C.h, 105
FC2_INTERFACE_GIGE
FlyCapture2Defs_C.h, 105
FC2_INTERFACE_IEEE1394
FlyCapture2Defs_C.h, 105
FC2_INTERFACE_TYPE_FORCE_32BITS
FlyCapture2Defs_C.h, 105
FC2_INTERFACE_UNKNOWN
FlyCapture2Defs_C.h, 105
FC2_INTERFACE_USB_2
FlyCapture2Defs_C.h, 105
FC2_IRIS
FlyCapture2Defs_C.h, 107
FC2_JPEG
FlyCapture2Defs_C.h, 105
FC2_JPEG2000
FlyCapture2Defs_C.h, 105
FC2_LINUX_X64
FlyCapture2Defs_C.h, 106
FC2_LINUX_X86
FlyCapture2Defs_C.h, 106
FC2_MAC
FlyCapture2Defs_C.h, 106
FC2_MODE_0
FlyCapture2Defs_C.h, 105
FC2_MODE_1
FlyCapture2Defs_C.h, 105
FC2_MODE_10
FlyCapture2Defs_C.h, 105
FC2_MODE_11
FlyCapture2Defs_C.h, 105
FC2_MODE_12
FlyCapture2Defs_C.h, 105
FC2_MODE_13
FlyCapture2Defs_C.h, 105
FC2_MODE_14
FlyCapture2Defs_C.h, 105
FC2_MODE_15
FlyCapture2Defs_C.h, 105
FC2_MODE_16
FlyCapture2Defs_C.h, 106
FC2_MODE_17
FlyCapture2Defs_C.h, 106
FC2_MODE_18
FlyCapture2Defs_C.h, 106
FC2_MODE_19
FlyCapture2Defs_C.h, 106
FC2_MODE_2
FlyCapture2Defs_C.h, 105
FC2_MODE_20
FlyCapture2Defs_C.h, 106
FC2_MODE_21
FlyCapture2Defs_C.h, 106
FC2_MODE_22
FlyCapture2Defs_C.h, 106
FC2_MODE_23
FlyCapture2Defs_C.h, 106
FC2_MODE_24
FlyCapture2Defs_C.h, 106
FC2_MODE_25
FlyCapture2Defs_C.h, 106
FC2_MODE_26
FlyCapture2Defs_C.h, 106
FC2_MODE_27
FlyCapture2Defs_C.h, 106
FC2_MODE_28
FlyCapture2Defs_C.h, 106
FC2_MODE_29
FlyCapture2Defs_C.h, 106
FC2_MODE_3
FlyCapture2Defs_C.h, 105
FC2_MODE_30
FlyCapture2Defs_C.h, 106
FC2_MODE_31
FlyCapture2Defs_C.h, 106
FC2_MODE_4
FlyCapture2Defs_C.h, 105
FC2_MODE_5
FlyCapture2Defs_C.h, 105
FC2_MODE_6
FlyCapture2Defs_C.h, 105
FC2_MODE_7
FlyCapture2Defs_C.h, 105
FC2_MODE_8
FlyCapture2Defs_C.h, 105
FC2_MODE_9
FlyCapture2Defs_C.h, 105
FC2_MODE_FORCE_32BITS
FlyCapture2Defs_C.h, 106
FC2_NEAREST_NEIGHBOR_FAST
FlyCapture2Defs_C.h, 102

FC2_NO_COLOR_PROCESSING
FlyCapture2Defs_C.h, 102
FC2_NUM_FRAMERATES
FlyCapture2Defs_C.h, 104
FC2_NUM_MODES
FlyCapture2Defs_C.h, 106
FC2_NUM_PIXEL_FORMATS
FlyCapture2Defs_C.h, 107
FC2_NUM_VIDEOMODES
FlyCapture2Defs_C.h, 109
FC2_OSTYPE_FORCE_32BITS
FlyCapture2Defs_C.h, 106
FC2_PAN
FlyCapture2Defs_C.h, 107
FC2_PGM
FlyCapture2Defs_C.h, 105
FC2_PIXEL_FORMAT_411YUV8
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_422YUV8
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_444YUV8
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_BGR
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_BGRU
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_MONO12
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_MONO16
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_MONO8
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_RAW12
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_RAW16
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_RAW8
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_RGB
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_RGB16
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_RGB8
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_RGBU
FlyCapture2Defs_C.h, 107
FC2_PIXEL_FORMAT_S_MONO16
FlyCapture2Defs_C.h, 106
FC2_PIXEL_FORMAT_S_RGB16
FlyCapture2Defs_C.h, 107
FC2_PNG
FlyCapture2Defs_C.h, 105
FC2_PPM
FlyCapture2Defs_C.h, 105
FC2_PROPERTY_TYPE_FORCE_32BITS
FlyCapture2Defs_C.h, 107
FC2_RAW
FlyCapture2Defs_C.h, 105
FC2_REMOVAL
FlyCapture2Defs_C.h, 101
FC2_RIGOROUS
FlyCapture2Defs_C.h, 102
FC2_SATURATION
FlyCapture2Defs_C.h, 107
FC2_SHARPNESS
FlyCapture2Defs_C.h, 107
FC2_SHUTTER
FlyCapture2Defs_C.h, 107
FC2_STATISTICS_BLUE
FlyCapture2Defs_C.h, 108
FC2_STATISTICS_FORCE_32BITS
FlyCapture2Defs_C.h, 108
FC2_STATISTICS_GREEN
FlyCapture2Defs_C.h, 108
FC2_STATISTICS_GREY
FlyCapture2Defs_C.h, 108
FC2_STATISTICS_HUE
FlyCapture2Defs_C.h, 108
FC2_STATISTICS_LIGHTNESS
FlyCapture2Defs_C.h, 108
FC2_STATISTICS_RED
FlyCapture2Defs_C.h, 108
FC2_STATISTICS_SATURATION
FlyCapture2Defs_C.h, 108
FC2_TEMPERATURE
FlyCapture2Defs_C.h, 107
FC2_TIFF
FlyCapture2Defs_C.h, 105
FC2_TIFF_ADOBE_DEFLATE
FlyCapture2Defs_C.h, 108
FC2_TIFF_CCITTFAX3
FlyCapture2Defs_C.h, 108
FC2_TIFF_CCITTFAX4
FlyCapture2Defs_C.h, 108
FC2_TIFF_DEFLATE
FlyCapture2Defs_C.h, 108
FC2_TIFF_JPEG
FlyCapture2Defs_C.h, 108
FC2_TIFF_LZW
FlyCapture2Defs_C.h, 108
FC2_TIFF_NONE
FlyCapture2Defs_C.h, 108
FC2_TIFF_PACKBITS
FlyCapture2Defs_C.h, 108
FC2_TILT
FlyCapture2Defs_C.h, 107
FC2_TIMEOUT_INFINITE
FlyCapture2Defs_C.h, 104

FC2_TIMEOUT_NONE
FlyCapture2Defs_C.h, 104
FC2_TIMEOUT_UNSPECIFIED
FlyCapture2Defs_C.h, 104
FC2_TRIGGER_DELAY
FlyCapture2Defs_C.h, 107
FC2_TRIGGER_MODE
FlyCapture2Defs_C.h, 107
FC2_UNKNOWN_OS
FlyCapture2Defs_C.h, 106
FC2_UNSPECIFIED_GRAB_MODE
FlyCapture2Defs_C.h, 104
FC2_UNSPECIFIED_PIXEL_FORMAT
FlyCapture2Defs_C.h, 107
FC2_UNSPECIFIED_PROPERTY_TYPE
FlyCapture2Defs_C.h, 107
FC2_VIDEOMODE_1024x768RGB
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_1024x768Y16
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_1024x768Y8
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_1024x768YUV422
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_1280x960RGB
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_1280x960Y16
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_1280x960Y8
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_1280x960YUV422
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_1600x1200RGB
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_1600x1200Y16
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_1600x1200Y8
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_1600x1200YUV422
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_160x120YUV444
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_320x240YUV422
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_640x480RGB
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_640x480Y16
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_640x480Y8
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_640x480YUV411
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_640x480YUV422
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_800x600RGB
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_800x600Y16
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_800x600Y8
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_800x600YUV422
FlyCapture2Defs_C.h, 108
FC2_VIDEOMODE_FORCE_32BITS
FlyCapture2Defs_C.h, 109
FC2_VIDEOMODE_FORMAT7
FlyCapture2Defs_C.h, 109
FC2_WHITE_BALANCE
FlyCapture2Defs_C.h, 107
FC2_WINDOWS_X64
FlyCapture2Defs_C.h, 106
FC2_WINDOWS_X86
FlyCapture2Defs_C.h, 106
FC2_ZOOM
FlyCapture2Defs_C.h, 107
fc2AsyncCommandCallback
FlyCapture2Defs_C.h, 100
fc2AVIAppend
FlyCapture2_C.h, 56
fc2AVIClose
FlyCapture2_C.h, 56
fc2AVIContext
FlyCapture2Defs_C.h, 100
fc2AVIOpen
FlyCapture2_C.h, 57
fc2AVIOption, 5
frameRate, 5
reserved, 5
fc2BandwidthAllocation
FlyCapture2Defs_C.h, 101
fc2BayerTileFormat
FlyCapture2Defs_C.h, 101
fc2BusCallbackType
FlyCapture2Defs_C.h, 101
fc2BusEventCallback
FlyCapture2Defs_C.h, 100
fc2BusSpeed
FlyCapture2Defs_C.h, 101
fc2ByteOrder
FlyCapture2Defs_C.h, 102
fc2CalculateImageStatistics
FlyCapture2_C.h, 57
fc2CallbackHandle
FlyCapture2Defs_C.h, 100
fc2CameraInfo, 6
bayerTileFormat, 7
configROM, 7
defaultGateway, 7
driverName, 7

firmwareBuildTime, 7
firmwareVersion, 7
gigEMajorVersion, 7
gigEMinorVersion, 7
iidcVer, 7
interfaceType, 7
ipAddress, 7
isColorCamera, 7
macAddress, 7
maximumBusSpeed, 7
modelName, 7
reserved, 7
sensorInfo, 7
sensorResolution, 7
serialNumber, 7
subnetMask, 7
userDefinedName, 7
vendorName, 7
xmlURL1, 7
xmlURL2, 7
fc2ColorProcessingAlgorithm
FlyCapture2Defs_C.h, 102
fc2Config, 8
asyncBusSpeed, 8
bandwidthAllocation, 8
grabMode, 8
grabTimeout, 8
isochBusSpeed, 8
numBuffers, 8
numImageNotifications, 8
reserved, 8
fc2ConfigROM, 9
chipIdHi, 9
chipIdLo, 9
nodeVendorId, 9
pszKeyword, 9
reserved, 9
unitSpecId, 9
unitSubSWVer, 9
unitSWVer, 9
vendorUniqueInfo_0, 9
vendorUniqueInfo_1, 9
vendorUniqueInfo_2, 9
vendorUniqueInfo_3, 9
fc2Connect
FlyCapture2_C.h, 57
fc2Context
FlyCapture2Defs_C.h, 100
fc2ConvertImage
FlyCapture2_C.h, 58
fc2ConvertImageTo
FlyCapture2_C.h, 58
fc2CreateAVI
FlyCapture2_C.h, 58
fc2CreateContext
FlyCapture2_C.h, 58
fc2CreateGigEContext
FlyCapture2_C.h, 59
fc2CreateGUIContext
FlyCapture2GUI_C.h, 110
fc2CreateImage
FlyCapture2_C.h, 59
fc2CreateImageStatistics
FlyCapture2_C.h, 59
fc2DestroyAVI
FlyCapture2_C.h, 60
fc2DestroyContext
FlyCapture2_C.h, 60
fc2DestroyGUIContext
FlyCapture2GUI_C.h, 111
fc2DestroyImage
FlyCapture2_C.h, 60
fc2DestroyImageStatistics
FlyCapture2_C.h, 60
fc2DetermineBitsPerPixel
FlyCapture2_C.h, 61
fc2Disconnect
FlyCapture2_C.h, 61
fc2DiscoverGigECameras
FlyCapture2_C.h, 61
fc2Disonnect
FlyCapture2GUI_C.h, 111
fc2EmbeddedImageInfo, 10
brightness, 11
exposure, 11
frameCounter, 11
gain, 11
GPIOPinState, 11
ROIPosition, 11
shutter, 11
strobePattern, 11
timestamp, 11
whiteBalance, 11
fc2EmbeddedImageInfoProperty, 12
available, 12
onOff, 12
fc2EnableLUT
FlyCapture2_C.h, 61
fc2Error
FlyCapture2Defs_C.h, 102
fc2ErrorToDescription
FlyCapture2_C.h, 62
fc2FireBusReset
FlyCapture2_C.h, 62
fc2FireSoftwareTrigger
FlyCapture2_C.h, 62
fc2FireSoftwareTriggerBroadcast
FlyCapture2_C.h, 62

fc2ForceIPAddressToCamera
  FlyCapture2_C.h, 63
fc2Format7ImageSettings, 13
  height, 13
  mode, 13
  offsetX, 13
  offsetY, 13
  pixelFormat, 13
  reserved, 13
  width, 13
fc2Format7Info, 14
  imageHStepSize, 14
  imageVStepSize, 14
  maxHeight, 14
  maxPacketSize, 14
  maxWidth, 14
  minPacketSize, 14
  mode, 14
  offsetHStepSize, 14
  offsetVStepSize, 14
  packetSize, 14
  percentage, 14
  pixelFormatBitField, 14
  reserved, 14
fc2Format7PacketInfo, 15
  maxBytesPerPacket, 15
  recommendedBytesPerPacket, 15
  reserved, 15
  unitBytesPerPacket, 15
fc2FrameRate
  FlyCapture2Defs_C.h, 104
fc2GetActiveLUTBank
  FlyCapture2_C.h, 63
fc2GetCameraFromIndex
  FlyCapture2_C.h, 63
fc2GetCameraFromSerialNumber
  FlyCapture2_C.h, 64
fc2GetCameraInfo
  FlyCapture2_C.h, 64
fc2GetCameraSerialNumberFromIndex
  FlyCapture2_C.h, 64
fc2GetConfiguration
  FlyCapture2_C.h, 65
fc2GetDefaultColorProcessing
  FlyCapture2_C.h, 65
fc2GetDefaultOutputFormat
  FlyCapture2_C.h, 65
fc2GetDeviceFromIndex
  FlyCapture2_C.h, 65
fc2GetEmbeddedImageInfo
  FlyCapture2_C.h, 66
fc2GetFormat7Configuration
  FlyCapture2_C.h, 66
fc2GetFormat7Info
  FlyCapture2_C.h, 66
fc2GetGigEImageBinningSettings
  FlyCapture2_C.h, 67
fc2GetGigEImageSettings
  FlyCapture2_C.h, 67
fc2GetGigEImageSettingsInfo
  FlyCapture2_C.h, 67
fc2GetGigEImagingMode
  FlyCapture2_C.h, 67
fc2GetGigEProperty
  FlyCapture2_C.h, 67
fc2GetGigEStreamChannelInfo
  FlyCapture2_C.h, 67
fc2GetGPIOPinDirection
  FlyCapture2_C.h, 67
fc2GetImageData
  FlyCapture2_C.h, 68
fc2GetImageStatistics
  FlyCapture2_C.h, 68
fc2GetImageTimeStamp
  FlyCapture2_C.h, 68
fc2GetInterfaceTypeFromGuid
  FlyCapture2_C.h, 69
fc2GetLibraryVersion
  FlyCapture2_C.h, 69
fc2GetLUTBankInfo
  FlyCapture2_C.h, 69
fc2GetLUTChannel
  FlyCapture2_C.h, 70
fc2GetLUTInfo
  FlyCapture2_C.h, 70
fc2GetMemoryChannel
  FlyCapture2_C.h, 70
fc2GetMemoryChannelInfo
  FlyCapture2_C.h, 70
fc2GetNumOfCameras
  FlyCapture2_C.h, 71
fc2GetNumOfDevices
  FlyCapture2_C.h, 71
fc2GetNumStreamChannels
  FlyCapture2_C.h, 71
fc2GetProperty
  FlyCapture2_C.h, 71
fc2GetPropertyInfo
  FlyCapture2_C.h, 72
fc2GetRegisterString
  FlyCapture2_C.h, 72
fc2GetStrobe
  FlyCapture2_C.h, 72
fc2GetStrobeInfo
  FlyCapture2_C.h, 72
fc2GetSystemInfo
  FlyCapture2_C.h, 73
fc2GetTriggerDelay

FlyCapture2_C.h, 73
fc2GetTriggerDelayInfo
FlyCapture2_C.h, 73
fc2GetTriggerMode
FlyCapture2_C.h, 73
fc2GetTriggerModeInfo
FlyCapture2_C.h, 74
fc2GetVideoModeAndFrameRate
FlyCapture2_C.h, 74
fc2GetVideoModeAndFrameRateInfo
FlyCapture2_C.h, 74
fc2GigEConfig, 16
channels, 16
numChannels, 16
reserved, 16
fc2GigEImageSettings, 17
height, 17
offsetX, 17
offsetY, 17
pixelFormat, 17
reserved, 17
width, 17
fc2GigEImageSettingsInfo, 18
imageHStepSize, 18
imageVStepSize, 18
maxHeight, 18
maxWidth, 18
offsetHStepSize, 18
offsetVStepSize, 18
pixelFormatBitField, 18
reserved, 18
fc2GigEProperty, 19
isReadable, 19
isWritable, 19
max, 19
min, 19
propType, 19
reserved, 19
value, 19
fc2GigEPropertyType
FlyCapture2Defs_C.h, 104
fc2GigEStreamChannel, 20
destinationIpAddress, 20
doNotFragment, 20
hostPost, 20
interPacketDelay, 20
networkInterfaceIndex, 20
packetSize, 20
reserved, 20
sourcePort, 20
fc2GrabMode
FlyCapture2Defs_C.h, 104
fc2GrabTimeout
FlyCapture2Defs_C.h, 104
fc2GUIConnect
FlyCapture2GUI_C.h, 111
fc2GuiContext
FlyCapture2Defs_C.h, 100
fc2Hide
FlyCapture2GUI_C.h, 111
fc2Image, 21
bayerFormat, 21
cols, 21
dataSize, 21
format, 21
imageImpl, 21
pData, 21
rows, 21
stride, 21
fc2ImageEventCallback
FlyCapture2Defs_C.h, 100
fc2ImageFileFormat
FlyCapture2Defs_C.h, 104
fc2ImageImpl
FlyCapture2Defs_C.h, 101
fc2ImageMetadata, 22
embeddedBrightness, 22
embeddedExposure, 22
embeddedFrameCounter, 22
embeddedGain, 22
embeddedGPIOPinState, 22
embeddedROIPosition, 22
embeddedShutter, 22
embeddedStrobePattern, 22
embeddedTimeStamp, 22
embeddedWhiteBalance, 22
reserved, 22
fc2ImageStatisticsContext
FlyCapture2Defs_C.h, 101
fc2InterfaceType
FlyCapture2Defs_C.h, 105
fc2InternalContext, 23
pBusMgr, 23
pCamera, 23
fc2InternalGuiContext, 24
pCameraControlDlg, 24
pCameraSelectionDlg, 24
fc2InternalImageCallback, 25
pCallback, 25
pCallbackData, 25
fc2IPAddress, 26
octets, 26
fc2IsVisible
FlyCapture2GUI_C.h, 111
fc2JPEGOption, 27
progressive, 27
quality, 27
reserved, 27

fc2JPG2Option, 28
  quality, 28
  reserved, 28
fc2LaunchBrowser
  FlyCapture2_C.h, 75
fc2LaunchCommand
  FlyCapture2_C.h, 75
fc2LaunchCommandAsync
  FlyCapture2_C.h, 75
fc2LaunchHelp
  FlyCapture2_C.h, 75
fc2LUTData, 29
  enabled, 29
  inputBitDepth, 29
  numBanks, 29
  numChannels, 29
  numEntries, 29
  outputBitDepth, 29
  reserved, 29
  supported, 29
fc2MACAddress, 30
  octets, 30
fc2Mode
  FlyCapture2Defs_C.h, 105
fc2OSType
  FlyCapture2Defs_C.h, 106
fc2PGMOption, 31
  binaryFile, 31
  reserved, 31
fc2PGRGuid, 32
  value, 32
fc2PixelFormat
  FlyCapture2Defs_C.h, 106
fc2PNGOption, 33
  compressionLevel, 33
  interlaced, 33
  reserved, 33
fc2PPMOption, 34
  binaryFile, 34
  reserved, 34
fc2PropertyType
  FlyCapture2Defs_C.h, 107
fc2QueryGigEImagingMode
  FlyCapture2_C.h, 76
fc2ReadGVCPMemory
  FlyCapture2_C.h, 76
fc2ReadGVCPRegister
  FlyCapture2_C.h, 76
fc2ReadGVCPRegisterBlock
  FlyCapture2_C.h, 76
fc2ReadRegister
  FlyCapture2_C.h, 77
fc2ReadRegisterBlock
  FlyCapture2_C.h, 77
fc2RegisterCallback
  FlyCapture2_C.h, 77
fc2RescanBus
  FlyCapture2_C.h, 78
fc2RestoreFromMemoryChannel
  FlyCapture2_C.h, 78
fc2RetrieveBuffer
  FlyCapture2_C.h, 78
fc2SaveImage
  FlyCapture2_C.h, 78
fc2SaveImageWithOption
  FlyCapture2_C.h, 79
fc2SaveToMemoryChannel
  FlyCapture2_C.h, 79
fc2SetActiveLUTBank
  FlyCapture2_C.h, 79
fc2SetCallback
  FlyCapture2_C.h, 80
fc2SetConfiguration
  FlyCapture2_C.h, 80
fc2SetDefaultColorProcessing
  FlyCapture2_C.h, 80
fc2SetDefaultOutputFormat
  FlyCapture2_C.h, 80
fc2SetEmbeddedImageInfo
  FlyCapture2_C.h, 81
fc2SetFormat7Configuration
  FlyCapture2_C.h, 81
fc2SetFormat7ConfigurationPacket
  FlyCapture2_C.h, 81
fc2SetGigEImageBinningSettings
  FlyCapture2_C.h, 82
fc2SetGigEImageSettings
  FlyCapture2_C.h, 82
fc2SetGigEImagingMode
  FlyCapture2_C.h, 82
fc2SetGigEProperty
  FlyCapture2_C.h, 82
fc2SetGigEStreamChannelInfo
  FlyCapture2_C.h, 82
fc2SetGPIOPinDirection
  FlyCapture2_C.h, 82
fc2SetGPIOPinDirectionBroadcast
  FlyCapture2_C.h, 82
fc2SetImageData
  FlyCapture2_C.h, 83
fc2SetImageDimensions
  FlyCapture2_C.h, 83
fc2SetLUTChannel
  FlyCapture2_C.h, 83
fc2SetProperty
  FlyCapture2_C.h, 84
fc2SetPropertyBroadcast
  FlyCapture2_C.h, 84

fc2SetStrobe
  FlyCapture2_C.h, 84
fc2SetStrobeBroadcast
  FlyCapture2_C.h, 85
fc2SetTriggerDelay
  FlyCapture2_C.h, 85
fc2SetTriggerDelayBroadcast
  FlyCapture2_C.h, 85
fc2SetTriggerMode
  FlyCapture2_C.h, 86
fc2SetTriggerModeBroadcast
  FlyCapture2_C.h, 86
fc2SetUserBuffers
  FlyCapture2_C.h, 86
fc2SetVideoModeAndFrameRate
  FlyCapture2_C.h, 86
fc2Show
  FlyCapture2GUI_C.h, 112
fc2ShowModal
  FlyCapture2GUI_C.h, 112
fc2StartCapture
  FlyCapture2_C.h, 87
fc2StartCaptureCallback
  FlyCapture2_C.h, 87
fc2StartSyncCapture
  FlyCapture2_C.h, 87
fc2StartSyncCaptureCallback
  FlyCapture2_C.h, 88
fc2StatisticsChannel
  FlyCapture2Defs_C.h, 107
fc2StopCapture
  FlyCapture2_C.h, 88
fc2StrobeControl, 35
  delay, 35
  duration, 35
  onOff, 35
  polarity, 35
  reserved, 35
  source, 35
fc2StrobeInfo, 36
  maxValue, 36
  minValue, 36
  onOffSupported, 36
  polaritySupported, 36
  present, 36
  readOutSupported, 36
  reserved, 36
  source, 36
fc2SystemInfo, 37
  byteOrder, 37
  cpuDescription, 37
  driverList, 37
  gpuDescription, 37
  libraryList, 37
  numCpuCores, 37
  osDescription, 37
  osType, 37
  reserved, 37
  screenHeight, 37
  screenWidth, 37
  sysMemSize, 37
fc2TIFFCompressionMethod
  FlyCapture2Defs_C.h, 108
fc2TIFFOption, 38
  compression, 38
  reserved, 38
fc2TimeStamp, 39
  cycleCount, 39
  cycleOffset, 39
  cycleSeconds, 39
  microSeconds, 39
  reserved, 39
  seconds, 39
fc2TriggerDelay, 40
  absControl, 40
  absValue, 40
  autoManualMode, 40
  onePush, 40
  onOff, 40
  present, 40
  reserved, 40
  type, 40
  valueA, 40
  valueB, 40
fc2TriggerDelayInfo, 41
  absMax, 42
  absMin, 42
  absValSupported, 42
  autoSupported, 42
  manualSupported, 42
  max, 42
  min, 42
  onePushSupported, 42
  onOffSupported, 42
  present, 42
  pUnitAbbr, 42
  pUnits, 42
  readOutSupported, 42
  reserved, 42
  type, 42
fc2TriggerMode, 43
  mode, 43
  onOff, 43
  parameter, 43
  polarity, 43
  reserved, 43
  source, 43
fc2TriggerModeInfo, 44

modeMask, 44
onOffSupported, 44
polaritySupported, 44
present, 44
readOutSupported, 44
reserved, 44
softwareTriggerSupported, 44
sourceMask, 44
valueReadable, 44
fc2UnregisterCallback
FlyCapture2_C.h, 88
fc2ValidateFormat7Settings
FlyCapture2_C.h, 89
fc2Version, 45
build, 45
major, 45
minor, 45
type, 45
fc2VideoMode
FlyCapture2Defs_C.h, 108
fc2WriteGVCPMemory
FlyCapture2_C.h, 89
fc2WriteGVCPRegister
FlyCapture2_C.h, 89
fc2WriteGVCPRegisterBlock
FlyCapture2_C.h, 90
fc2WriteGVCPRegisterBroadcast
FlyCapture2_C.h, 90
fc2WriteRegister
FlyCapture2_C.h, 90
fc2WriteRegisterBlock
FlyCapture2_C.h, 90
fc2WriteRegisterBroadcast
FlyCapture2_C.h, 91
firmwareBuildTime
fc2CameraInfo, 7
firmwareVersion
fc2CameraInfo, 7
FlyCapture2_C.h, 47
fc2AVIAppend, 56
fc2AVIClose, 56
fc2AVIOpen, 57
fc2CalculateImageStatistics, 57
fc2Connect, 57
fc2ConvertImage, 58
fc2ConvertImageTo, 58
fc2CreateAVI, 58
fc2CreateContext, 58
fc2CreateGigEContext, 59
fc2CreateImage, 59
fc2CreateImageStatistics, 59
fc2DestroyAVI, 60
fc2DestroyContext, 60
fc2DestroyImage, 60
fc2DestroyImageStatistics, 60
fc2DetermineBitsPerPixel, 61
fc2Disconnect, 61
fc2DiscoverGigECameras, 61
fc2EnableLUT, 61
fc2ErrorToDescription, 62
fc2FireBusReset, 62
fc2FireSoftwareTrigger, 62
fc2FireSoftwareTriggerBroadcast, 62
fc2ForceIPAddressToCamera, 63
fc2GetActiveLUTBank, 63
fc2GetCameraFromIndex, 63
fc2GetCameraFromSerialNumber, 64
fc2GetCameraInfo, 64
fc2GetCameraSerialNumberFromIndex, 64
fc2GetConfiguration, 65
fc2GetDefaultColorProcessing, 65
fc2GetDefaultOutputFormat, 65
fc2GetDeviceFromIndex, 65
fc2GetEmbeddedImageInfo, 66
fc2GetFormat7Configuration, 66
fc2GetFormat7Info, 66
fc2GetGigEImageBinningSettings, 67
fc2GetGigEImageSettings, 67
fc2GetGigEImageSettingsInfo, 67
fc2GetGigEImagingMode, 67
fc2GetGigEProperty, 67
fc2GetGigEStreamChannelInfo, 67
fc2GetGPIOPinDirection, 67
fc2GetImageData, 68
fc2GetImageStatistics, 68
fc2GetImageTimeStamp, 68
fc2GetInterfaceTypeFromGuid, 69
fc2GetLibraryVersion, 69
fc2GetLUTBankInfo, 69
fc2GetLUTChannel, 70
fc2GetLUTInfo, 70
fc2GetMemoryChannel, 70
fc2GetMemoryChannelInfo, 70
fc2GetNumOfCameras, 71
fc2GetNumOfDevices, 71
fc2GetNumStreamChannels, 71
fc2GetProperty, 71
fc2GetPropertyInfo, 72
fc2GetRegisterString, 72
fc2GetStrobe, 72
fc2GetStrobeInfo, 72
fc2GetSystemInfo, 73
fc2GetTriggerDelay, 73
fc2GetTriggerDelayInfo, 73
fc2GetTriggerMode, 73
fc2GetTriggerModeInfo, 74
fc2GetVideoModeAndFrameRate, 74
fc2GetVideoModeAndFrameRateInfo, 74

fc2LaunchBrowser, 75
fc2LaunchCommand, 75
fc2LaunchCommandAsync, 75
fc2LaunchHelp, 75
fc2QueryGigEImagingMode, 76
fc2ReadGVCPMemory, 76
fc2ReadGVCPRegister, 76
fc2ReadGVCPRegisterBlock, 76
fc2ReadRegister, 77
fc2ReadRegisterBlock, 77
fc2RegisterCallback, 77
fc2RescanBus, 78
fc2RestoreFromMemoryChannel, 78
fc2RetrieveBuffer, 78
fc2SaveImage, 78
fc2SaveImageWithOption, 79
fc2SaveToMemoryChannel, 79
fc2SetActiveLUTBank, 79
fc2SetCallback, 80
fc2SetConfiguration, 80
fc2SetDefaultColorProcessing, 80
fc2SetDefaultOutputFormat, 80
fc2SetEmbeddedImageInfo, 81
fc2SetFormat7Configuration, 81
fc2SetFormat7ConfigurationPacket, 81
fc2SetGigEImageBinningSettings, 82
fc2SetGigEImageSettings, 82
fc2SetGigEImagingMode, 82
fc2SetGigEProperty, 82
fc2SetGigEStreamChannelInfo, 82
fc2SetGPIOPinDirection, 82
fc2SetGPIOPinDirectionBroadcast, 82
fc2SetImageData, 83
fc2SetImageDimensions, 83
fc2SetLUTChannel, 83
fc2SetProperty, 84
fc2SetPropertyBroadcast, 84
fc2SetStrobe, 84
fc2SetStrobeBroadcast, 85
fc2SetTriggerDelay, 85
fc2SetTriggerDelayBroadcast, 85
fc2SetTriggerMode, 86
fc2SetTriggerModeBroadcast, 86
fc2SetUserBuffers, 86
fc2SetVideoModeAndFrameRate, 86
fc2StartCapture, 87
fc2StartCaptureCallback, 87
fc2StartSyncCapture, 87
fc2StartSyncCaptureCallback, 88
fc2StopCapture, 88
fc2UnregisterCallback, 88
fc2ValidateFormat7Settings, 89
fc2WriteGVCPMemory, 89
fc2WriteGVCPRegister, 89
fc2WriteGVCPRegisterBlock, 90
fc2WriteGVCPRegisterBroadcast, 90
fc2WriteRegister, 90
fc2WriteRegisterBlock, 90
fc2WriteRegisterBroadcast, 91
FLYCAPTURE2_C_API
    FlyCapture2Platform_C.h, 114
FLYCAPTURE2_C_CALL_CONVEN
    FlyCapture2Platform_C.h, 114
FlyCapture2Defs_C.h
    FC2_ARRIVAL, 101
    FC2_AUTO_EXPOSURE, 107
    FC2_BANDWIDTH_ALLOCATION_-FORCE_32BITS, 101
    FC2_BANDWIDTH_ALLOCATION_OFF, 101
    FC2_BANDWIDTH_ALLOCATION_ON, 101
    FC2_BANDWIDTH_ALLOCATION_-UNSPECIFIED, 101
    FC2_BANDWIDTH_ALLOCATION_-UNSUPPORTED, 101
    FC2_BMP, 105
    FC2_BRIGHTNESS, 107
    FC2_BT_BGGR, 101
    FC2_BT_FORCE_32BITS, 101
    FC2_BT_GBRG, 101
    FC2_BT_GRBG, 101
    FC2_BT_NONE, 101
    FC2_BT_RGGB, 101
    FC2_BUFFER_FRAMES, 104
    FC2_BUS_RESET, 101
    FC2_BUSSPEED_10000BASE_T, 102
    FC2_BUSSPEED_1000BASE_T, 102
    FC2_BUSSPEED_100BASE_T, 102
    FC2_BUSSPEED_10BASE_T, 102
    FC2_BUSSPEED_ANY, 102
    FC2_BUSSPEED_FORCE_32BITS, 102
    FC2_BUSSPEED_S100, 102
    FC2_BUSSPEED_S1600, 102
    FC2_BUSSPEED_S200, 102
    FC2_BUSSPEED_S3200, 102
    FC2_BUSSPEED_S400, 102
    FC2_BUSSPEED_S480, 102
    FC2_BUSSPEED_S800, 102
    FC2_BUSSPEED_S_FASTEST, 102
    FC2_BUSSPEED_SPEED_UNKNOWN, 102
    FC2_BYTE_ORDER_BIG_ENDIAN, 102
    FC2_BYTE_ORDER_FORCE_32BITS, 102
    FC2_BYTE_ORDER_LITTLE_ENDIAN, 102
    FC2_CALLBACK_TYPE_FORCE_32BITS, 101

FC2_COLOR_PROCESSING_-
ALGORITHM_FORCE_32BITS, 102
FC2_DEFAULT, 102
FC2_DROP_FRAMES, 104
FC2_EDGE_SENSING, 102
FC2_ERROR_BUFFER_TOO_SMALL, 104
FC2_ERROR_BUS_MASTER_FAILED, 103
FC2_ERROR_FAILED, 103
FC2_ERROR_FAILED_BUS_MASTER_-
CONNECTION, 103
FC2_ERROR_FAILED_GUID, 103
FC2_ERROR_FORCE_32BITS, 104
FC2_ERROR_IIDC_FAILED, 103
FC2_ERROR_IMAGE_CONSISTENCY_-
ERROR, 104
FC2_ERROR_IMAGE_CONVERSION_-
FAILED, 103
FC2_ERROR_IMAGE_LIBRARY_-
FAILURE, 104
FC2_ERROR_INIT_FAILED, 103
FC2_ERROR_INVALID_BUS_MANAGER,
103
FC2_ERROR_INVALID_GENERATION,
103
FC2_ERROR_INVALID_MODE, 103
FC2_ERROR_INVALID_PACKET_SIZE,
103
FC2_ERROR_INVALID_PARAMETER, 103
FC2_ERROR_INVALID_SETTINGS, 103
FC2_ERROR_ISOCH_ALREADY_-
STARTED, 103
FC2_ERROR_ISOCH_BANDWIDTH_-
EXCEEDED, 103
FC2_ERROR_ISOCH_FAILED, 103
FC2_ERROR_ISOCH_NOT_STARTED, 103
FC2_ERROR_ISOCH_RETRIEVE_-
BUFFER_FAILED, 103
FC2_ERROR_ISOCH_START_FAILED, 103
FC2_ERROR_ISOCH_STOP_FAILED, 103
FC2_ERROR_ISOCH_SYNC_FAILED, 103
FC2_ERROR_LOW_LEVEL_FAILURE, 103
FC2_ERROR_LUT_FAILED, 103
FC2_ERROR_MEMORY_ALLOCATION_-
FAILED, 103
FC2_ERROR_NOT_CONNECTED, 103
FC2_ERROR_NOT_FOUND, 103
FC2_ERROR_NOT_IMPLEMENTED, 103
FC2_ERROR_NOT_IN_FORMAT7, 103
FC2_ERROR_NOT_INTITIALIZED, 103
FC2_ERROR_NOT_SUPPORTED, 103
FC2_ERROR_OK, 103
FC2_ERROR_PROPERTY_FAILED, 103
FC2_ERROR_PROPERTY_NOT_PRESENT,
103
FC2_ERROR_READ_REGISTER_FAILED,
103
FC2_ERROR_REGISTER_FAILED, 103
FC2_ERROR_STROBE_FAILED, 103
FC2_ERROR_TIMEOUT, 103
FC2_ERROR_TRIGGER_FAILED, 103
FC2_ERROR_UNDEFINED, 103
FC2_ERROR_WRITE_REGISTER_FAILED,
103
FC2_FOCUS, 107
FC2_FRAME_RATE, 107
FC2_FRAMERATE_120, 104
FC2_FRAMERATE_15, 104
FC2_FRAMERATE_1_875, 104
FC2_FRAMERATE_240, 104
FC2_FRAMERATE_30, 104
FC2_FRAMERATE_3_75, 104
FC2_FRAMERATE_60, 104
FC2_FRAMERATE_7_5, 104
FC2_FRAMERATE_FORCE_32BITS, 104
FC2_FRAMERATE_FORMAT7, 104
FC2_FROM_FILE_EXT, 105
FC2_GAIN, 107
FC2_GAMMA, 107
FC2_GRAB_MODE_FORCE_32BITS, 104
FC2_GRAB_TIMEOUT_FORCE_32BITS,
104
FC2_HEARTBEAT, 104
FC2_HEARTBEAT_TIMEOUT, 104
FC2_HQ_LINEAR, 102
FC2_HUE, 107
FC2_IMAGE_FILE_FORMAT_FORCE_-
32BITS, 105
FC2_INTERFACE_GIGE, 105
FC2_INTERFACE_IEEE1394, 105
FC2_INTERFACE_TYPE_FORCE_32BITS,
105
FC2_INTERFACE_UNKNOWN, 105
FC2_INTERFACE_USB_2, 105
FC2_IRIS, 107
FC2_JPEG, 105
FC2_JPEG2000, 105
FC2_LINUX_X64, 106
FC2_LINUX_X86, 106
FC2_MAC, 106
FC2_MODE_0, 105
FC2_MODE_1, 105
FC2_MODE_10, 105
FC2_MODE_11, 105
FC2_MODE_12, 105
FC2_MODE_13, 105
FC2_MODE_14, 105
FC2_MODE_15, 105
FC2_MODE_16, 106

FC2_MODE_17, 106
FC2_MODE_18, 106
FC2_MODE_19, 106
FC2_MODE_2, 105
FC2_MODE_20, 106
FC2_MODE_21, 106
FC2_MODE_22, 106
FC2_MODE_23, 106
FC2_MODE_24, 106
FC2_MODE_25, 106
FC2_MODE_26, 106
FC2_MODE_27, 106
FC2_MODE_28, 106
FC2_MODE_29, 106
FC2_MODE_3, 105
FC2_MODE_30, 106
FC2_MODE_31, 106
FC2_MODE_4, 105
FC2_MODE_5, 105
FC2_MODE_6, 105
FC2_MODE_7, 105
FC2_MODE_8, 105
FC2_MODE_9, 105
FC2_MODE_FORCE_32BITS, 106
FC2_NEAREST_NEIGHBOR_FAST, 102
FC2_NO_COLOR_PROCESSING, 102
FC2_NUM_FRAMERATES, 104
FC2_NUM_MODES, 106
FC2_NUM_PIXEL_FORMATS, 107
FC2_NUM_VIDEOMODES, 109
FC2_OSTYPE_FORCE_32BITS, 106
FC2_PAN, 107
FC2_PGM, 105
FC2_PIXEL_FORMAT_411YUV8, 106
FC2_PIXEL_FORMAT_422YUV8, 106
FC2_PIXEL_FORMAT_444YUV8, 106
FC2_PIXEL_FORMAT_BGR, 107
FC2_PIXEL_FORMAT_BGRU, 107
FC2_PIXEL_FORMAT_MONO12, 107
FC2_PIXEL_FORMAT_MONO16, 106
FC2_PIXEL_FORMAT_MONO8, 106
FC2_PIXEL_FORMAT_RAW12, 107
FC2_PIXEL_FORMAT_RAW16, 107
FC2_PIXEL_FORMAT_RAW8, 107
FC2_PIXEL_FORMAT_RGB, 107
FC2_PIXEL_FORMAT_RGB16, 106
FC2_PIXEL_FORMAT_RGB8, 106
FC2_PIXEL_FORMAT_RGBU, 107
FC2_PIXEL_FORMAT_S_MONO16, 106
FC2_PIXEL_FORMAT_S_RGB16, 107
FC2_PNG, 105
FC2_PPM, 105
FC2_PROPERTY_TYPE_FORCE_32BITS, 107
FC2_RAW, 105
FC2_REMOVAL, 101
FC2_RIGOROUS, 102
FC2_SATURATION, 107
FC2_SHARPNESS, 107
FC2_SHUTTER, 107
FC2_STATISTICS_BLUE, 108
FC2_STATISTICS_FORCE_32BITS, 108
FC2_STATISTICS_GREEN, 108
FC2_STATISTICS_GREY, 108
FC2_STATISTICS_HUE, 108
FC2_STATISTICS_LIGHTNESS, 108
FC2_STATISTICS_RED, 108
FC2_STATISTICS_SATURATION, 108
FC2_TEMPERATURE, 107
FC2_TIFF, 105
FC2_TIFF_ADOBE_DEFLATE, 108
FC2_TIFF_CCITTFAX3, 108
FC2_TIFF_CCITTFAX4, 108
FC2_TIFF_DEFLATE, 108
FC2_TIFF_JPEG, 108
FC2_TIFF_LZW, 108
FC2_TIFF_NONE, 108
FC2_TIFF_PACKBITS, 108
FC2_TILT, 107
FC2_TIMEOUT_INFINITE, 104
FC2_TIMEOUT_NONE, 104
FC2_TIMEOUT_UNSPECIFIED, 104
FC2_TRIGGER_DELAY, 107
FC2_TRIGGER_MODE, 107
FC2_UNKNOWN_OS, 106
FC2_UNSPECIFIED_GRAB_MODE, 104
FC2_UNSPECIFIED_PIXEL_FORMAT, 107
FC2_UNSPECIFIED_PROPERTY_TYPE, 107
FC2_VIDEOMODE_1024x768RGB, 108
FC2_VIDEOMODE_1024x768Y16, 108
FC2_VIDEOMODE_1024x768Y8, 108
FC2_VIDEOMODE_1024x768YUV422, 108
FC2_VIDEOMODE_1280x960RGB, 109
FC2_VIDEOMODE_1280x960Y16, 109
FC2_VIDEOMODE_1280x960Y8, 109
FC2_VIDEOMODE_1280x960YUV422, 109
FC2_VIDEOMODE_1600x1200RGB, 109
FC2_VIDEOMODE_1600x1200Y16, 109
FC2_VIDEOMODE_1600x1200Y8, 109
FC2_VIDEOMODE_1600x1200YUV422, 109
FC2_VIDEOMODE_160x120YUV444, 108
FC2_VIDEOMODE_320x240YUV422, 108
FC2_VIDEOMODE_640x480RGB, 108
FC2_VIDEOMODE_640x480Y16, 108
FC2_VIDEOMODE_640x480Y8, 108
FC2_VIDEOMODE_640x480YUV411, 108

FC2_VIDEOMODE_640x480YUV422, 108
FC2_VIDEOMODE_800x600RGB, 108
FC2_VIDEOMODE_800x600Y16, 108
FC2_VIDEOMODE_800x600Y8, 108
FC2_VIDEOMODE_800x600YUV422, 108
FC2_VIDEOMODE_FORCE_32BITS, 109
FC2_VIDEOMODE_FORMAT7, 109
FC2_WHITE_BALANCE, 107
FC2_WINDOWS_X64, 106
FC2_WINDOWS_X86, 106
FC2_ZOOM, 107
FlyCapture2Defs_C.h, 92
BOOL, 100
FALSE, 100
fc2AsyncCommandCallback, 100
fc2AVIContext, 100
fc2BandwidthAllocation, 101
fc2BayerTileFormat, 101
fc2BusCallbackType, 101
fc2BusEventCallback, 100
fc2BusSpeed, 101
fc2ByteOrder, 102
fc2CallbackHandle, 100
fc2ColorProcessingAlgorithm, 102
fc2Context, 100
fc2Error, 102
fc2FrameRate, 104
fc2GigEPropertyType, 104
fc2GrabMode, 104
fc2GrabTimeout, 104
fc2GuiContext, 100
fc2ImageEventCallback, 100
fc2ImageFileFormat, 104
fc2ImageImpl, 101
fc2ImageStatisticsContext, 101
fc2InterfaceType, 105
fc2Mode, 105
fc2OSType, 106
fc2PixelFormat, 106
fc2PropertyType, 107
fc2StatisticsChannel, 107
fc2TIFFCompressionMethod, 108
fc2VideoMode, 108
FULL_32BIT_VALUE, 100
MAX_STRING_LENGTH, 100
TRUE, 100
FlyCapture2GUI_C.h, 110
fc2CreateGUIContext, 110
fc2DestroyGUIContext, 111
fc2Disonnect, 111
fc2GUIConnect, 111
fc2Hide, 111
fc2IsVisible, 111
fc2Show, 112
fc2ShowModal, 112
FlyCapture2Internal_C.h, 113
IsContextValid, 113
IsGuiContextValid, 113
SyncCppImageToStruct, 113
FlyCapture2Platform_C.h, 114
FLYCAPTURE2_C_API, 114
FLYCAPTURE2_C_CALL_CONVEN, 114
format
fc2Image, 21
frameCounter
fc2EmbeddedImageInfo, 11
frameRate
fc2AVIOption, 5
FULL_32BIT_VALUE
FlyCapture2Defs_C.h, 100

gain
fc2EmbeddedImageInfo, 11
gigEMajorVersion
fc2CameraInfo, 7
gigEMinorVersion
fc2CameraInfo, 7
GPIOPinState
fc2EmbeddedImageInfo, 11
gpuDescription
fc2SystemInfo, 37
grabMode
fc2Config, 8
grabTimeout
fc2Config, 8

height
fc2Format7ImageSettings, 13
fc2GigEImageSettings, 17
hostPost
fc2GigEStreamChannel, 20

iidcVer
fc2CameraInfo, 7
imageHStepSize
fc2Format7Info, 14
fc2GigEImageSettingsInfo, 18
imageImpl
fc2Image, 21
imageVStepSize
fc2Format7Info, 14
fc2GigEImageSettingsInfo, 18
inputBitDepth
fc2LUTData, 29
interfaceType
fc2CameraInfo, 7
interlaced
fc2PNGOption, 33

interPacketDelay
    fc2GigEStreamChannel, 20
ipAddress
    fc2CameraInfo, 7
isColorCamera
    fc2CameraInfo, 7
IsContextValid
    FlyCapture2Internal_C.h, 113
IsGuiContextValid
    FlyCapture2Internal_C.h, 113
isochBusSpeed
    fc2Config, 8
isReadable
    fc2GigEProperty, 19
isWritable
    fc2GigEProperty, 19

libraryList
    fc2SystemInfo, 37

macAddress
    fc2CameraInfo, 7
major
    fc2Version, 45
manualSupported
    fc2TriggerDelayInfo, 42
max
    fc2GigEProperty, 19
    fc2TriggerDelayInfo, 42
MAX_STRING_LENGTH
    FlyCapture2Defs_C.h, 100
maxBytesPerPacket
    fc2Format7PacketInfo, 15
maxHeight
    fc2Format7Info, 14
    fc2GigEImageSettingsInfo, 18
maximumBusSpeed
    fc2CameraInfo, 7
maxPacketSize
    fc2Format7Info, 14
maxValue
    fc2StrobeInfo, 36
maxWidth
    fc2Format7Info, 14
    fc2GigEImageSettingsInfo, 18
microSeconds
    fc2TimeStamp, 39
min
    fc2GigEProperty, 19
    fc2TriggerDelayInfo, 42
minor
    fc2Version, 45
minPacketSize
    fc2Format7Info, 14
minValue
    fc2StrobeInfo, 36
mode
    fc2Format7ImageSettings, 13
    fc2Format7Info, 14
    fc2TriggerMode, 43
modelName
    fc2CameraInfo, 7
modeMask
    fc2TriggerModeInfo, 44

networkInterfaceIndex
    fc2GigEStreamChannel, 20
nodeVendorId
    fc2ConfigROM, 9
numBanks
    fc2LUTData, 29
numBuffers
    fc2Config, 8
numChannels
    fc2GigEConfig, 16
    fc2LUTData, 29
numCpuCores
    fc2SystemInfo, 37
numEntries
    fc2LUTData, 29
numImageNotifications
    fc2Config, 8

octets
    fc2IPAddress, 26
    fc2MACAddress, 30
offsetHStepSize
    fc2Format7Info, 14
    fc2GigEImageSettingsInfo, 18
offsetVStepSize
    fc2Format7Info, 14
    fc2GigEImageSettingsInfo, 18
offsetX
    fc2Format7ImageSettings, 13
    fc2GigEImageSettings, 17
offsetY
    fc2Format7ImageSettings, 13
    fc2GigEImageSettings, 17
onePush
    fc2TriggerDelay, 40
onePushSupported
    fc2TriggerDelayInfo, 42
onOff
    fc2EmbeddedImageInfoProperty, 12
    fc2StrobeControl, 35
    fc2TriggerDelay, 40
    fc2TriggerMode, 43
onOffSupported

fc2StrobeInfo, 36
fc2TriggerDelayInfo, 42
fc2TriggerModeInfo, 44
osDescription
fc2SystemInfo, 37
osType
fc2SystemInfo, 37
outputBitDepth
fc2LUTData, 29

packetSize
fc2Format7Info, 14
fc2GigEStreamChannel, 20
parameter
fc2TriggerMode, 43
pBusMgr
fc2InternalContext, 23
pCallback
fc2InternalImageCallback, 25
pCallbackData
fc2InternalImageCallback, 25
pCamera
fc2InternalContext, 23
pCameraControlDlg
fc2InternalGuiContext, 24
pCameraSelectionDlg
fc2InternalGuiContext, 24
pData
fc2Image, 21
percentage
fc2Format7Info, 14
pixelFormat
fc2Format7ImageSettings, 13
fc2GigEImageSettings, 17
pixelFormatBitField
fc2Format7Info, 14
fc2GigEImageSettingsInfo, 18
polarity
fc2StrobeControl, 35
fc2TriggerMode, 43
polaritySupported
fc2StrobeInfo, 36
fc2TriggerModeInfo, 44
present
fc2StrobeInfo, 36
fc2TriggerDelay, 40
fc2TriggerDelayInfo, 42
fc2TriggerModeInfo, 44
progressive
fc2JPEGOption, 27
propType
fc2GigEProperty, 19
pszKeyword
fc2ConfigROM, 9
pUnitAbbr
fc2TriggerDelayInfo, 42
pUnits
fc2TriggerDelayInfo, 42

quality
fc2JPEGOption, 27
fc2JPG2Option, 28

readOutSupported
fc2StrobeInfo, 36
fc2TriggerDelayInfo, 42
fc2TriggerModeInfo, 44
recommendedBytesPerPacket
fc2Format7PacketInfo, 15
reserved
fc2AVIOption, 5
fc2CameraInfo, 7
fc2Config, 8
fc2ConfigROM, 9
fc2Format7ImageSettings, 13
fc2Format7Info, 14
fc2Format7PacketInfo, 15
fc2GigEConfig, 16
fc2GigEImageSettings, 17
fc2GigEImageSettingsInfo, 18
fc2GigEProperty, 19
fc2GigEStreamChannel, 20
fc2ImageMetadata, 22
fc2JPEGOption, 27
fc2JPG2Option, 28
fc2LUTData, 29
fc2PGMOption, 31
fc2PNGOption, 33
fc2PPMOption, 34
fc2StrobeControl, 35
fc2StrobeInfo, 36
fc2SystemInfo, 37
fc2TIFFOption, 38
fc2TimeStamp, 39
fc2TriggerDelay, 40
fc2TriggerDelayInfo, 42
fc2TriggerMode, 43
fc2TriggerModeInfo, 44
ROIPosition
fc2EmbeddedImageInfo, 11
rows
fc2Image, 21

screenHeight
fc2SystemInfo, 37
screenWidth
fc2SystemInfo, 37
seconds

fc2TimeStamp, 39
sensorInfo
fc2CameraInfo, 7
sensorResolution
fc2CameraInfo, 7
serialNumber
fc2CameraInfo, 7
shutter
fc2EmbeddedImageInfo, 11
softwareTriggerSupported
fc2TriggerModeInfo, 44
source
fc2StrobeControl, 35
fc2StrobeInfo, 36
fc2TriggerMode, 43
sourceMask
fc2TriggerModeInfo, 44
sourcePort
fc2GigEStreamChannel, 20
stride
fc2Image, 21
strobePattern
fc2EmbeddedImageInfo, 11
subnetMask
fc2CameraInfo, 7
supported
fc2LUTData, 29
SyncCppImageToStruct
FlyCapture2Internal_C.h, 113
sysMemSize
fc2SystemInfo, 37

timestamp
fc2EmbeddedImageInfo, 11
TRUE
FlyCapture2Defs_C.h, 100
type
fc2TriggerDelay, 40
fc2TriggerDelayInfo, 42
fc2Version, 45

unitBytesPerPacket
fc2Format7PacketInfo, 15
unitSpecId
fc2ConfigROM, 9
unitSubSWVer
fc2ConfigROM, 9
unitSWVer
fc2ConfigROM, 9
userDefinedName
fc2CameraInfo, 7

value
fc2GigEProperty, 19
fc2PGRGuid, 32
valueA
fc2TriggerDelay, 40
valueB
fc2TriggerDelay, 40
valueReadable
fc2TriggerModeInfo, 44
vendorName
fc2CameraInfo, 7
vendorUniqueInfo_0
fc2ConfigROM, 9
vendorUniqueInfo_1
fc2ConfigROM, 9
vendorUniqueInfo_2
fc2ConfigROM, 9
vendorUniqueInfo_3
fc2ConfigROM, 9

whiteBalance
fc2EmbeddedImageInfo, 11
width
fc2Format7ImageSettings, 13
fc2GigEImageSettings, 17

xmlURL1
fc2CameraInfo, 7
xmlURL2
fc2CameraInfo, 7