Contents

1

LIBFLIAPI **FLICancelExposure** (flidev_t dev)

Cancel an exposure for a given camera.

Cancel an exposure for a given camera. This function cancels an exposure in progress by closing the shutter.

See Also:
FLIExposeFrame
FLIEndExposure
FLIGetExposureStatus
FLISetExposureTime

2

LIBFLIAPI **FLIEndExposure** (flidev_t dev)

End an exposure for a given camera.

End an exposure for a given camera. This function causes the exposure to end and image download begins immediately.

See Also: FLIExposeFrame
FLICancelExposure
FLIGetExposureStatus
FLISetExposureTime

3

LIBFLIAPI **FLITriggerExposure** (flidev_t dev)

Trigger an exposure that is awaiting an external trigger.

Trigger an exposure that is awaiting an external trigger. This is a software override for the external trigger option.

See Also:

FLIExposeFrame
FLICancelExposure
FLIEndExposure
FLIGetExposureStatus
FLISetExposureTime

4

LIBFLIAPI **FLIClose** (flidev_t dev)

Close a handle to a FLI device

Close a handle to a FLI device

See Also: FLIOpen

5

LIBFLIAPI **FLIGetArrayArea** (flidev_t dev, long* ul_x, long* ul_y, long* lr_x, long* lr_y)

Get the array area of the given camera.

`ul_x` Pointer to where the upper-left x-coordinate is to be placed.

`ul_y` Pointer to where the upper-left y-coordinate is to be placed.

`lr_x` Pointer to where the lower-right x-coordinate is to be placed.

`lr_y` Pointer to where the lower-right y-coordinate is to be placed.

Get the array area of the given camera. This function finds the *total* area of the CCD array for camera `dev`. This area is specified in terms of a upper-left point and a lower-right point. The upper-left x-coordinate is placed in `ul_x`, the upper-left y-coordinate is placed in `ul_y`, the lower-right x-coordinate is placed in `lr_x`, and the lower-right y-coordinate is placed in `lr_y`.

See Also: FLIGetVisibleArea
FLISetImageArea

6

LIBFLIAPI **FLIFlushRow** (flidev_t dev, long rows, long repeat)

Flush rows of a given camera.

`rows` Number of rows to flush.

`repeat` Number of times to flush each row.

Flush rows of a given camera. This function flushes `rows` rows of camera `dev` `repeat` times.

See Also: FLISetNFlushes

7

LIBFLIAPI **FLIGetFWRevision** (flidev_t dev, long* fwrev)

Get firmware revision of a given device

`fwrev` Pointer to a long which will receive the firmware revision.

Get firmware revision of a given device

See Also: FLIGetModel
FLIGetHWRevision
FLIGetSerialNum

8

LIBFLIAPI **FLIGetHWRevision** (flidev_t dev, long* hwrev)

Get the hardware revision of a given device

`hwrev` Pointer to a long which will receive the hardware revision.

Get the hardware revision of a given device

See Also: FLIGetModel
FLIGetFWRevision
FLIGetSerialNum

9

LIBFLIAPI **FLIGetLibVersion** (char* ver, size_t len)

Get the current library version.

`len` The size in bytes of the buffer pointed to by `ver`.

Get the current library version. This function copies up to `len - 1` characters of the current library version string followed by a terminating `NULL` character into the buffer pointed to by `ver`.

Return Value:	`Zero`	on success.
	`Non-zero`	on failure.

10

LIBFLIAPI **FLIGetSerialString** (flidev_t dev, char* serial, size_t len)

Get the serial string of a given device.

`serial` Pointer to a character buffer where the serial string isto be placed..

`len` The size in bytes of buffer pointed to by `serial`.

Get the serial string of a given device. This function copies up to `len - 1` characters of the serial string for device `dev`, followed by a terminating `NULL` character into the buffer pointed to by `serial`.

See Also: FLIGetHWRevision
FLIGetFWRevision
FLIGetModel

11

LIBFLIAPI **FLIGetModel** (flidev_t dev, char* model, size_t len)

Get the model of a given device.

`model` Pointer to a character buffer where the model string is to be placed.

`len` The size in bytes of buffer pointed to by `model`.

Get the model of a given device. This function copies up to `len - 1` characters of the model string for device `dev`, followed by a terminating `NULL` character into the buffer pointed to by `model`.

See Also: FLIGetHWRevision
FLIGetFWRevision
FLIGetSerialNum

12

LIBFLIAPI **FLIGetPixelSize** (flidev_t dev, double* pixel_x, double* pixel_y)

Find the dimensions of a pixel in the array of the given device

`pixel_x` Pointer to a double which will receive the size (in micons) of a piixel in the x direction.

`pixel_y` Pointer to a double which will receive the size (in micons) of a piixel in the y direction.

Find the dimensions of a pixel in the array of the given device

See Also:	FLIGetArrayArea FLIGetVisibleArea

13

LIBFLIAPI **FLIGetVisibleArea** (flidev_t dev, long* ul_x, long* ul_y, long* lr_x, long* lr_y)

Get the visible area of the given camera.

`ul_x` Pointer to where the upper-left x-coordinate is to be placed.

`ul_y` Pointer to where the upper-left y-coordinate is to be placed.

`lr_x` Pointer to where the lower-right x-coordinate is to be placed.

`lr_y` Pointer to where the lower-right y-coordinate is to be placed.

Get the visible area of the given camera. This function finds the *visible* area of the CCD array for the camera `dev`. This area is specified in terms of a upper-left point and a lower-right point. The upper-left x-coordinate is placed in `ul_x`, the upper-left y-coordinate is placed in `ul_y`, the lower-right x-coordinate is placed in `lr_x`, the lower-right y-coordinate is placed in `lr_y`.

See Also: FLIGetArrayArea
FLISetImageArea

14

LIBFLIAPI **FLIOpen** (flidev_t* dev, char* name, flidomain_t domain)

Get a handle to an FLI device.

`name` Pointer to a string where the device filename to be openedis storedd. For parallel port devices that are not probed by FLIList()) (Windows 95/98/Me), place the address of the prllel port in a string in ascii form ie: ”0x3778”.

`domain` Domain to apply to `name` for device opening. Thi is a bitwisse ORed combination of interface method and device tpe. Valid intterfaces include texttt FLIDOMAIIN_PARALLEL_PORT, `FLIDOMAIN_USB`, FLIDOMAIN_SERIIAL, and `LIDOMAIIN_INET`. Valid device ypes include FLIDEVICCEE_CAMERA, `FLIDOMAIN_FILTEWHEEL`, and FLIDOMAAIN_FOCUSER.

Get a handle to an FLI device. This function requires the filename and domain of the requested device. Valid device filenames can be obtained using the `FLIList()` function. An application may use any number of handles associated with the same physical device. When doing so, it is important to lock the appropriate device to ensure that multiple accesses to the same device do not occur during critical operations.

See Also: FLIList
FLIClose
flidomain_t

15

LIBFLIAPI **FLISetDebugLevel** (char* host, flidebug_t level)

Enable debugging of API operations and communications.

`level` Debug level. A value of `FLIDEBUG_NONE` disables debugging. Values of `FLIDEBUG_FAIL`, `FLIDEBUG_WARN`, and __INFO enable progressively more verbbose debug messages.

Enable debugging of API operations and communications. Use this function in combination with FLIDebug to assist in diagnosing problems that may be encountered during programming.

When usings Microsoft Windows operating systems, creating an empty file C: FLIDBG.TXT will override this option. All debug output will then be directed to this file.

Return Value:	`Zero`	on success.
	`Non-zero`	on failure.

16

LIBFLIAPI **FLISetExposureTime** (flidev_t dev, long exptime)

Set the exposure time for a camera.

`exptime` Exposure time in msec.

Set the exposure time for a camera. This function sets the exposure time for the camera `dev` to `exptime` msec.

See Also: FLIExposeFrame
FLICancelExposure
FLIGetExposureStatus

17

LIBFLIAPI **FLISetHBin** (flidev_t dev, long hbin)

Set the horizontal bin factor for a given camera.

`hbin` Horizontal bin factor.

Set the horizontal bin factor for a given camera. This function sets the horizontal bin factor for the camera `dev` to `hbin`. The valid range of the `hbin` parameter is from 1 to 16.

See Also: FLISetVBin
FLISetImageArea

18

LIBFLIAPI **FLISetFrameType** (flidev_t dev, fliframe_t frametype)

Set the frame type for a given camera.

`frametype` Frame type: `FLI_FRAME_TYPE_NORMAL` or `FLI_FRAME_TYPE_DARK`.

Set the frame type for a given camera. This function sets the frame type for camera `dev` to `frametype`. The `frametype` parameter is either `FLI_FRAME_TYPE_NORMAL` for a normal frame where the shutter opens or `FLI_FRAME_TYPE_DARK` for a dark frame where the shutter remains closed.

See Also: fliframe_t
FLIExposeFrame

19

LIBFLIAPI **FLIGetCoolerPower** (flidev_t dev, double* power)

Get the cooler power level.

`timeleft` Pointer to where the cooler power (in percent) will be placed.

Get the cooler power level. The function places the current cooler power in percent in the location pointed to by `power`.

See Also: FLISetTemperature
FLIGetTemperature

20

LIBFLIAPI **FLISetImageArea** (flidev_t dev, long ul_x, long ul_y, long lr_x, long lr_y)

Set the image area for a given camera.

`ul_x` Upper-left x-coordinate of image area.

`ul_y` Upper-left y-coordinate of image area.

`lr_x` Lower-right x-coordinate of image area (lr'_x from above).

`lr_y` Lower-right y-coordinate of image area (lr'_y from above).

Set the image area for a given camera. This function sets the image area for camera `dev` to an area specified in terms of a upper-left point and a lower-right point. The upper-left x-coordinate is `ul_x`, the upper-left y-coordinate is `ul_y`, the lower-right x-coordinate is `lr_x`, and the lower-right y-coordinate is `lr_y`. Note that the given lower-right coordinate must take into account the horizontal and vertical bin factor settings, but the upper-left coordinate is absolute. In other words, the lower-right coordinate used to set the image area is a virtual point (lr'_x, lr'_y) determined by:

$$lr'_x = ul_x + (lr_x - ul_x)/hbin$$

$$lr'_y = ul_y + (lr_y - ul_y)/vbin$$

Where (lr'_x, lr'_y) is the coordinate to pass to the `FLISetImageArea` function, (ul_x, ul_y) and (lr_x, lr_y) are the absolute coordinates of the desired image area, *hbin* is the horizontal bin factor, and *vbin* is the vertical bin factor.

See Also: FLIGetVisibleArea
FLIGetArrayArea

21

LIBFLIAPI **FLISetVBin** (flidev_t dev, long vbin)

Set the vertical bin factor for a given camera.

`vbin` Vertical bin factor.

Set the vertical bin factor for a given camera. This function sets the vertical bin factor for the camera `dev` to `vbin`. The valid range of the `vbin` parameter is from 1 to 16.

See Also: FLISetHBin
FLISetImageArea

22

LIBFLIAPI **FLIGetExposureStatus** (flidev_t dev, long* timeleft)

Find the remaining exposure time of a given camera.

`timeleft` Pointer to where the remaining exposure time (in milliseonds) will be placed.

Find the remaining exposure time of a given camera. This functions places the remaining exposure time (in milliseconds) in the location pointed to by `timeleft`.

See Also:	FLIExposeFrame FLICancelExposure FLISetExposureTime

23

LIBFLIAPI **FLISetTemperature** (flidev_t dev, double temperature)

Set the temperature of a given camera.

`temperature` Temperature in Celsius to set CCD camera cold finger to.

Set the temperature of a given camera. This function sets the temperature of the CCD camera `dev` to `temperature` degrees Celsius. The valid range of the `temperature` parameter is from -55 C to 45 C.

See Also: FLIGetTemperature

24

LIBFLIAPI **FLIGetTemperature** (flidev_t dev, double* temperature)

Get the temperature of a given camera.

`temperature` Pointer to where the temperature will be placed.

Get the temperature of a given camera. This function places the temperature of the CCD camera cold finger of device `dev` in the location pointed to by `temperature`.

See Also: FLISetTemperature

25

LIBFLIAPI **FLIGrabRow** (flidev_t dev, void* buff, size_t width)

Grab a row of an image.

`buff` Pointer to where the next available row will be placed.

`width` Row width in pixels.

Grab a row of an image. This function grabs the next available row of the image from camera device `dev`. The row of width `width` is placed in the buffer pointed to by `buff`. The size of the buffer pointed to by `buff` must take into account the bit depth of the image, meaning the buffer size must be at least `width` bytes for an 8-bit image, and at least 2*`width` for a 16-bit image.

See Also: FLIGrabFrame

26

LIBFLIAPI **FLIExposeFrame** (flidev_t dev)

Expose a frame for a given camera.

Expose a frame for a given camera. This function exposes a frame according to the settings (image area, exposure time, bit depth, etc.) of camera `dev`. The settings of `dev` must be valid for the camera device. They are set by calling the appropriate set library functions. This function returns after the exposure has started.

See Also:

FLISetExposureTime
FLISetFrameType
FLISetImageArea
FLISetHBin
FLISetVBin
FLISetNFlushes
FLISetBitDepth
FLIGrabFrame
FLICancelExposure
FLIGetExposureStatus

27

LIBFLIAPI **FLISetBitDepth** (flidev_t dev, flibitdepth_t bitdepth)

Set the gray-scale bit depth for a given camera.

`bitdepth` Gray-scale bit depth: `FLI_MODE_8BIT` or FLI_MODEE_16BIT.

Set the gray-scale bit depth for a given camera. This function sets the gray-scale bit depth of camera `dev` to `bitdepth`. The `bitdepth` parameter is either `FLI_MODE_8BIT` for 8-bit mode or `FLI_MODE_16BIT` for 16-bit mode. Many cameras do not support this mode.

See Also:	flibitdepth_t FLIExposeFrame

28

LIBFLIAPI **FLISetNFlushes** (flidev_t dev, long nflushes)

Set the number of flushes for a given camera.

`nflushes` Number of times to flush CCD array before an exposure.

Set the number of flushes for a given camera. This function sets the number of times the CCD array of camera `dev` is flushed by the FLIExposeFrame *before* exposing a frame to `nflushes`. The valid range of the `nflushes` parameter is from 0 to 16. Some FLI cameras support background flushing. Background flushing continuously flushes the CCD eliminating the need for pre-exposure flushing.

See Also: FLIFlushRow
FLIExposeFrame
FLIControlBackgroundFlush

29

LIBFLIAPI **FLIReadIOPort** (flidev_t dev, long* ioportset)

Read the I/O port of a given camera.

`ioportset` Pointer to where the I/O port data will be stored.

Read the I/O port of a given camera. This function reads the I/O port on camera `dev` and places the value in the location pointed to by `ioportset`.

See Also: FLIWriteIOPort
FLIConfigureIOPort

30

LIBFLIAPI **FLIWriteIOPort** (flidev_t dev, long ioportset)

Write to the I/O port of a given camera.

`ioportset` Data to be written to the I/O port.

Write to the I/O port of a given camera. This function writes the value `ioportset` to the I/O port on camera `dev`.

See Also: FLIReadIOPort
FLIConfigureIOPort

31

LIBFLIAPI **FLIConfigureIOPort** (flidev_t dev, long ioportset)

Configure the I/O port of a given camera.

`ioportset` Data to configure the I/O port with.

Configure the I/O port of a given camera. This function configures the I/O port on camera `dev` with the value `ioportset`.

The I/O configuration of each pin on a given camera is determined by the value of `ioportset`. Setting a respective I/O bit enables the port bit for output while clearing an I/O bit enables to port bit for input. By default, all I/O ports are configured as inputs.

See Also: FLIReadIOPort
FLIWriteIOPort

32

LIBFLIAPI **FLILockDevice** (flidev_t dev)

Lock a specified device.

Lock a specified device. This function establishes an exclusive lock (mutex) on the given device to prevent access to the device by any other function or process.

See Also: FLIUnlockDevice

33

LIBFLIAPI **FLIUnlockDevice** (flidev_t dev)

Unlock a specified device.

Unlock a specified device. This function releases a previously established exclusive lock (mutex) on the given device to allow access to the device by any other function or process.

See Also: FLILockDevice

34

LIBFLIAPI **FLIControlShutter** (flidev_t dev, flishutter_t shutter)

Control the shutter on a given camera.

`shutter` How to control the shutter. A value of `FLI_SHUTTER_CLOSE` closes the shutter and `FLI_SUTTER_OPEN` opns the shutter. FLI_SHUTTTER_EXTERNAL_TTRIGGER_LOW, FLI_SHUTTER_EXTERNAL_TRIGGERR causes the exposure to begin ony when a logiic LOW is detected on I/O port bit 0. `FLISUTEER_EXTERNAL_TRIGGER_HIGH` causes the exposuure to begin onlywen a logicc HIGH is detected on I/O port bit 00. This setting ma not be availlable on all cameras.

Control the shutter on a given camera. This function controls the shutter function on camera `dev` according to the `shutter` parameter.

See Also: flishutter_t

35

LIBFLIAPI **FLIControlBackgroundFlush** (flidev_t dev, flibgflush_t bgflush)

Enables background flushing of CCD array.

`bgflush` Enables or disables background flushing. A value of FLI_BGFLLUSH_START begins background flushing. It is important to note tha ackgroound flushing is stoppedd whenever `FLIExposeFrame(` or `FLIoontrolShutter()` are called. `FLI_BGFLUSH_STOP` stops all background flush activity.

Enables background flushing of CCD array. This function enables the background flushing of the CCD array camera `dev` according to the `bgflush` parameter. Note that this function may not succeed on all FLI products as this feature may not be available.

See Also: flibgflush_t

36

LIBFLIAPI **FLIList** (flidomain_t domain, char*** names)

List available devices.

`names` Pointer to where the device name list will be placed.

List available devices. This function returns a pointer to a NULL terminated list of device names. The pointer should be freed later with `FLIFreeList()`. Each device name in the returned list includes the filename needed by `FLIOpen()`, a separating semicolon, followed by the model name or user assigned device name.

See Also: flidomain_t
FLIFreeList
FLIOpen

37

LIBFLIAPI **FLIFreeList** (char** names)

Free a previously generated device list.

Free a previously generated device list. Use this function after `FLIList()` to free the list of device names.

See Also: FLIList

38

LIBFLIAPI **FLISetFilterPos** (flidev_t dev, long filter)

Set the filter wheel position of a given device.

`filter` Desired filter wheel position.

Set the filter wheel position of a given device. Use this function to set the filter wheel position of `dev` to `filter`.

See Also: FLIGetFilterPos

39

LIBFLIAPI **FLIGetFilterPos** (flidev_t dev, long* filter)

Get the filter wheel position of a given device.

`filter` Pointer to where the filter wheel position will be placed.

Get the filter wheel position of a given device. Use this function to get the filter wheel position of `dev`.

See Also: FLISetFilterPos

40

LIBFLIAPI **FLIGetStepsRemaining** (flidev_t dev, long* steps)

Get the number of motor steps remaining.

`filter` Pointer to where the number of remaning steps will be placed.

Get the number of motor steps remaining. Use this function to determine if the stepper motor of `dev` is still moving.

See Also: FLISetFilterPos

41

LIBFLIAPI **FLIGetFilterCount** (flidev_t dev, long* filter)

Get the filter wheel filter count of a given device.

`filter` Pointer to where the filter wheel filter count will be placed.

Get the filter wheel filter count of a given device. Use this function to get the filter count of filter wheel `dev`.

Return Value:	`Zero`	on success.
	`Non-zero`	on failure.

42

LIBFLIAPI **FLIStepMotorAsync** (flidev_t dev, long steps)

Step the filter wheel or focuser motor of a given device.

`steps` Number of steps to move the focuser or filter wheel.

Step the filter wheel or focuser motor of a given device. Use this function to move the focuser or filter wheel `dev` by an amount `steps`. This function is non-blocking.

See Also: FLIGetStepperPosition

43

LIBFLIAPI **FLIStepMotor** (flidev_t dev, long steps)

Step the filter wheel or focuser motor of a given device.

`steps` Number of steps to move the focuser or filter wheel.

Step the filter wheel or focuser motor of a given device. Use this function to move the focuser or filter wheel `dev` by an amount `steps`.

See Also: FLIGetStepperPosition

44

LIBFLIAPI **FLIGetStepperPosition** (flidev_t dev, long* position)

Get the stepper motor position of a given device.

`position` Pointer to where the postion of the stepper motor will be placed.

Get the stepper motor position of a given device. Use this function to read the stepper motor position of filter wheel or focuser `dev`.

See Also: FLIStepMotor

45

LIBFLIAPI **FLIHomeDevice** (flidev_t dev)

Home focuser or filter wheel specified by `dev` *The home position of a device is defined as where the electromechanical home sensor detects home.*

Home focuser or filter wheel specified by `dev` The home position of a device is defined as where the electromechanical home sensor detects home. Note that on color filter wheels this may not be located at filter slot zero and may in fact be between filter slots. It should be noted that this function replaces the deprecated function FLIHomeFocuser(). This function may not return immediately as older FLI devices blocked during a HOME operation. Use the function FLIGetDeviceStatus() to determine if the filter wheel or focuser is still moving (or is capable of reporting device status).

See Also: FLIGetDeviceStatus

46

LIBFLIAPI **FLIHomeFocuser** (flidev_t dev)

Home focuser `dev`.

Home focuser `dev`. The home position is closed as far as mechanically possiable.

Return Value:

`Zero`	on success.
`Non-zero`	on failure.

47

LIBFLIAPI **FLIGetFocuserExtent** (flidev_t dev, long* extent)

Retreive the maximum extent for FLI focuser `dev`

Retreive the maximum extent for FLI focuser `dev`

Return Value:	`Zero`	on success.
	`Non-zero`	on failure.

48

LIBFLIAPI **FLIReadTemperature** (flidev_t dev, flichannel_t channel, double* temperature)

Retreive temperature from the FLI focuser `dev`.

Retreive temperature from the FLI focuser `dev`. Valid channels are `FLI_TEMPERATURE_INTERNAL` and `FLI_TEMPERATURE_EXTERNAL`.

Return Value:	`Zero`	on success.
	`Non-zero`	on failure.

49

LIBFLIAPI **FLICreateList** (flidomain_t domain)

Creates a list of all devices within a specified `domain`.

Creates a list of all devices within a specified `domain`. Use `FLIDeleteList()` to delete the list created with this function. This function is the first called begin the iteration through the list of current FLI devices attached.

See Also: FLIDeleteList
FLIListFirst
FLIListNext

50

LIBFLIAPI **FLIDeleteList** (void)

Deletes a list of devices created by `FLICreateList()`

Deletes a list of devices created by `FLICreateList()`

See Also: FLICreateList
FLIListFirst
FLIListNext

51

LIBFLIAPI **FLIListFirst** (flidomain_t* domain, char* filename, size_t fnlen, char* name, size_t namelen)

Obtains the first device in the list.

`filename` Pointer to where the filename of the device will be placed.

`fnlen` Length of the supplied buffer to hold the filename.

`name` Pointer to where the name of the device will be placed.

`namelen` Length of the supplied buffer to hold the name.

Obtains the first device in the list. Use this function to get the first `domain`, `filename` and `name` from the list of attached FLI devices created using the function `FLICreateList()`. Use `FLIListNext()` to obtain more found devices.

See Also: FLICreateList
FLIDeleteList
FLIListNext

52

LIBFLIAPI **FLIListNext** (flidomain_t* domain, char* filename, size_t fnlen, char* name, size_t namelen)

Obtains the next device in the list.

`filename` Pointer to where the filename of the device will be placed.

`fnlen` Length of the supplied buffer to hold the filename.

`name` Pointer to where the name of the device will be placed.

`namelen` Length of the supplied buffer to hold the name.

Obtains the next device in the list. Use this function to get the next `domain`, `filename` and `name` from the list of attached FLI devices created using the function `FLICreateList()`.

See Also: FLICreateList
FLIDeleteList
FLIListFirst