

FLI Software Development Library

— Version 1.40 —

*Windows and Linux support for FLI CCD cameras, filter
wheels, and focusers.*

Finger Lakes Instrumentation
Copyright (c) 2000-2002 Finger Lakes Instrumentation (FLI), LLC.
All rights reserved.

Contents

1	Introduction	4
2	Library Defined Types	5
3	Library Functions	9

Introduction

This library provides a core set of functions for programming FLI CCD cameras, filter wheels, and focusers under Windows and Linux. The type definitions, function prototypes, and definitions/enumerations of constant values used by library functions are specified in `libfli.h`. All library functions return zero on successful completion, and non-zero if an error occurred. The exact nature of an error can be found by treating the negative of a function's return value as a system error code, for example:

```
if ((err = FLIOpen(&dev, name, domain))){
    fprintf(stderr, "Error FLIOpen: %s\n", strerror(-(int)err));
    exit(1);
}
```

2 Library Defined Types

Names

2.1	#define	FLI_INVALID_DEVICE (-1)	<i>An opaque handle used by library functions to refer to FLI hardware .</i>	5
2.2	typedef long	flidomain_t	<i>The domain of an FLI device.</i>	6
2.3	typedef long	fliframe_t	<i>The frame type for an FLI CCD camera device.</i>	6
2.4	typedef long	flibitdepth_t	<i>The gray-scale bit depth for an FLI camera device.</i>	6
2.5	typedef long	flishutter_t	<i>Type used for shutter operations for an FLI camera device.</i>	7
2.6	typedef long	flibgflush_t	<i>Type used for background flush operations for an FLI camera device. ..</i>	7
2.7	typedef long	flichannel_t	<i>Type used to determine which temperature channel to read.</i>	8
2.8	typedef long	flidebug_t	<i>Type specifying library debug levels.</i>	8

2.1
#define **FLI_INVALID_DEVICE** (-1)

An opaque handle used by library functions to refer to FLI hardware

An opaque handle used by library functions to refer to FLI hardware

2.2

typedef long flidomain_t

The domain of an FLI device.

The domain of an FLI device. This consists of a bitwise ORed combination of interface method and device type. Valid interfaces are `FLIDOMAIN_PARALLEL_PORT`, `FLIDOMAIN_USB`, `FLIDOMAIN_SERIAL`, and `FLIDOMAIN_INET`. Valid device types are `FLIDEVICE_CAMERA`, `FLIDOMAIN_FILTERWHEEL`, and `FLIDOMAIN_FOCUSER`.

See Also: `FLIOpen`
`FLIList`

2.3

typedef long fliframe_t

The frame type for an FLI CCD camera device.

The frame type for an FLI CCD camera device. Valid frame types are `FLI_FRAME_TYPE_NORMAL` and `FLI_FRAME_TYPE_DARK`.

See Also: `FLISetFrameType`

2.4

typedef long flibitdepth_t

The gray-scale bit depth for an FLI camera device.

The gray-scale bit depth for an FLI camera device. Valid bit depths are `FLI_MODE_8BIT` and `FLI_MODE_16BIT`.

See Also: `FLISetBitDepth`

2.5

```
typedef long flishutter_t
```

Type used for shutter operations for an FLI camera device.

Type used for shutter operations for an FLI camera device. Valid shutter types are `FLI_SHUTTER_CLOSE`, `FLI_SHUTTER_OPEN`, `FLI_SHUTTER_EXTERNAL_TRIGGER`, `FLI_SHUTTER_EXTERNAL_TRIGGER_LOW`, and `FLI_SHUTTER_EXTERNAL_TRIGGER_HIGH`.

See Also: `FLIControlShutter`

2.6

```
typedef long flibgflush_t
```

Type used for background flush operations for an FLI camera device.

Type used for background flush operations for an FLI camera device. Valid bgflush types are `FLI_BGFLUSH_STOP` and `FLI_BGFLUSH_START`.

See Also: `FLIControlBackgroundFlush`

2.7

typedef long flichannel_t

Type used to determine which temperature channel to read.

Type used to determine which temperature channel to read. Valid channel types are `FLI_TEMPERATURE_INTERNAL` and `FLI_TEMPERATURE_EXTERNAL`.

See Also: `FLIReadTemperature`

2.8

typedef long flidebug_t

Type specifying library debug levels.

Type specifying library debug levels. Valid debug levels are `FLIDEBUG_NONE`, `FLIDEBUG_INFO`, `FLIDEBUG_WARN`, and `FLIDEBUG_FAIL`.

See Also: `FLISetDebugLevel`

3 Library Functions

Names

- | | | |
|------|--|----|
| 3.1 | LIBFLIAPI FLICancelExposure (flidev_t dev)
<i>Cancel an exposure for a given camera.</i> | 13 |
| 3.2 | LIBFLIAPI FLIEndExposure (flidev_t dev)
<i>End an exposure for a given camera.</i> | 13 |
| 3.3 | LIBFLIAPI FLITriggerExposure (flidev_t dev)
<i>Trigger an exposure that is awaiting an external trigger.</i> | 14 |
| 3.4 | LIBFLIAPI FLIClose (flidev_t dev)
<i>Close a handle to a FLI device</i> | 14 |
| 3.5 | LIBFLIAPI FLIGetArrayArea (flidev_t dev, long* ul_x, long* ul_y, long* lr_x, long* lr_y)
<i>Get the array area of the given camera.</i> | 15 |
| 3.6 | LIBFLIAPI FLIFlushRow (flidev_t dev, long rows, long repeat)
<i>Flush rows of a given camera.</i> | 15 |
| 3.7 | LIBFLIAPI FLIGetFWRevision (flidev_t dev, long* fwrev)
<i>Get firmware revision of a given device</i> | 16 |
| 3.8 | LIBFLIAPI FLIGetHWRevision (flidev_t dev, long* hwrev)
<i>Get the hardware revision of a given device</i> | 16 |
| 3.9 | LIBFLIAPI FLIGetLibVersion (char* ver, size_t len)
<i>Get the current library version. ...</i> | 17 |
| 3.10 | LIBFLIAPI FLIGetSerialString (flidev_t dev, char* serial, size_t len)
<i>Get the serial string of a given device.</i> | 17 |
| 3.11 | LIBFLIAPI FLIGetModel (flidev_t dev, char* model, size_t len)
<i>Get the model of a given device. ...</i> | 18 |
| 3.12 | LIBFLIAPI FLIGetPixelSize (flidev_t dev, double* pixel_x, double* pixel_y) | |

	<i>Find the dimensions of a pixel in the array of the given device</i>	19
3.13	LIBFLIAPI FLIGetVisibleArea (flidev_t dev, long* ul_x, long* ul_y, long* lr_x, long* lr_y) <i>Get the visible area of the given camera.</i>	19
3.14	LIBFLIAPI FLIOpen (flidev_t* dev, char* name, flidomain_t domain) <i>Get a handle to an FLI device.</i>	20
3.15	LIBFLIAPI FLISetDebugLevel (char* host, flidebug_t level) <i>Enable debugging of API operations and communications.</i>	21
3.16	LIBFLIAPI FLISetExposureTime (flidev_t dev, long exptime) <i>Set the exposure time for a camera.</i>	22
3.17	LIBFLIAPI FLISetHBin (flidev_t dev, long hbin) <i>Set the horizontal bin factor for a given camera.</i>	22
3.18	LIBFLIAPI FLISetFrameType (flidev_t dev, fliframe_t frametype) <i>Set the frame type for a given camera.</i>	23
3.19	LIBFLIAPI FLIGetCoolerPower (flidev_t dev, double* power) <i>Get the cooler power level.</i>	23
3.20	LIBFLIAPI FLISetImageArea (flidev_t dev, long ul_x, long ul_y, long lr_x, long lr_y) <i>Set the image area for a given camera.</i>	24
3.21	LIBFLIAPI FLISetVBin (flidev_t dev, long vbin) <i>Set the vertical bin factor for a given camera.</i>	25
3.22	LIBFLIAPI FLIGetExposureStatus (flidev_t dev, long* timeleft) <i>Find the remaining exposure time of a given camera.</i>	25
3.23	LIBFLIAPI FLISetTemperature (flidev_t dev, double temperature) <i>Set the temperature of a given camera.</i>	26
3.24	LIBFLIAPI FLIGetTemperature (flidev_t dev, double* temperature) <i>Get the temperature of a given camera.</i>	26
3.25	LIBFLIAPI FLIGrabRow (flidev_t dev, void* buff, size_t width)	

	<i>Grab a row of an image.</i>	27
3.26	LIBFLIAPI FLIExposeFrame (flidev_t dev) <i>Expose a frame for a given camera.</i>	27
3.27	LIBFLIAPI FLISetBitDepth (flidev_t dev, flibitdepth_t bitdepth) <i>Set the gray-scale bit depth for a given camera.</i>	28
3.28	LIBFLIAPI FLISetNFlushes (flidev_t dev, long nflushes) <i>Set the number of flushes for a given camera.</i>	28
3.29	LIBFLIAPI FLIReadIOPort (flidev_t dev, long* ioportset) <i>Read the I/O port of a given camera.</i>	29
3.30	LIBFLIAPI FLIWriteIOPort (flidev_t dev, long ioportset) <i>Write to the I/O port of a given camera.</i>	30
3.31	LIBFLIAPI FLIConfigureIOPort (flidev_t dev, long ioportset) <i>Configure the I/O port of a given camera.</i>	30
3.32	LIBFLIAPI FLILockDevice (flidev_t dev) <i>Lock a specified device.</i>	31
3.33	LIBFLIAPI FLIUnlockDevice (flidev_t dev) <i>Unlock a specified device.</i>	31
3.34	LIBFLIAPI FLIControlShutter (flidev_t dev, flishutter_t shutter) <i>Control the shutter on a given camera.</i>	31
3.35	LIBFLIAPI FLIControlBackgroundFlush (flidev_t dev, flibgflush_t bgflush) <i>Enables background flushing of CCD array.</i>	32
3.36	LIBFLIAPI FLIList (flidomain_t domain, char*** names) <i>List available devices.</i>	33
3.37	LIBFLIAPI FLIFreeList (char** names) <i>Free a previously generated device list.</i>	33
3.38	LIBFLIAPI FLISetFilterPos (flidev_t dev, long filter) <i>Set the filter wheel position of a given device.</i>	34
3.39	LIBFLIAPI FLIGetFilterPos (flidev_t dev, long* filter)	

		<i>Get the filter wheel position of a given device.</i>	34
3.40	LIBFLIAPI FLIGetStepsRemaining (flidev_t dev, long* steps)	<i>Get the number of motor steps remaining.</i>	35
3.41	LIBFLIAPI FLIGetFilterCount (flidev_t dev, long* filter)	<i>Get the filter wheel filter count of a given device.</i>	35
3.42	LIBFLIAPI FLIStepMotorAsync (flidev_t dev, long steps)	<i>Step the filter wheel or focuser motor of a given device.</i>	36
3.43	LIBFLIAPI FLIStepMotor (flidev_t dev, long steps)	<i>Step the filter wheel or focuser motor of a given device.</i>	36
3.44	LIBFLIAPI FLIGetStepperPosition (flidev_t dev, long* position)	<i>Get the stepper motor position of a given device.</i>	37
3.45	LIBFLIAPI FLIHomeDevice (flidev_t dev)	<i>Home focuser or filter wheel specified by dev The home position of a device is defined as where the electromechanical home sensor detects home.</i>	37
3.46	LIBFLIAPI FLIHomeFocuser (flidev_t dev)	<i>Home focuser dev.</i>	38
3.47	LIBFLIAPI FLIGetFocuserExtent (flidev_t dev, long* extent)	<i>Retrieve the maximum extent for FLI focuser dev</i>	38
3.48	LIBFLIAPI FLIReadTemperature (flidev_t dev, flichannel_t channel, double* temperature)	<i>Retrieve temperature from the FLI focuser dev.</i>	39
3.49	LIBFLIAPI FLICreateList (flidomain_t domain)	<i>Creates a list of all devices within a specified domain.</i>	39
3.50	LIBFLIAPI FLIDeleteList (void)	<i>Deletes a list of devices created by FLICreateList ()</i>	40
3.51	LIBFLIAPI FLIListFirst (flidomain_t* domain, char* filename, size_t flen, char* name, size_t namelen)		

- Obtains the first device in the list.* . 40
- 3.52 LIBFLIAPI **FLIListNext** (flidomain_t* domain, char* filename,
size_t fnlen, char* name, size_t namelen)
Obtains the next device in the list. . 41

3.1LIBFLIAPI **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.

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

Parameters: dev Camera to cancel the exposure of.

See Also: FLIExposeFrame
 FLIEndExposure
 FLIGetExposureStatus
 FLISetExposureTime

3.2LIBFLIAPI **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.

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

Parameters: dev Camera to end the exposure of.

See Also: FLIExposeFrame
 FLICancelExposure
 FLIGetExposureStatus
 FLISetExposureTime

3.3

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.

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

Parameters: dev Camera to trigger the exposure of.

See Also: FLIExposeFrame
 FLICancelExposure
 FLIEndExposure
 FLIGetExposureStatus
 FLISetExposureTime

3.4

LIBFLIAPI FLIClose (flidev_t dev)

Close a handle to a FLI device

Close a handle to a FLI device

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

Parameters: dev The device handle to be closed.

See Also: FLIOpen

3.5

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

Get the array area of the given camera.

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.

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

Parameters:

dev	Camera to get the array area of.
ul_x	Pointer to where the upper-left x-coordinate is to beplaced.
ul_y	Pointer to where the upper-left y-coordinate is to beplaced.
lr_x	Pointer to where the lower-right x-coordinate is to beplaced.
lr_y	Pointer to where the lower-right y-coordinate is to beplaced.

See Also: FLIGetVisibleArea
FLISetImageArea

3.6

LIBFLIAPI FLIFlushRow (fidev_t dev, long rows, long repeat)

Flush rows of a given camera.

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

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

Parameters: dev Camera to flush rows of.
rows Number of rows to flush.
repeat Number of times to flush each row.

See Also: FLISetNFlushes

3.7

LIBFLIAPI FLIGetFWRevision (flidev_t dev, long* fwrev)

Get firmware revision of a given device

Get firmware revision of a given device

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

Parameters: dev Device to find the firmware revision of.
fwrev Pointer to a long which will receive the
firmware revision.

See Also: FLIGetModel
FLIGetHWRevision
FLIGetSerialNum

3.8

LIBFLIAPI FLIGetHWRevision (flidev_t dev, long* hwrev)

Get the hardware revision of a given device

Get the hardware revision of a given device

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

Parameters: dev Device to find the hardware revision of.
hwrev Pointer to a long which will receive the
hardware revision.

See Also: FLIGetModel
FLIGetFWRevision
FLIGetSerialNum

3.9

LIBFLIAPI FLIGetLibVersion (char* ver, size_t len)

Get the current library version.

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.

Parameters: `ver` Pointer to a character buffer where the library versionstring is to be placed.
`len` The size in bytes of the buffer pointed to by `ver`.

3.10

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

Get the serial string of a given device.

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`.

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

Parameters: dev Device to find serial of.
serial Pointer to a character buffer where the serial string is to be placed.
len The size in bytes of buffer pointed to by serial..

See Also: FLIGetHWRevision
FLIGetFWRevision
FLIGetModel

3.11

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

Get the model of a given device.

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`.

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

Parameters: dev Device to find model of.
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`.

See Also: FLIGetHWRevision
FLIGetFWRevision
FLIGetSerialNum

3.12

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

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

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

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

Parameters: dev Device to find the pixel size of.
 pixel_x Pointer to a double which will receive the
 size (inmicons) of a piixel in the x direc-
 tion.
 pixel_y Pointer to a double which will receive the
 size (inmicons) of a piixel in the y direc-
 tion.

See Also: FLIGetArrayArea
 FLIGetVisibleArea

3.13

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

Get the visible area of the given camera.

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.

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

Parameters:

<code>dev</code>	Camera to get the visible area of.
<code>ul_x</code>	Pointer to where the upper-left x-coordinate is to be placed.
<code>ul_y</code>	Pointer to where the upper-left y-coordinate is to be placed.
<code>lr_x</code>	Pointer to where the lower-right x-coordinate is to be placed.
<code>lr_y</code>	Pointer to where the lower-right y-coordinate is to be placed.

See Also: `FLIGetArrayArea`
`FLISetImageArea`

3.14

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

Get a handle to an FLI device.

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.

Return Value:

<code>Zero</code>	on success.
<code>Non-zero</code>	on failure.

Parameters:	<code>dev</code>	Pointer to where a device handle will be placed.
	<code>name</code>	Pointer to a string where the device file-name to be opened is stored. For parallel port devices that are not probed by <code>FLIList()</code> (Windows 95/98/Me), place the address of the parallel port in a string in ASCII form i.e. "0x3778".
	<code>domain</code>	Domain to apply to <code>name</code> for device opening. This is a bitwise ORed combination of interface method and device type. Valid interfaces include <code>FLIDOMAIN_PARALLEL_PORT</code> , <code>FLIDOMAIN_USB</code> , <code>FLIDOMAIN_SERIAL</code> , and <code>FLIDOMAIN_INET</code> . Valid device types include <code>FLIDEVICE_CAMERA</code> , <code>FLIDOMAIN_FILTERWHEEL</code> , and <code>FLIDOMAIN_FOCUSER</code> .
See Also:	<code>FLIList</code> <code>FLIClose</code> <code>flidomain_t</code>	

3.15

LIBFLIAPI FLISetDebugLevel (char* host, flidebug_t level)

Enable debugging of API operations and communications.

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 using 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.

3.16

LIBFLIAPI FLISetExposureTime (flidev_t dev, long exptime)

Set the exposure time for a camera.

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

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

Parameters: `dev` Camera to set the exposure time of.
`exptime` Exposure time in msec.

See Also: `FLIExposeFrame`
`FLICancelExposure`
`FLIGetExposureStatus`

3.17

LIBFLIAPI FLISetHBin (flidev_t dev, long hbin)

Set the horizontal bin factor for a given camera.

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.

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

Parameters: `dev` Camera to set horizontal bin factor of.
`hbin` Horizontal bin factor.

See Also: `FLISetVBin`
`FLISetImageArea`

3.18

LIBFLIAPI FLISetFrameType (flidev_t dev, fliframe_t frame-type)

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.

LIBFLI API FLIGetCoolerPower (flidev_t dev, double* power)

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

Return Value:	Zero	on success.
	Non-zero	on failure.
Parameters:	dev	Camera to find the cooler power of.
	timeleft	Pointer to where the cooler power (in per- cent) will be placed.
See Also:	FLISetTemperature	
	FLIGetTemperature	

3.20

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.

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.

Return Value:	Zero on success. Non-zero on failure.
Parameters:	<div> <div>dev</div> <div>Camera to set image area of.</div> </div> <div> <div>ul_x</div> <div>Upper-left x-coordinate of image area.</div> </div> <div> <div>ul_y</div> <div>Upper-left y-coordinate of image area.</div> </div> <div> <div>lr_x</div> <div>Lower-right x-coordinate of image area (lr'_x from above).</div> </div> <div> <div>lr_y</div> <div>Lower-right y-coordinate of image area (lr'_y from above).</div> </div>
See Also:	<div>FLIGetVisibleArea</div> <div>FLIGetArrayArea</div>

3.21

LIBFLIAPI FLISetVBin (flidev_t dev, long vbin)

Set the vertical bin factor for a given camera.

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.

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

Parameters: dev Camera to set vertical bin factor of.
vbin Vertical bin factor.

See Also: FLISetHBin
FLISetImageArea

3.22

LIBFLIAPI FLIGetExposureStatus (flidev_t dev, long*
timeleft)

Find the remaining exposure time of a given camera.

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`.

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

Parameters: dev Camera to find the remaining exposure time of.
timeleft Pointer to where the remaining exposure time (in milliseconds) will be placed.

See Also: FLIExposeFrame
FLICancelExposure
FLISetExposureTime

3.23

LIBFLIAPI FLISetTemperature (flidev_t dev, double temperature)

Set the temperature of a given camera.

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.

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

Parameters: `dev` Camera device to set the temperature of.
 `temperature` Temperature in Celsius to set CCD camera cold finger to.

See Also: FLIGetTemperature

3.24

LIBFLIAPI FLIGetTemperature (flidev_t dev, double* temperature)

Get the temperature of a given camera.

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`.

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

Parameters: `dev` Camera device to get the temperature of.
 `temperature` Pointer to where the temperature will be placed.

See Also: FLISetTemperature

3.25

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

Grab a row of an image.

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.

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

Parameters: `dev` Camera whose image to grab the next available row from.
 `buff` Pointer to where the next available row will be placed.
 `width` Row width in pixels.

See Also: FLIGrabFrame

3.26

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.

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

Parameters: `dev` Camera to expose the frame of.

See Also: FLISetExposureTime
 FLISetFrameType
 FLISetImageArea
 FLISetHBin
 FLISetVBin
 FLISetNFlashes
 FLISetBitDepth
 FLIGrabFrame
 FLICancelExposure
 FLIGetExposureStatus

3.27

LIBFLIAPI FLISetBitDepth (flidev_t dev, flibitdepth_t bitdepth)

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

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.

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

Parameters: dev Camera to set the bit depth of.
 bitdepth Gray-scale bit depth: FLI_MODE_8BIT or FLI_MODE_16BIT.

See Also: flibitdepth_t
 FLIExposeFrame

3.28

LIBFLIAPI FLISetNFlashes (flidev_t dev, long nflushes)

Set the number of flushes for a given camera.

Return Value:	Zero	on success.
	Non-zero	on failure.
Parameters:	dev	Camera to set the number of flushes of.
	nflushes	Number of times to flush CCD array before an exposure.
See Also:	FLIFlushRow	
	FLIExposeFrame	
	FLIControlBackgroundFlush	

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

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`.

Return Value:	Zero	on success.
	Non-zero	on failure.
Parameters:	dev	Camera to read the I/O port of.
	ioportset	Pointer to where the I/O port data will be stored.
See Also:	FLIWriteIOPort	
	FLIConfigureIOPort	

LIBFLIAPI FLIWriteIOPort (flidev_t dev, long ioportset)

This page was generated with the help of DOC++
<http://www.linuxsupportline.com/~doc++>

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

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

Parameters: `dev` Camera to write I/O port of.
 `ioportset` Data to be written to the I/O port.

See Also: `FLIReadIOPort`
 `FLIConfigureIOPort`

3.31

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

Configure the I/O port of a given camera.

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.

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

Parameters: `dev` Camera to configure the I/O port of.
 `ioportset` Data to configure the I/O port with.

See Also: `FLIReadIOPort`
 `FLIWriteIOPort`

3.32

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.

Return Value: Zero on success.
 Non-zero on failure.
Parameters: dev Device to lock.
See Also: FLIUnlockDevice

3.33

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.

Return Value: Zero on success.
 Non-zero on failure.
Parameters: dev Device to unlock.
See Also: FLILockDevice

3.34

LIBFLIAPI FLIControlShutter (flidev_t dev, flishutter_t shutter)

Control the shutter on a given camera.

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

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

Parameters: dev Device to control the shutter of.
shutter How to control the shutter. A value of `FLI_SHUTTER_CLOSE` closes the shutter and `FLI_SHUTTER_OPEN` opens the shutter. `FLI_SHUTTER_EXTERNAL_TRIGGER_LOW`, `FLI_SHUTTER_EXTERNAL_TRIGGER_HIGH` causes the exposure to begin only when a logic LOW is detected on I/O port bit 0. `FLI_SHUTTER_EXTERNAL_TRIGGER_HIGH` causes the exposure to begin only when a logic HIGH is detected on I/O port bit 0. This setting may not be available on all cameras.

See Also: `flishutter_t`

3.35

LIBFLI API FLIControlBackgroundFlush (flidev_t dev,
flibgflush_t bgflush)

Enables background flushing of CCD array.

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.

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

Parameters: dev Device to control the background flushing of.
bgflush Enables or disables background flushing. A value of `FLI_BGFLUSH_START` begins background flushing. It is important to note that background flushing is stopped whenever `FLIExposeFrame()` or `FLIControlShutter()` are called. `FLI_BGFLUSH_STOP` stops all background flushing activity.

See Also: `flibgflush.t`

3.36

LIBFLI API FLIList (flidomain_t domain, char*** names)

List available devices.

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.

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

Parameters: domain Domain to list the devices of. This is a bit-wise ORed combination of interface method and device type. Valid interfaces include `FLIDOMAIN_PARALLEL_PORT`, `FLIDOMAIN_UB`, `FLIDOMAIN_SERIAL`, and `FLIDOMAIN_INET`. Valid device types include `LIDevice_CAMERA`, `FLIDOMAIN_FILTERWHEEL`, and `FIDOMAIN_FOCUSER`.

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

See Also: `flidomain.t`
`FLIFreeList`
`FLIOpen`

3.37

LIBFLI API 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.

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

Parameters: names Pointer to the list.

See Also: FLIList

3.38

LIBFLIAPI FLISetFilterPos (flidev_t dev, long filter)

Set the filter wheel position of a given device.

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

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

Parameters: dev Filter wheel device handle.
 filter Desired filter wheel position.

See Also: FLIGetFilterPos

3.39

LIBFLIAPI FLIGetFilterPos (flidev_t dev, long* filter)

Get the filter wheel position of a given device.

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

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

Parameters: dev Filter wheel device handle.
 filter Pointer to where the filter wheel position
 will be placed.

See Also: FLISetFilterPos

3.40

LIBFLIAPI FLIGetStepsRemaining (flidev_t dev, long* steps)

Get the number of motor steps remaining.

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

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

Parameters: dev Filter wheel device handle.
filter Pointer to where the number of remaning steps will beplaced.

See Also: FLISetFilterPos

3.41

LIBFLIAPI FLIGetFilterCount (flidev_t dev, long* filter)

Get the filter wheel filter count of a given device.

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.

Parameters: dev Filter wheel device handle.
filter Pointer to where the filter wheel filter count willbe placed.

3.42**LIBFLIAPI FLIStepMotorAsync** (flidev_t dev, long steps)*Step the filter wheel or focuser motor of a given device.*

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.

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

Parameters: dev Filter wheel or focuser device handle.
steps Number of steps to move the focuser or filter wheel.

See Also: FLIGetStepperPosition

3.43**LIBFLIAPI FLIStepMotor** (flidev_t dev, long steps)*Step the filter wheel or focuser motor of a given device.*

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.

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

Parameters: dev Filter wheel or focuser device handle.
steps Number of steps to move the focuser or filter wheel.

See Also: FLIGetStepperPosition

3.44

LIBFLIAPI FLIGetStepperPosition (flidev_t dev, long* position)

Get the stepper motor position of a given device.

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

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

Parameters: dev Filter wheel or focuser device handle.
 position Pointer to where the position of the stepper motor will be placed.

See Also: FLIStepMotor

3.45

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 FLIHome-Focuser(). 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).

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

Parameters: dev Device handle.

See Also: FLIGetDeviceStatus

LIBFLIAPI **FLIHomeFocuser** (flidev_t dev)

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

Parameters: dev Focuser device handle.

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

Retreive the maximum extent for FLI focuser dev

Parameters:	dev	Focuser device handle.
	extent	Pointer to where the maximum extent of the focuser will be placed.

```
LIBFLIAPI FLIReadTemperature (flidev_t dev, flichannel_t
                                channel, double* tempera-
                                ture)
```

This page was generated with the help of DOC++
<http://www.linuxsupportline.com/~doc++>

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.

Parameters: dev Focuser device handle.
channel Channel to be read.
extent Pointer to where the channel temperature will be placed.

3.49**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.

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

Parameters: domain Domain to search for devices, set to zero to search all domains. This parameter must contain the device type.

See Also: `FLIDeleteList`
`FLIListFirst`
`FLIListNext`

3.50**LIBFLIAPI FLIDeleteList** (void)

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

Return Value:	Zero	on success.
	Non-zero	on failure.
See Also:	FLICreateList	
	FLIListFirst	
	FLIListNext	

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

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.

Return Value:	Zero	on success.
	Non-zero	on failure.
Parameters:	domain	Pointer to where to domain of the device will be placed.
	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.
See Also:	FLICreateList	
	FLIDeleteList	
	FLIListNext	

3.52

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

Obtains the next device in the list.

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().

Return Value:	Zero	on success.
	Non-zero	on failure.
Parameters:	domain	Pointer to where to domain of the device will be placed.
	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.
See Also:	FLICreateList	
	FLIDeleteList	
	FLIListFirst	