Blink SDK with Overdrive Plus

Generated by Doxygen 1.8.11

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 Blink_SDK Class Reference

```
#include <Blink_SDK.h>
```

Public Member Functions

- Blink_SDK (unsigned int SLM_bit_depth, unsigned int ∗n_boards_found, bool ∗constructed_ok, bool is_↩ nematic_type=true, bool RAM_write_enable=true, bool use_GPU_if_available=true, size_t max_transient_↩ frames=20U, const char ∗static_regional_lut_file=0)

  Constructor for the Blink SDK.
- ∼Blink_SDK ()

  Destructor for the Blink SDK.
- bool Is_overdrive_available () const

  Returns `true` if overdrive functionality is built into this version of the SDK, otherwise `false`.
- bool Is_slm_transient_constructed () const

  Returns the state of the overdrive wrapper class responsible for transient frame calculations.
- bool Write_overdrive_image (int board, const unsigned char ∗target_phase, bool wait_for_trigger=false, bool external_pulse=false, unsigned int trigger_timeout_ms=0.0)

  Writes an image to the SLM using the intermediate transient frames calculated with overdrive.
- bool Calculate_transient_frames (const unsigned char ∗target_phase, unsigned int ∗byte_count)

  Calculates the series of frames to be sent to the SLM to transition to `target_phase` using overdrive.
- bool Retrieve_transient_frames (unsigned char ∗frame_buffer)

  Retrieves the data for a previously-calculated series of frames. Typically a call to this function is preceded by a call to Calculate_transient_frames.
- bool Write_transient_frames (int board, const unsigned char ∗frame_buffer, unsigned int max_display_↩ frames=0U, bool wait_for_trigger=false, bool external_pulse=false, unsigned int trigger_timeout_ms=0.0)

  Writes the sequence of frames in `frame_buffer` to the SLM.
- bool Read_transient_buffer_size (const char ∗filename, unsigned int ∗byte_count)

  Reads the file header and retrieves the number of bytes to be allocated for reading the frame.
- bool Read_transient_buffer (const char ∗filename, unsigned int byte_count, unsigned char ∗frame_buffer)

Reads the series of transient frames from the file into `frame_buffer`, which must point to sufficient memory to hold the entire buffer.

- bool Save_transient_frames (const char ∗filename, const unsigned char ∗frame_buffer)

  Writes transient frame data to a file.
- void Stop_sequence ()
- const char ∗ Get_last_error_message () const

  Returns a pointer to the string corresponding to the last error condition detected. If no error has been detected, the string is "Blink SDK: No error".
- bool Load_overdrive_LUT_file (const char ∗static_regional_lut_file)

  Loads a new set of LUT data for transient calculations.
- bool Load_linear_LUT (int board)

  Forces a linear LUT to be loaded to the SLM.
- size_t Get_bits_per_pixel () const

  Returns the number of bits for each pixel on the SLM (typically 8 or 16).
- int Get_image_height (int board) const
- int Get_image_width (int board) const
- const char ∗ Get_version_info () const

  Returns a pointer to the string with version information for this SDK.
- bool SLM_power (int board, bool power_state)

  Turns the SLM on or off for `board`.
- void SLM_power (bool power_state)

  Turns all SLMs on or off.
- bool Write_image (int board, const unsigned char ∗image, unsigned int image_size, bool wait_for_trigger=false, bool external_pulse=false, unsigned int trigger_timeout_ms=0.0)

  Write a non-overdrive image to the SLM controlled by `board`.
- bool Load_LUT_file (int board, const char ∗LUT_file)

  Loads the specified LUT file to the SLM.
- int Compute_TF (float frame_rate)
- void Set_true_frames (int true_frames)
- bool Set_coverglass_flipping (int board, bool flipping)
- bool Set_correction_type (int board, bool WFC)
- bool Write_cal_buffer (int board, const unsigned char ∗buffer)
- bool Select_cal_frame (int board, int frame)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 Blink_SDK::Blink_SDK (unsigned int *SLM_bit_depth*, unsigned int ∗ *n_boards_found*, bool ∗ *constructed_ok*, bool *is_nematic_type* = `true`, bool *RAM_write_enable* = `true`, bool *use_GPU_if_available* = `true`, size_t *max_transient_frames* = `20U`, const char ∗ *static_regional_lut_file* = `0`)

Constructor for the Blink SDK.

Parameters

SLM_bit_depth	Options are currently `8` or `16`
n_boards_found	Initial value ignored; set to the number of SLM boards found that have the requested resolution.
constructed_ok	`true` if all elements of the SDK were properly constructed, else `false`.

Parameters

is_nematic_type	`true` for a nematic SLM (usual case); `false` for FLC.
RAM_write_enable	`true` for writing to RAM (usual case) `false` for slower writes.
use_GPU_if_available	`true` to use a GPU; `false` to use a CPU for OverDrive calculations. If `true` is provided, but no GPU is available, then a CPU will be used.
max_transient_frames	The maximum number of transient frames calculated by the OverDrive Plus algorithm.
static_regional_lut_file	Regional LUT file; used for OverDrive calculations. Null for non-OD.

See also

Get_last_error_message, Is_slm_transient_constructed

3.1.1.2 Blink_SDK::∼Blink_SDK ( )

Destructor for the Blink SDK.

3.1.2 Member Function Documentation

3.1.2.1 bool Blink_SDK::Calculate_transient_frames (const unsigned char ∗ *target_phase,* unsigned int ∗ *byte_count*)

Calculates the series of frames to be sent to the SLM to transition to `target_phase` using overdrive.

Parameters

target_phase	Image of the target phase for the SLM. Phase values from 0 to 1.0 correspond to pixel value 0 and 255.
byte_count	Set by this function to the number of bytes required to store the sequence of frames. This parameter must not be NULL. Initial value is ignored.

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message.

3.1.2.2 int Blink_SDK::Compute_TF (float *frame_rate*)

Parameters

frame_rate	

Returns

`true` if there were no errors, otherwise `false`.

3.1.2.3 size_t Blink_SDK::Get_bits_per_pixel ( ) const

Returns the number of bits for each pixel on the SLM (typically 8 or 16).

Returns

Number of bits per pixel.

3.1.2.4 int Blink_SDK::Get_image_height (int *board*) const

3.1.2.5 int Blink_SDK::Get_image_width (int *board*) const

3.1.2.6 const char$*$ Blink_SDK::Get_last_error_message ( ) const

Returns a pointer to the string corresponding to the last error condition detected. If no error has been detected, the string is "Blink SDK: No error".

Returns

Null-terminated C string.

3.1.2.7 const char$*$ Blink_SDK::Get_version_info ( ) const

Returns a pointer to the string with version information for this SDK.

Returns

Null-terminated C string.

3.1.2.8 bool Blink_SDK::Is_overdrive_available ( ) const

Returns `true` if overdrive functionality is built into this version of the SDK, otherwise `false`.

3.1.2.9 bool Blink_SDK::Is_slm_transient_constructed ( ) const

Returns the state of the overdrive wrapper class responsible for transient frame calculations.

Returns

`true` if there were no internal errors constructing the SLM_transient class, otherwise `false`.

See also

Get_last_error_message.

3.1.2.10 bool Blink_SDK::Load_linear_LUT (int *board*)

Forces a linear LUT to be loaded to the SLM.

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message()

3.1.2.11 bool Blink_SDK::Load_LUT_file (int *board,* const char $*$ *LUT_file*)

Loads the specified LUT file to the SLM.

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
LUT_file	Fully-qualified path to LUT file.

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message

3.1.2.12 bool Blink_SDK::Load_overdrive_LUT_file (const char ∗ *static_regional_lut_file*)

Loads a new set of LUT data for transient calculations.

Parameters

static_regional_lut_file	File with regional LUT data.

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message()

3.1.2.13 bool Blink_SDK::Read_transient_buffer (const char ∗ *filename,* unsigned int *byte_count,* unsigned char ∗ *frame_buffer*)

Reads the series of transient frames from the file into `frame_buffer`, which must point to sufficient memory to hold the entire buffer.

Call Read_transient_buffer_size() to determine the required buffer size. Pass the size of frame_buffer in byte_count (for error checking).

Parameters

filename	Name of the file containing transient data.
byte_count	Number of bytes that have been allocated in frame_buffer.
frame_buffer	Buffer to hold the frame data read from the file.

Returns

`true` if there were no errors, otherwise `false`.

See also

Read_transient_buffer_size(), Get_last_error_message().

3.1.2.14 bool Blink_SDK::Read_transient_buffer_size (const char ∗ *filename,* unsigned int ∗ *byte_count*)

Reads the file header and retrieves the number of bytes to be allocated for reading the frame.

Call this function before calling Read_transient_buffer(), and allocate the appropriate buffer size for subsequent use by Read_transient_buffer().

Parameters

filename	Name of the file containing transient data.
byte_count	Set by this function to the number of bytes to be allocated. This parameter must not be NULL. Initial value is ignored.

Returns

`true` if there were no errors, otherwise `false`.

See also

Read_transient_buffer(), Get_last_error_message().

3.1.2.15 bool Blink_SDK::Retrieve_transient_frames (unsigned char ∗ *frame_buffer*)

Retrieves the data for a previously-calculated series of frames. Typically a call to this function is preceded by a call to Calculate_transient_frames.

Parameters

frame_buffer	Pointer to a caller-provided memory area of sufficient size to store the frame data.

Returns

`true` if there were no errors, otherwise `false`.

See also

CalculateTransientFrames, Get_last_error_message.

3.1.2.16 bool Blink_SDK::Save_transient_frames (const char ∗ *filename,* const unsigned char ∗ *frame_buffer*)

Writes transient frame data to a file.

Parameters

filename	Name of the file to be written.
frame_buffer	Frame data to be written to the file.

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message().

3.1.2.17 bool Blink_SDK::Select_cal_frame (int *board,* int *frame*)

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
frame	

Returns

`true` if there were no errors, otherwise `false`.

3.1.2.18 bool Blink_SDK::Set_correction_type (int *board,* bool *WFC*)

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
WFC	

Returns

`true` if there were no errors, otherwise `false`.

3.1.2.19 bool Blink_SDK::Set_coverglass_flipping (int *board,* bool *flipping*)

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
flipping	

Returns

`true` if there were no errors, otherwise `false`.

3.1.2.20 void Blink_SDK::Set_true_frames (int *true_frames*)

Parameters

true_frames	

Returns

3.1.2.21 bool Blink_SDK::SLM_power (int *board*, bool *power_state*)

Turns the SLM on or off for `board`.

Parameters

power_state	`true` for ON, `false` for OFF
board	Index of the board with the required SLM. The index is 1-based (not 0-based).

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message

3.1.2.22 void Blink_SDK::SLM_power (bool *power_state*)

Turns all SLMs on or off.

Parameters

power_state	`true` for ON, `false` for OFF

3.1.2.23 void Blink_SDK::Stop_sequence ( )

3.1.2.24 bool Blink_SDK::Write_cal_buffer (int *board*, const unsigned char $*$ *buffer*)

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
buffer	

Returns

`true` if there were no errors, otherwise `false`.

3.1.2.25 bool Blink_SDK::Write_image (int *board*, const unsigned char ∗ *image*, unsigned int *image_size*, bool *wait_for_trigger* = `false`, bool *external_pulse* = `false`, unsigned int *trigger_timeout_ms* = `0.0`)

Write a non-overdrive image to the SLM controlled by `board`.

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
image	The image to write to the SLM.
image_size	SLM width or height (a square SLM is assumed).
wait_for_trigger	If supported by hardware, this enables use of an external trigger to load images to the SLM.
external_pulse	Enables an external pulse when the image is written to the SLM.
trigger_timeout_ms	If triggering is enabled and no trigger arrives within this time, function returns (false).

Returns

`true` if the image was written successfully, otherwise `false`.

See also

Get_last_error_message

3.1.2.26 bool Blink_SDK::Write_overdrive_image (int *board*, const unsigned char ∗ *target_phase*, bool *wait_for_trigger* = `false`, bool *external_pulse* = `false`, unsigned int *trigger_timeout_ms* = `0.0`)

Writes an image to the SLM using the intermediate transient frames calculated with overdrive.

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
target_phase	Image of the target phase for the SLM.
wait_for_trigger	If supported by hardware, this enables use of an external trigger to load images to the SLM.
external_pulse	Enables an external pulse on the last transient frame.
trigger_timeout_ms	If triggering is enabled and no trigger arrives within this time, function returns (returns false).

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message.

3.1.2.27 bool Blink_SDK::Write_transient_frames (int *board*, const unsigned char ∗ *frame_buffer*, unsigned int *max_display_frames* = `0U`, bool *wait_for_trigger* = `false`, bool *external_pulse* = `false`, unsigned int *trigger_timeout_ms* = `0.0`)

Writes the sequence of frames in `frame_buffer` to the SLM.

Parameters

board	Index of the board with the required SLM. The index is 1-based (not 0-based).
frame_buffer	Contains the sequence of frames to be written to the SLM.
max_display_frames	0 to display all frames in the sequence; non-zero to display no more than `max_display_frames` of the frames in `frame_buffer`.
wait_for_trigger	If supported by hardware, this enables use of an external trigger to load images to the SLM.
external_pulse	Enables an external pulse on the last transient frame.
trigger_timeout_ms	If triggering is enabled and no trigger arrives within this time, function returns (returns false).

Returns

`true` if there were no errors, otherwise `false`.

See also

Get_last_error_message.

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

- Blink_SDK.h

Chapter 4

File Documentation

4.1 Blink_SDK.h File Reference

```
#include <cstddef>
```

Classes

- class Blink_SDK

Macros

- #define BLINK_SDK_API

4.1.1 Detailed Description

Interface to the Blink SDK.

4.1.2 Using the Blink OverDrive SDK

4.1.2.1 General Overview

All but two overdrive functions return a `bool` value to indicate success or failure. When a function returns `false`, call Get_last_error_message() to get a text string with information about the failure. There are effectively three modes of operation using this SDK with overdrive.

4.1.2.2 Calculate and send frames to SLM

```
<<>>
```

4.1.2.3 Pre-calculate frames and store in memory before sending to SLM

<<>>

4.1.2.4 Load/save pre-calculated frames to files

<<>>

4.1.3 Macro Definition Documentation

4.1.3.1 #define BLINK_SDK_API

Index