Thorlabs Polarization Processor C API Reference

Rev. 2.0 2021-08-05 ITN004107-D01

Contents

0.1 Thorlabs Scientific Polarization Processor

0.1.1 Introduction

The target audience for this document is the experienced software engineer with a background in polarized image processing.

0.2 File Index

0.2.1 File List

Here is a list of all documented files with brief descriptions:

0.3 File Documentation

0.3.1 tl_polarization_processor.h File Reference

```
#include "tl_polarization_processor_enums.h"
```

Typedefs

- typedef int(∗ TL_POLARIZATION_PROCESSOR_MODULE_INITIALIZE) (void)
- typedef int(∗ TL_POLARIZATION_PROCESSOR_CREATE_POLARIZATION_PROCESSOR) (void ∗∗)
- typedef int(∗ TL_POLARIZATION_PROCESSOR_SET_CUSTOM_CALIBRATION_COEFFICIENTS) (void ∗, float ∗, float ∗, float ∗, float ∗)
- typedef int(∗ TL_POLARIZATION_PROCESSOR_GET_CUSTOM_CALIBRATION_COEFFICIENTS) (void ∗, float ∗, float ∗, float ∗, float ∗)
- typedef int(∗ TL_POLARIZATION_PROCESSOR_TRANSFORM) (void ∗, enum TL_POLARIZATION_PROCESSOR_POLAR_PHASE, unsigned short ∗, int, int, int, int, int, unsigned short, float ∗, unsigned short ∗, unsigned short ∗, unsigned short ∗, unsigned short ∗, unsigned short ∗)
- typedef int(∗ TL_POLARIZATION_PROCESSOR_DESTROY_POLARIZATION_PROCESSOR) (void ∗)
- typedef int(∗ TL_POLARIZATION_PROCESSOR_MODULE_TERMINATE) (void)

0.3.1.1 Typedef Documentation

0.3.1.1.1 TL_POLARIZATION_PROCESSOR_CREATE_POLARIZATION_PROCESSOR

```
typedef int(* TL_POLARIZATION_PROCESSOR_CREATE_POLARIZATION_PROCESSOR) (void **)
```

This function creates a polarization processing instance and returns a pointer to that instance.

The polarization processor instance is a handle to the internal polarization processing state which consists of:

- polarization calibration coefficients

Parameters

out	*pp_handle*	A pointer to pointer to a polarization processor handle. This argument captures the instance that is returned by the function.

Returns

A TL_POLARIZATION_PROCESSOR_ERROR value to indicate success or failure (TL_POLARIZATION_PROCESSOR_ERROR_NONE indicates success).

0.3.1.1.2 TL_POLARIZATION_PROCESSOR_DESTROY_POLARIZATION_PROCESSOR

```
typedef int(* TL_POLARIZATION_PROCESSOR_DESTROY_POLARIZATION_PROCESSOR) (void *)
```

This function destroys the specified polarization processing instance. After this function has been called for the specified instance handle, it is an error to subsequently use that instance in any way. Any attempt to do so could result in undefined and unpredictable behavior.

Parameters

in	*pp_handle*	A polarization processor handle.

Returns

A TL_POLARIZATION_PROCESSOR_ERROR value to indicate success or failure (TL_POLARIZATION_PROCESSOR_ERROR_NONE indicates success).

0.3.1.1.3 TL_POLARIZATION_PROCESSOR_GET_CUSTOM_CALIBRATION_COEFFICIENTS

```
typedef int(* TL_POLARIZATION_PROCESSOR_GET_CUSTOM_CALIBRATION_COEFFICIENTS) (void *, float *, float *, float *, float *)
```

This function allows the caller to get the polarization processor calibration coefficients.

The coefficients are obtained for each pixel in the polarization processor quartet.

The coefficients are specified for the 4 different polar phases of the origin pixel in the quartet:

- 0 (zero) degrees polar phase
- 45 degrees polar phase
- 90 degress polar phase
- 135 degrees polar phase

Parameters

in	*pp_handle*	A polarization processor handle.
in	*calibration_4_x_4_matrix_0_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 0 (zero) degrees phase pixel in the quartet.
in	*calibration_4_x_4_matrix_45_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 45 degrees phase pixel in the quartet.
in	*calibration_4_x_4_matrix_90_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 90 degrees phase pixel in the quartet.
in	*calibration_4_x_4_matrix_135_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 135 degrees phase pixel in the quartet.

Returns

A TL_POLARIZATION_PROCESSOR_ERROR value to indicate success or failure (TL_POLARIZATION_PROCESSOR_ERROR_NONE indicates success).

0.3.1.1.4 TL_POLARIZATION_PROCESSOR_MODULE_INITIALIZE

```
typedef int(* TL_POLARIZATION_PROCESSOR_MODULE_INITIALIZE) (void)
```

This function initializes the polarization processing module. It must be called prior to calling any other polarization processing module API function.

Returns

A TL_POLARIZATION_PROCESSOR_ERROR value to indicate success or failure (TL_POLARIZATION_PROCESSOR_ERROR_NONE indicates success).

0.3.1.1.5 TL_POLARIZATION_PROCESSOR_MODULE_TERMINATE

```
typedef int(* TL_POLARIZATION_PROCESSOR_MODULE_TERMINATE) (void)
```

This function gracefully terminates the polarization processing module. It must be called prior to unloading the polarization processor component to ensure proper cleanup of platform resources.

Returns

A TL_POLARIZATION_PROCESSOR_ERROR value to indicate success or failure (TL_POLARIZATION_PROCESSOR_ERROR_NONE indicates success).

0.3.1.1.6 TL_POLARIZATION_PROCESSOR_SET_CUSTOM_CALIBRATION_COEFFICIENTS

```
typedef int(* TL_POLARIZATION_PROCESSOR_SET_CUSTOM_CALIBRATION_COEFFICIENTS) (void *, float *, float *, float *, float *)
```

This function allows the caller to set the polarization processor calibration coefficients.

The coefficients are set for each pixel in the polarization processor quartet.

The coefficients are specified for the 4 different polar phases of the origin pixel in the quartet:

- 0 (zero) degrees polar phase
- 45 degrees polar phase
- 90 degress polar phase
- 135 degrees polar phase

Parameters

in	*pp_handle*	A pointer to a polarization processor handle (pointer to pointer).
out	*calibration_4_x_4_matrix_0_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 0 (zero) degrees phase pixel in the quartet.
out	*calibration_4_x_4_matrix_45_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 45 degrees phase pixel in the quartet.
out	*calibration_4_x_4_matrix_90_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 90 degrees phase pixel in the quartet.
out	*calibration_4_x_4_matrix_135_degrees_phase*	A 16 element float array specifying the calibration coefficients for the 135 degrees phase pixel in the quartet.

Returns

A TL_POLARIZATION_PROCESSOR_ERROR value to indicate success or failure (TL_POLARIZATION_PROCESSOR_ERROR_NONE indicates success).

0.3.1.1.7 TL_POLARIZATION_PROCESSOR_TRANSFORM

```
typedef int(* TL_POLARIZATION_PROCESSOR_TRANSFORM) (void *, enum TL_POLARIZATION_PROCESSOR_POLAR_PHASE, unsigned short *, int, int, int, int, int,
unsigned short, float *, unsigned short *, unsigned short *, unsigned short *, unsigned short *, unsigned short *)
```

This function implements the actual polarization processing computation.

It takes an input array consisting of pixel values from a 2-dimensional image and transforms that data into 1 of several output arrays depending on the desired computation. The following computations are supported:

- Normalized stokes vector coefficients
- Total power
- Horizontal/Vertical linear polarization
- Diagonal linear polarization
- Azimuth
- DOLP (degree of linear polarization)

The caller indicates the desired output computation by specifying a non-zero buffer pointer for the corresponding function argument.

A zero output buffer argument indicates that the corresponding output computation is not wanted and in that case, the computation is skipped.

Parameters

in	*pp_handle*	A polarization processor handle.
in	*polar_phase*	The polar phase (in degrees) of the origin pixel in the input buffer.
in	*input_image_buffer*	A pointer to the input image buffer.
in	*input_image_buffer_x_origin*	The input buffer origin x coordinate relative to the full frame (necessary to support arbitrary ROIs)
in	*input_image_buffer_y_origin*	The input buffer origin y coordinate relative to the full frame (necessary to support arbitrary ROIs)
in	*input_image_buffer_width*	Input image buffer width.
in	*input_image_buffer_height*	Input image buffer height.
in	*input_image_buffer_data_bit_depth*	Input image buffer bit depth.

Parameters

in	*output_buffer_max_scaling_value*	The maximum pixel intensity value that should be used for the output buffers. This value must be between 1 and 65535.
out	*normalized_stokes_vector_coefficients_x2_output_buffer*	Output buffer which captures the normalized stokes vector coefficients s0, s1, s2, and s3. s0 is always 1.0 since s1 and s2 are normalized to it and s3 is always 0.0 since it is not possible for us to determine its value. Therefore, this buffer only contains values for s1 and s2. The order of data in the output buffer is s1_0, s2_0, s1_1, s2_1, ... In other words, it is interleaved. The user should specify a 0 pointer if this output is not needed.
out	*total_optical_power_output_buffer*	Output buffer of the total power (intensity). This is value of the s0 stokes vector coefficient for each pixel. The user should specify a 0 pointer if this output is not needed.
out	*horizontal_vertical_linear_polarization_output_buffer*	Output buffer of the horizontal/vertical linear polarization. This is the value of the s1 stokes vector coefficient for each pixel. The user should specify a 0 pointer if this output is not needed.
out	*diagonal_linear_polarization_output_buffer*	Output buffer of the diagonal linear polarization. This is the value of the s2 stokes vector coefficient for each pixel. The user should specify a 0 pointer if this output is not needed.
out	*azimuth_output_buffer*	Output buffer of the azimuth (polar angle) of each pixel. The user should specify a 0 pointer if this output is not needed.
out	*DOLP_output_buffer*	Output buffer of the DOLP (degree of linear polarization) for each pixel. The user should specify a 0 pointer if this output is not needed.

Returns

A TL_POLARIZATION_PROCESSOR_ERROR value to indicate success or failure (TL_POLARIZATION_PROCESSOR_ERROR_NONE indicates success).

0.3.2 tl_polarization_processor_enums.h File Reference

Enumerations

- enum TL_POLARIZATION_PROCESSOR_POLAR_PHASE { TL_POLARIZATION_PROCESSOR_POLAR_PHASE_0_DEGREES, TL_POLARIZATION_PROCESSOR_POLAR_PHASE_4
TL_POLARIZATION_PROCESSOR_POLAR_PHASE_90_DEGREES, TL_POLARIZATION_PROCESSOR_POLAR_PHASE_135_DEGREES, TL_POLARIZATION_PROCESSOR_POLAR
}

0.3.2.1 Enumeration Type Documentation

0.3.2.1.1 TL_POLARIZATION_PROCESSOR_POLAR_PHASE

```
enum TL_POLARIZATION_PROCESSOR_POLAR_PHASE
```

The TL_POLARIZATION_PROCESSOR_POLAR_PHASE enumeration lists all the possible values (in degrees) that a pixel in a polarization sensor could assume.

The polarization phase pattern is

```
------------------------
|          |          |
|    0     |   -45    |
|          |          |
------------------------
|          |          |
|    45    |    90    |
|          |          |
------------------------
```

The primitive pattern shown above represents the fundamental polarization phase arrangement in a polarization sensor. The basic pattern would extend in the X and Y directions in a real polarization sensor containing millions of pixels.

Notice that the phase of the origin (0, 0) pixel logically determines the phase of every other pixel.

It is for this reason that the phase of this origin pixel is termed the polarization "phase" because it represents the reference point for the phase determination of all other pixels.

Every TSI polarization camera provides the sensor specific polarization phase of the full frame origin pixel as a discoverable parameter.

Enumerator

TL_POLARIZATION_PROCESSOR_POLAR_PHASE_0_DEGREES	0 degrees polarization phase
TL_POLARIZATION_PROCESSOR_POLAR_PHASE_45_DEGREES	45 degrees polarization phase
TL_POLARIZATION_PROCESSOR_POLAR_PHASE_90_DEGREES	90 degrees polarization phase
TL_POLARIZATION_PROCESSOR_POLAR_PHASE_135_DEGREES	135 (-45) degrees polarization phase
TL_POLARIZATION_PROCESSOR_POLAR_PHASE_MAX	A sentinel value (DO NOT USE).

0.3.3 tl_polarization_processor_error.h File Reference

Enumerations

- enum TL_POLARIZATION_PROCESSOR_ERROR {
TL_POLARIZATION_PROCESSOR_ERROR_NONE, TL_POLARIZATION_PROCESSOR_ERROR_UNKNOWN, TL_POLARIZATION_PROCESSOR_ERROR_MODULE_NOT_INITIALIZ
TL_POLARIZATION_PROCESSOR_ERROR_MEMORY_ALLOCATION_FAILURE, TL_POLARIZATION_PROCESSOR_ERROR_NULL_INSTANCE_HANDLE,
TL_POLARIZATION_PROCESSOR_ERROR_NULL_INPUT_BUFFER_POINTER, TL_POLARIZATION_PROCESSOR_ERROR_ALL_OUTPUT_BUFFER_POINTERS_ARE_NULL,
TL_POLARIZATION_PROCESSOR_ERROR_IDENTICAL_INPUT_AND_OUTPUT_BUFFERS, TL_POLARIZATION_PROCESSOR_ERROR_DUPLICATE_OUTPUT_BUFFER,
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_POLAR_PHASE,
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_MAX_SCALING_VALUE, TL_POLARIZATION_PROCESSOR_ERROR_INVALID_IMAGE_WIDTH, TL_POLARIZATION_PROCES
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_IMAGE_DATA_BIT_DEPTH, TL_POLARIZATION_PROCESSOR_ERROR_INITIALIZATION_ERROR,
TL_POLARIZATION_PROCESSOR_ERROR_TERMINATION_ERROR, TL_POLARIZATION_PROCESSOR_ERROR_MAX }

0.3.3.1 Enumeration Type Documentation

0.3.3.1.1 TL_POLARIZATION_PROCESSOR_ERROR

```
enum TL_POLARIZATION_PROCESSOR_ERROR
```

The TL_POLARIZATION_PROCESSOR_ERROR enumeration lists all the possible error codes that any polarization processor API function could return.

Enumerator

TL_POLARIZATION_PROCESSOR_ERROR_NONE	This error code indicates SUCCESS.
TL_POLARIZATION_PROCESSOR_ERROR_UNKNOWN	This error code indicates an unknown error.
TL_POLARIZATION_PROCESSOR_ERROR_MODULE_NOT_INITIALIZED	The module has not been initialized therefore it is in an undefined state.
TL_POLARIZATION_PROCESSOR_ERROR_MEMORY_ALLOCATION_FAILURE	The module has not been initialized therefore it is in an undefined state.
TL_POLARIZATION_PROCESSOR_ERROR_NULL_INSTANCE_HANDLE	The specified module instance handle is NULL.
TL_POLARIZATION_PROCESSOR_ERROR_NULL_INPUT_BUFFER_POINTER	The specified input buffer pointer is NULL.
TL_POLARIZATION_PROCESSOR_ERROR_ALL_OUTPUT_BUFFER_POINTERS_ARE_NULL	All specified output buffers are NULL.

Enumerator

TL_POLARIZATION_PROCESSOR_ERROR_IDENTICAL_INPUT_AND_OUTPUT_BUFFERS	An output buffer has been specified that is identical to the input buffer.
TL_POLARIZATION_PROCESSOR_ERROR_DUPLICATE_OUTPUT_BUFFER	Two or more output buffers are identical.
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_POLAR_PHASE	An invalid (unknown) polar phase was specified.
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_MAX_SCALING_VALUE	An invalid maximum scaling value was specified.
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_IMAGE_WIDTH	An invalid image width was specified.
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_IMAGE_HEIGHT	An invalid image height was specified.
TL_POLARIZATION_PROCESSOR_ERROR_INVALID_IMAGE_DATA_BIT_DEPTH	An invalid image bit depth was specified.
TL_POLARIZATION_PROCESSOR_ERROR_INITIALIZATION_ERROR	This indicates an error during initialization, usually attributed to missing or incompatible dynamic libraries.
TL_POLARIZATION_PROCESSOR_ERROR_TERMINATION_ERROR	This indicates an error during cleanup.
TL_POLARIZATION_PROCESSOR_ERROR_MAX	A sentinel value (DO NOT USE).

Index