SDK C Programming Guide

Seek Thermal Incorporated

Version 4.2.0.0

Table of Contents

Introduction . . . 1
Getting started . . . 2
- Starter kits . . . 2
- Samples . . . 2
- Installation . . . 3
  - Installation steps . . . 4
    - Linux installation . . . 4
    - Windows installation . . . 5
- Build configuration . . . 6
  - Linux build configuration . . . 6
- Building the samples . . . 6
- Building the samples on Linux . . . 6
- Building the samples on Windows . . . 7

Filesystem usage . . . 9
- Filesystem usage on Linux . . . 9
- Filesystem usage on Windows . . . 10

Log level configuration . . . 11
Frame data layout . . . 12
API . . . 14
- Seek Camera Manager API . . . 14
  - `seekcamera_manager_t` . . . 14
  - `seekcamera_manager_event_t` . . . 14
  - `seekcamera_manager_event_callback_t` . . . 15
  - `seekcamera_manager_create` . . . 15
  - `seekcamera_manager_destroy` . . . 16
  - `seekcamera_manager_register_event_callback` . . . 17
  - `seekcamera_manager_get_event_str` . . . 17
- Seek Camera API . . . 18
  - `seekcamera_t` . . . 18
  - `seekcamera_chipid_t` . . . 18
  - `seekcamera_serial_number_t` . . . 18
  - `seekcamera_core_part_number_t` . . . 19
  - `seekcamera_io_type_t` . . . 19
  - `seekcamera_firmware_version_t` . . . 19
  - `seekcamera_app_resources_t` . . . 20
  - `seekcamera_memory_access_callback_t` . . . 20
  - `seekcamera_frame_available_callback_t` . . . 21
  - `seekcamera_color_palette_t` . . . 22

seekcamera_color_palette_data_entry_t 23
seekcamera_color_palette_data_t 23
seekcamera_agc_mode_t 24
seekcamera_linear_agc_lock_mode_t 24
seekcamera_histeq_agc_gain_limit_factor_mode_t 25
seekcamera_histeq_agc_plateau_redistribution_mode_t 26
seekcamera_shutter_mode_t 26
seekcamera_temperature_unit_t 27
seekcamera_usb_io_properties_t 27
seekcamera_spi_io_properties_t 28
seekcamera_io_properties_t 29
seekcamera_filter_t 29
seekcamera_filter_state_t 30
seekcamera_flat_scene_correction_id_t 30
seekcamera_get_io_type 31
seekcamera_get_io_properties 31
seekcamera_get_chipid 32
seekcamera_get_serial_number 32
seekcamera_get_core_part_number 33
seekcamera_get_firmware_version 33
seekcamera_get_thermography_window 34
seekcamera_set_thermography_window 35
seekcamera_update_firmware 35
seekcamera_store_calibration_data 36
seekcamera_store_flat_scene_correction 37
seekcamera_delete_flat_scene_correction 38
seekcamera_load_app_resources 39
seekcamera_store_app_resources 40
seekcamera_capture_session_start 41
seekcamera_capture_session_stop 42
seekcamera_register_frame_available_callback 42
seekcamera_get_color_palette 43
seekcamera_set_color_palette 44
seekcamera_set_color_palette_data 44
seekcamera_get_agc_mode 45
seekcamera_set_agc_mode 45
seekcamera_get_histeq_agc_plateau 46
seekcamera_set_histeq_agc_plateau 47
seekcamera_get_histeq_agc_plateau_redistribution_mode 47
seekcamera_set_histeq_agc_plateau_redistribution_mode 48
seekcamera_get_histeq_agc_gain_limit 49

seekcamera_set_histeq_agc_gain_limit ... 49
seekcamera_get_histeq_agc_gain_limit_factor_mode ... 50
seekcamera_set_histeq_agc_gain_limit_factor_mode ... 51
seekcamera_get_histeq_agc_gain_limit_factor_ymin ... 51
seekcamera_set_histeq_agc_gain_limit_factor_ymin ... 52
seekcamera_get_histeq_agc_gain_limit_factor_xmax ... 52
seekcamera_set_histeq_agc_gain_limit_factor_xmax ... 53
seekcamera_get_histeq_agc_alpha_time_seconds ... 54
seekcamera_set_histeq_agc_alpha_time_seconds ... 54
seekcamera_get_histeq_agc_trim_left ... 55
seekcamera_set_histeq_agc_trim_left ... 56
seekcamera_get_histeq_agc_trim_right ... 56
seekcamera_set_histeq_agc_trim_right ... 57
seekcamera_get_linear_agc_lock_mode ... 58
seekcamera_set_linear_agc_lock_mode ... 58
seekcamera_get_linear_agc_lock_min ... 59
seekcamera_set_linear_agc_lock_min ... 60
seekcamera_get_linear_agc_lock_max ... 60
seekcamera_set_linear_agc_lock_max ... 61
seekcamera_get_shutter_mode ... 62
seekcamera_set_shutter_mode ... 62
seekcamera_get_temperature_unit ... 63
seekcamera_set_temperature_unit ... 63
seekcamera_shutter_trigger ... 64
seekcamera_get_scene_emissivity ... 64
seekcamera_set_scene_emissivity ... 65
seekcamera_get_thermography_offset ... 66
seekcamera_set_thermography_offset ... 66
seekcamera_set_filter_state ... 67
seekcamera_get_filter_state ... 67
Seek Camera Frame API ... 68
seekcamera_frame_t ... 68
seekcamera_frame_format_t ... 68
seekcamera_frame_header_t ... 70
seekcamera_frame_get_frame_by_format ... 73
seekcamera_frame_lock ... 74
seekcamera_frame_unlock ... 75
Seek Camera Error API ... 75
seekcamera_error_t ... 75
seekcamera_error_get_str ... 77
Seek Camera Version API ... 78

seekcamera_version_get_major . . . 78
seekcamera_version_get_minor . . . 78
seekcamera_version_get_patch . . . 79
seekcamera_version_get_internal . . . 79
seekcamera_version_get_qualifier . . . 80
Seek Frame API . . . 80
seekframe_t . . . 80
seekframe_get_width . . . 80
seekframe_get_height . . . 81
seekframe_get_channels . . . 81
seekframe_get_pixel_depth . . . 82
seekframe_get_pixel_padding . . . 82
seekframe_get_line_stride . . . 83
seekframe_get_line_padding . . . 83
seekframe_get_data_size . . . 84
seekframe_get_data . . . 84
seekframe_get_row . . . 85
seekframe_get_pixel . . . 85
seekframe_is_empty . . . 86
seekframe_get_header_size . . . 86
seekframe_get_header . . . 87

Introduction

The Seek Thermal SDK allows developers to interface with our devices easily, flexibly, and efficiently. It uses an event-driven architecture that allows for responsive interaction and control. In addition to the event-driven architecture, the SDK is designed with modularity and separation of responsibility in mind. An important realization of the new design is easy use of multiple cameras for the first time ever.

Cross-platform portability is achieved by providing a C99 interface. We provide pre-built binaries and libraries for a large variety of host systems and toolchains.

Batteries included.

NOTE	Please see LICENSE.txt for legal terms, conditions, and attributions. Contact Seek Thermal if this file is missing.

Getting started

This section discusses how to get started with the Seek Thermal SDK.

Starter kits

Seek Thermal offers plug and play starter kits that allows for easy evaluation of our products. The starter kits come in several flavors (Mosaic Cores, Micro Cores) and IO types (USB and SPI).

Getting started with Mosaic Core USB starter kits

1. Follow the installation steps
2. Connect the starter kit over USB
3. Run one of the pre-built sample applications

Getting started with Mosaic Core SPI starter kits

1. This SDK does not currently support Mosaic Cores over SPI. Please contact Seek Sales for the more information of the current status.

Getting started with Micro Core USB starter kits

1. Follow the installation steps
2. Connect the starter kit over USB.
3. Pair the camera
    - Using the pre-built `seekcamera-sdl` sample application with auto pairing enabled
4. Run one of the pre-built sample applications

Getting started with Micro Core SPI starer kits

1. This SDK does not currently support Micro Cores over SPI. Please contact Seek Sales for the more information of the current status.

Samples

Jump in by running one of our sample applications. Pre-built sample applications are provided for all supported architectures or they can be built from scratch.

NOTE	The sample applications are intended to teach users how to use the SDK. They are not meant to be used as the basis for production quality applications.

Table 1. Samples

Name	Build systems	Description
seekcamera-fsc	Unix Makefile, CMake	Demonstrates how to use the SDK to store and delete flat scene corrections (FSC). Supports USB and SPI. Supports single camera.
seekcamera-fw-update	Unix Makefile, CMake	Demonstrates how to use the SDK to update firmware on the device. Supports USB and SPI. Supports single camera.
seekcamera-probe	Unix Makefile, CMake	Demonstrates how to use the SDK to probe the camera connection information. Supports USB and SPI. Supports multiple cameras.
seekcamera-simple	Unix Makefile, CMake	Demonstrates how to use the SDK to log thermography data to a CSV. Supports USB and SPI. Supports multiple cameras.
seekcamera-sdl	Unix Makefile, CMake	Demonstrates how to use the SDK to render ARGB frames. Supports USB and SPI. Supports multiple cameras.

Installation

Pre-built binaries, libraries, and packages are available on the developer portal.

Table 2. Supported architectures

OS	ARCH	FPU	ABI	LIBC
Linux	aarch32	neon-vfpv4		glibc 2.4+
Linux	aarch64	neon-vfpv4		glibc 2.17+
Linux	aarch64	neon-vfpv4		musl 2.0+
Linux	armv7-a		soft	glibc 2.4+
Linux	armv7-a		soft	musl 2.0+
Linux	armv7-a	neon-vfpv4	softfp	glibc 2.4+
Linux	armv7-a	neon-vfpv4	hard	glibc 2.4+
Linux	armv7-a	neon-vfpv4	softfp	musl 2.0+
Linux	armv7-a	neon-vfpv4	hard	musl 2.0+
Linux	armv6	vfpv2	hard	glibc 2.4+

OS	ARCH	FPU	ABI	LIBC
Linux	x64_64	SSE4.1		glibc 2.14+
Windows	x64	SSE4.1		MSVC 19
macOS	Coming soon			

Installation steps

Linux installation

The recommended method of installing and upgrading the SDK on Linux varies depending on the OS flavor. For Debian based systems (e.g. Raspberry Pi 3), the recommended installation method is via a `.deb` package. For other systems, the recommended installation method is manual.

Debian installation

- Unzip the SDK download.
- Install the `.deb` by using the command line and OS package manager. For example, here is the command when using a Rasberry Pi:
  `<SDK>$ sudo apt-get install ./seekthermal-sdk-dev-4.2.0.0_armhf.deb`
- Reboot or reload the udev rules if this is the first time this version of the SDK has been installed.
  `<SDK>$ sudo udevadm control --reload`

NOTE The SDK can be uninstalled via apt.

Manual installation

- Unzip the SDK download
- Install SDL support.
  `<SDK>$ sudo apt-get install libsdl2-dev`
- From the root of the SDK, determine which architecture you need and move into that directory.
  `<SDK>$ cd <arch>`
- Copy the libraries for the appropriate build architecture to the host system.
  `<SDK/arch>$ sudo cp lib/libseekcamera.so /usr/local/lib`
- Copy the headers to the host system.
  `<SDK/arch>$ sudo cp -r include/* /usr/local/include`
- Install the driver udev rules.
  `<SDK/arch>$ sudo cp driver/udev/10-seekthermal.rules /etc/udev/rules.d`
- Reboot or reload the udev rules if this is the first time this version of the SDK has been installed.
  `<SDK>$ sudo udevadm control --reload`
- Change the permissions of the pre-built sample application.
  `<SDK/arch>$ sudo chmod u+x bin/*`

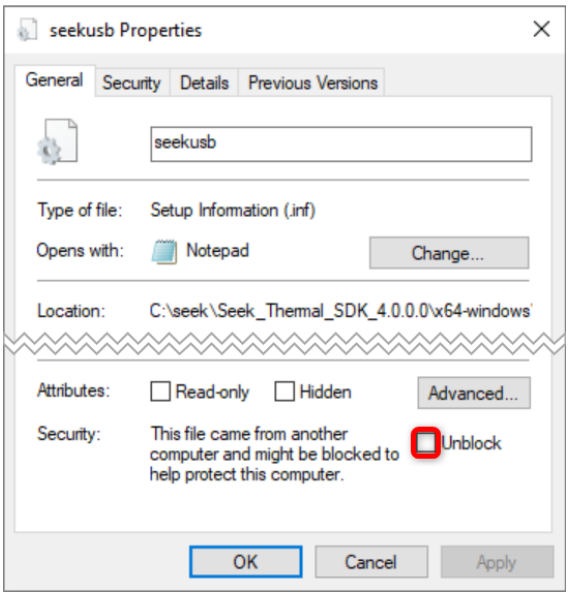
Windows installation

Manual installation

- Unzip the SDK download.
- Install the driver by right click `<SDK>\x64-windows\driver\seekusb.inf` and selecting "Install".

NOTE

Depending on the extraction tool you use to unzip the SDK and the security settings you have on your Windows machine, you may get an **"Unknown Publisher" warning** when installing the driver. You can either **safely ignore** it or select the properties of the SDK zip file and **check the "Unblock" field**. Alternatively you can check the "Unblock" on the `seekusb.inf` once the SDK is unzipped.



Build configuration

Linux build configuration

Building against the SDK is simple. The application build system only needs to know where to locate the libraries and header files.

To have the linker locate the library, add its location to the linker path:

Setting the linker path on a per bash session basis

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
```

Setting the linker path on a per user basis

```
$ echo "export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/usr/local/lib" >> ~/.bashrc
$ source ~/.bashrc
```

Setting the linker path on a per host basis

```
$ sudo echo "LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib" >> /etc/environment
$ sudo reboot
```

Building the samples

The SDK samples can be built using one of the supported build systems.

Building the samples on Linux

CMake is the recommended build system for building the sample applications on Linux.

To generate Unix Makefiles from CMake from the command line:

```
$ cd examples
$ mkdir build
$ cd build
$ cmake ..
```

The above will generate `Makefile` in the the build directory. The samples can then be built from the command line.

```
$ make
```

The binaries will be be placed in a target specific directory under the build directory root.

```
- build/
    - seekcamera-sdl/
        - seekcamera-sdl
        ...
```

Alternatively, the sample applications can be built one by one using the generated Unix Makefile.

```
$ cd seekcamera-sdl
$ make
```

The binaries will be be placed in the current directory.

```
- examples/
    - seekcamera-sdl/
        - seekcamera-sdl
        ...
```

Building the samples on Windows

CMake is the recommended build system for building the sample applications on Windows. CMake is available at https://cmake.org/. Please use version 3.11 or higher.

To generate a Visual Studio solution from the command line:

```
$ cd examples
$ mkdir build
$ cd build
$ cmake ..
```

The above will generate `seekcamera_examples.sln` in the the build directory. The solution file can then be opened in Visual Studio.

Recent versions of CMake allow for building via the command line instead of manually opening and building using a solution file. To build the sample applications from the command line, run the above and then run:

```
$ cmake --build . --config Debug
```

The binaries will be be placed in a `Debug` directory under the build directory root.

```
- build/
    - Debug/
        - seekcamera-sdl.exe
        ...
```

Filesystem usage

Seek Thermal uses the filesystem to cache data and log system status. The paths are standardized based on the host operating system conventions.

Directories

- `cal/`
  - Directory where calibration data is stored for Micro Core cameras. The SDK reads/writes to this directory automatically. Users do not need to manually manage it.
- `conf/`
  - Directory where configuration data is stored. The SDK reads from this directory. Users need to explicitly manage this directory.
- `log/`
  - Directory where logging data is stored. The SDK reads/writes to this directory automatically. Users do not need to manually manage it. It is generally helpful to examine the logs for status information, errors, and warnings.
- `cache/`
  - Directory where miscellaneous data is stored. The SDK reads/writes to this directory automatically. Users do not need to explicitly manage this directory.

Filesystem usage on Linux

The default base SDK directory stored under `$HOME`. It is customizable by setting the environment variable `$SEEKTHERMAL_ROOT`. If customizing the path, then the directory must exist.

```
- $HOME/
    - .seekthermal/
        - cal/
        - conf/
        - log/
        - cache/
```

```
- $SEEKTHERMAL_ROOT/
    - .seekthermal/
        - cal/
        - conf/
        - log/
        - cache/
```

Filesystem usage on Windows

The default base SDK directory stored under `%APPDATA%`. It is customizable by setting the environment variable `%SEEKTHERMAL_ROOT%`. If customizing the path, then the directory must exist.

```
- %APPDATA%/
    - SeekThermal/
        - cal/
        - conf/
        - log/
        - cache/
```

```
- %SEEKTHERMAL_ROOT%/
    - SeekThermal/
        - cal/
        - conf/
        - log/
        - cache/
```

Log level configuration

The SDK log level is customizable by setting an environment variable. On Linux, the environment variable is: `$SEEKTHERMAL_LOG_LEVEL`. On Windows, the environment variable is: `%SEEKTHERMAL_LOG_LEVEL%`. If a log statement is at a lower level than the current level, then it will be suppressed. The default log level is `INFO`.

Level	Priority	Description
TRACE	0	Trace statements, debug statements, info statements, warnings, and errors will be logged.
DEBUG	1	Debug statements, info statements, warnings, and errors will be logged.
INFO	2	Info statements, warnings, and errors will be logged.
WARN	3	Warnings and errors will be logged.
ERROR	4	Errors will be logged.

Frame data layout

Seek Thermal uses a standardized data layout for all our frames. Frame data is stored in row-major order, meaning rows vary fastest and then columns. Pixel channels are stored contiguously, with optional padding between pixels.

A table of supported frame formats is shown below.

Table 3. Supported frame formats

Display	Thermography	Partially Processed
Grayscale	Floating point	Corrected
ARGB888	Fixed point	Pre-AGC
RGB565		
AYUV		
YUY2		

A high-level flow graph of the Seek Thermal frame processing pipeline is shown below.

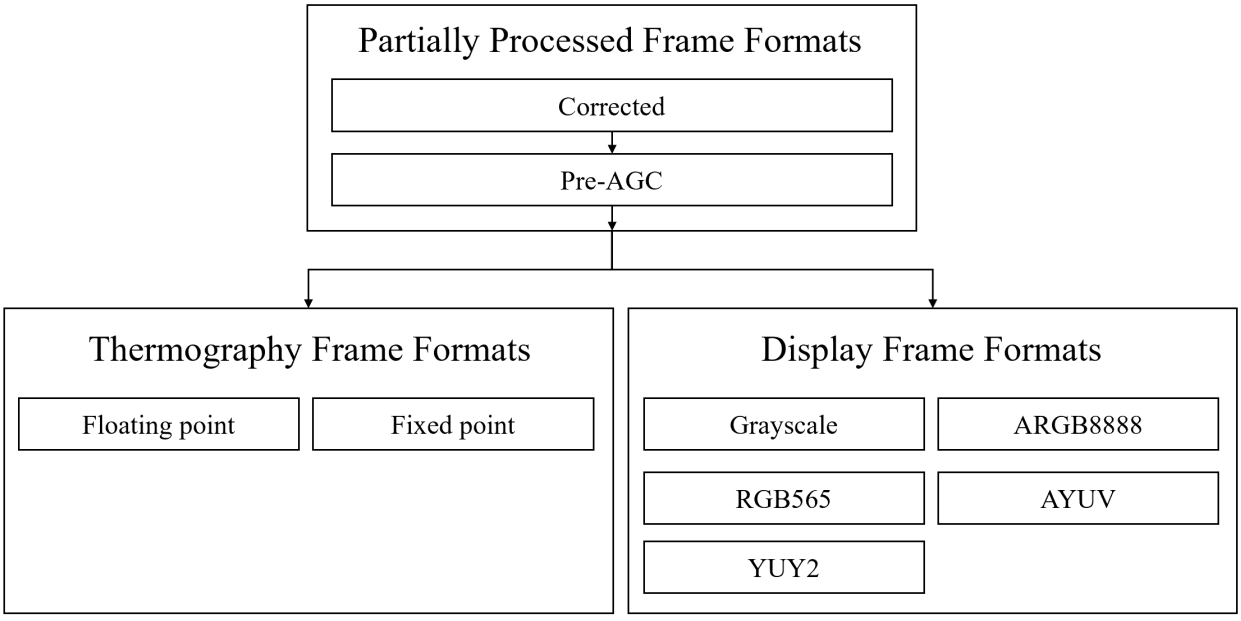


Figure 1. Frame processing flow graph

For descriptions of each supported frame formats, please see the Seek Camera Frame API description.

NOTE Frame buffers also contain a header that is not shown below.

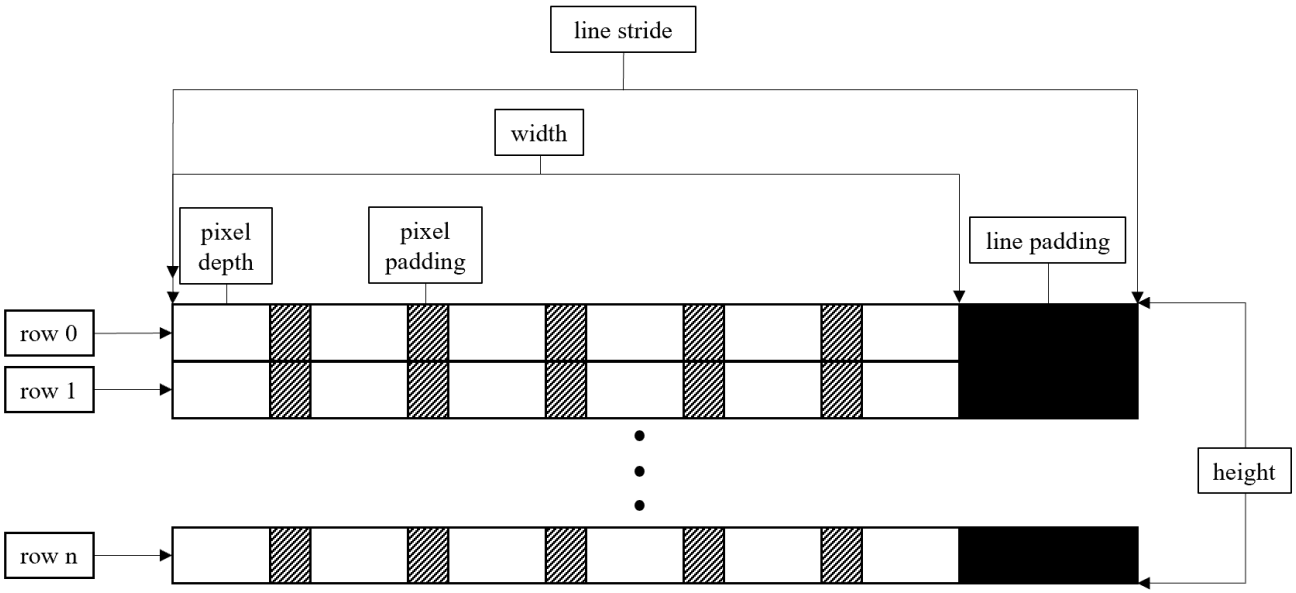


Figure 2. Frame layout

API

Seek Camera Manager API

The Seek Camera Manager API is a high-level interface which allows user applications to interact with Seek devices. It is a part of the Seek Camera API interface. The interface manages multiple Seek devices within a single manager context. It is event driven and uses user-defined callback functions.

`seekcamera_manager_t`

Declaration

```
typedef struct seekcamera_manager_t seekcamera_manager_t;
```

Overview

Structure that manages Seek cameras. It is the interface through which users can set discovery modes, register event callbacks, and access cameras. It is created and destroyed by the user.

`seekcamera_manager_event_t`

Declaration

```
typedef enum seekcamera_manager_event_t
{
    ...
} seekcamera_manager_event_t;
```

Overview

Enumerated type representing types of events used by the camera manager.

Cases

- `SEEKCAMERA_MANAGER_EVENT_CONNECT`
  - Event case when a new camera connects in a paired state.
- `SEEKCAMERA_MANAGER_EVENT_DISCONNECT`
  - Event case when an existing camera disconnects.
- `SEEKCAMERA_MANAGER_EVENT_ERROR`
  - Event case when an existing camera has an error.
- `SEEKCAMERA_MANAGER_EVENT_READY_TO_PAIR`
  - Event case when a new camera connects in an unpaired state.

See also

- seekcamera_manager_get_event_str

seekcamera_manager_event_callback_t

Declaration

```
typedef void (*seekcamera_manager_event_callback_t)(
    seekcamera_t* camera,
    seekcamera_manager_event_t event,
    seekcamera_error_t event_status,
    void* user_data);
```

Overview

Callback function type that is fired every time the camera manager receives an event.

Parameters

- camera
  - Camera on which the event occurred.
- event
  - Enumerated value indicating what type of event occurred.
- event_status
  - Enumerated value indicating the status of the event.
- user_data
  - Optional parameter pointing to user defined data.

Return value

None

See also

- seekcamera_t
- seekcamera_manager_event_t
- seekcamera_manager_register_event_callback

seekcamera_manager_create

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_manager_create(
    seekcamera_manager_t** camera_manager,
    uint32_t discovery_mode);
```

Description

Creates the camera manager responsible for managing Seek cameras.

WARNING	This method will call the C standard library function `getenv`, which is not MT-safe. Avoid calling `getenv` from other threads during this call.

Parameters

- `camera_manager`
  - Camera manager to allocate and initialize. It should be a pointer to unallocated memory (i.e. NULL). On success, it will be in a usable state; otherwise it will be NULL.
- `discovery_mode`
  - Integer indicating the type of devices to be discovered. It corresponds to the enumerated type `seekcamera_io_type_t`. That is, it is a value of `seekcamera_io_type_t` or it is a bitwise OR combination of the enum values.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_manager_t
- seekcamera_manager_destroy
- seekcamera_io_type_t

seekcamera_manager_destroy

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_manager_destroy(
    seekcamera_manager_t** camera_manager);
```

Description

Destroys and invalidates an existing camera manager.

Parameters

- `camera_manager`
  - Camera manager to destroy. It should be a pointer to allocated and initialized memory.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_manager_t
- seekcamera_manager_destroy

seekcamera_manager_register_event_callback

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_manager_register_event_callback(
    seekcamera_manager_t* camera_manager,
    seekcamera_manager_event_callback_t callback,
    void* user_data);
```

Description

Registers a user-defined event callback function with the camera manager. The callback is fired every time the camera manager receives an event. One event callback is allowed to be registered per camera manager.

Parameters

- `camera_manager`
  - Camera manager for which the handler should be registered. It should be a pointer to allocated memory configured by the initialization method.
- `callback`
  - User-defined callback function. The callback function will be called every time an connect, disconnect, or event occurrs.
- `user_data`
  - Optional user-defined data which is provided to the callback.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_manager_t
- seekcamera_manager_event_callback_t

seekcamera_manager_get_event_str

Declaration

```
SEEKCAMERA_API const char* seekcamera_manager_get_event_str(
    seekcamera_manager_event_t event);
```

Description

Gets the description of a camera manager event type.

Parameters

- `event`
  - Enumerated value indicating a camera event.

Return value

The description of a camera manager event type.

See also

- seekcamera_manager_event_t

Seek Camera API

The Seek Camera API is a high-level interface which allows user applications to interact with Seek devices. The interface manages a single Seek device per context. It is event driven and makes use of user-defined callback functions.

seekcamera_t

Declaration

```
typedef struct seekcamera_t seekcamera_t
```

Overview

Structure that represents a single Seek camera. It is the interface through which users can query device characteristics, receive frame data, receive thermography data, etc.

seekcamera_chipid_t

Declaration

```
typedef char seekcamera_chipid_t[16];
```

Overview

A camera chip identifier (CID) represented as a C-string. It is guaranteed to be NULL terminated. It is 16 bytes (B) in size.

See also

- seekcamera_get_chipid

seekcamera_serial_number_t

Declaration

```
typedef char seekcamera_serial_number_t[16];
```

Overview

A camera serial number (SN) represented as a C-string. It is guaranteed to be NULL terminated. It is 16 bytes (B) in size.

See also

- seekcamera_get_serial_number

seekcamera_core_part_number_t

Declaration

```
typedef char seekcamera_core_part_number_t[32];
```

Overview

A camera core part number (CPN) represented as a C-string. It is guaranteed to be NULL terminated. It is 32 bytes (B) in size.

See also

- seekcamera_get_core_part_number

seekcamera_io_type_t

Declaration

```
typedef enum seekcamera_io_type_t
{
    ...
} seekcamera_io_type_t;
```

Cases

- SEEKCAMERA_IO_TYPE_USB
  - IO type case for USB cameras.
- SEEKCAMERA_IO_TYPE_SPI
  - IO type case for SPI cameras.

See also

- seekcamera_get_io_type
- seekcamera_manager_init

seekcamera_firmware_version_t

Declaration

```
typedef struct seekcamera_firmware_version_t
{
    ...
} seekcamera_firmware_version_t;
```

Overview

Structure that represents the firmware version of a Seek camera.

Fields

- `product: uint8_t`
  - Product firmware version.
- `variant: uint8_t`
  - Variant firmware version.
- `major: uint8_t`
  - Major firmware version.
- `minor: uint8_t`
  - Minor firmware version.

See also

- seekcamera_get_firmware_version

seekcamera_app_resources_t

Declaration

```
typedef enum seekcamera_app_resources_region_t
{
    ...
} seekcamera_app_resources_region_t;
```

Overview

Types of application resource regions. Application resource regions are memory regions on the device that are reserved for customer use.

Cases

- `SEEKCAMERA_APP_RESOURCES_REGION_0`
  - Application resource region 0.
- `SEEKCAMERA_APP_RESOURCES_REGION_1`
  - Application resource region 1.
- `SEEKCAMERA_APP_RESOURCES_REGION_2`
  - Application resource region 2.

See also

- seekcamera_load_app_resources
- seekcamera_store_app_resources

seekcamera_memory_access_callback_t

Declaration

```
typedef void (*seekcamera_memory_access_callback_t)(
    size_t progress,
    void* user_data);
```

Overview

Callback function fired every time a memory access is performed. It is used for providing progress feedback for long memory operations. The callbacks execute on the calling thread.

Parameters

- `progress`
  - Progress percentage represented as an integer. It can take value between [0,100].
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

None

`seekcamera_frame_available_callback_t`

Declaration

```
typedef void (*seekcamera_frame_available_callback_t)(
    seekcamera_t* camera,
    seekcamera_frame_t* frame,
    void* user_data);
```

Overview

Callback function fired every time a frame is available. The callbacks execute on a separate, internally owned SDK thread.

Parameters

- `camera`
  - Camera for which the frame is available.
- `frame`
  - Camera frame wrapper structure where the newly available frames are encapsulated. The structure contains the frames formats which were specified during the capture session initialization. The frame is valid during the lifetime of the callback.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

None

See also

- seekcamera_t
- seekcamera_frame_t
- seekcamera_register_frame_available_callback

seekcamera_color_palette_t

Declaration

```
typedef enum seekcamera_color_palette_t
{
    ...
} seekcamera_color_palette_t;
```

Overview

Enumerated type representing types of display color palettes.

Cases

- SEEKCAMERA_COLOR_PALETTE_WHITE_HOT
  - Color palette type case for White Hot.
- SEEKCAMERA_COLOR_PALETTE_BLACK_HOT
  - Color palette type case for Black Hot.
- SEEKCAMERA_COLOR_PALETTE_SPECTRA
  - Color palette type case for Spectra.
- SEEKCAMERA_COLOR_PALETTE_PRISM
  - Color palette type case for Prism.
- SEEKCAMERA_COLOR_PALETTE_TYRIAN
  - Color palette type case for Tyrian.
- SEEKCAMERA_COLOR_PALETTE_IRON
  - Color palette type case for Iron.
- SEEKCAMERA_COLOR_PALETTE_AMBER
  - Color palette type case for Amber.
- SEEKCAMERA_COLOR_PALETTE_HI
  - Color palette type case for Hi.
- SEEKCAMERA_COLOR_PALETTE_GREEN
  - Color palette type case for Green.
- SEEKCAMERA_COLOR_PALETTE_USER_0
  - Color palette type for a user defined color palette.
- SEEKCAMERA_COLOR_PALETTE_USER_1

    - Color palette type for a user defined color palette.
- `SEEKCAMERA_COLOR_PALETTE_USER_2`
    - Color palette type for a user defined color palette.
- `SEEKCAMERA_COLOR_PALETTE_USER_3`
    - Color palette type for a user defined color palette.
- `SEEKCAMERA_COLOR_PALETTE_USER_4`
    - Color palette type for a user defined color palette.

See also

- seekcamera_color_palette_data_t
- seekcamera_color_palette_data_entry_t
- seekcamera_get_color_palette
- seekcamera_set_color_palette

seekcamera_color_palette_data_entry_t

Declaration

```
typedef struct seekcamera_color_palette_data_entry_t
{
    uint8_t a;
    uint8_t r;
    uint8_t g;
    uint8_t b;
} seekcamera_color_palette_data_entry_t;
```

Overview

Structure that represents an entry in color palette data structure. Each field of structure represents a component of an ARGB8888 pixel. The structure has a total size of 32 bytes (B).

See also

- seekcamera_color_palette_t
- seekcamera_color_palette_data_t

seekcamera_color_palette_data_t

Declaration

```
typedef seekcamera_color_palette_data_entry_t seekcamera_color_palette_data_t[256];
```

Overview

An array of color values used to colorize a thermal image. The values should be in ascending order going from coldest to hottest temperature. It has 256 distinct entries and each entry is 32 bytes (B)

in size.

See also

- seekcamera_color_palette_t
- seekcamera_color_palette_data_entry_t
- seekcamera_set_color_palette_data

seekcamera_agc_mode_t

Declaration

```
typedef enum seekcamera_agc_mode_t
{
    ...
} seekcamera_agc_mode_t;
```

Overview

Enumerated type representing types of automated gain correction (AGC) modes.

Cases

- SEEKCAMERA_AGC_MODE_LINEAR
  - AGC mode type case for linear min/max.
- SEEKCAMERA_AGC_MODE_HISTEQ
  - AGC mode type case for histogram equalization.

See also

- seekcamera_get_agc_mode
- seekcamera_set_agc_mode

seekcamera_linear_agc_lock_mode_t

Declaration

```
typedef enum seekcamera_linear_agc_lock_mode_t
{
    ...
} seekcamera_linear_agc_lock_mode_t;
```

Overview

Enumerated type representing types of lock modes used by Linear AGC. The input modes allow for the automatic, semi-automatic, or manual setting of the output AGC range bounds. Regardless of which method is used to set the bounds, the output range (i.e. the closed interval [0, 255]) will be evenly stretched between the bounding values.

Cases

- SEEKCAMERA_LINEAR_AGC_LOCK_MODE_AUTO
  - Lock mode type case for automatic linear AGC control. The minimum and maximum will be determined from the lowest/highest scene values.
- SEEKCAMERA_LINEAR_AGC_LOCK_MODE_MANUAL
  - Lock mode type case for manual linear AGC control. Both the minimum and maximum will be set by the user, as opposed to from the scene.
- SEEKCAMERA_LINEAR_AGC_LOCK_MODE_MANUAL_MIN
  - Lock mode type case for semi-manual linear AGC control. The minimum bound value will be set by the user; the maximum bound value will be determined from the highest scene value.
- SEEKCAMERA_LINEAR_AGC_LOCK_MODE_MANUAL_MAX
  - Lock mode type case for semi-manual linear AGC control. The minimum bound value will be determined from the lowest scene value; the maximum bound value will be set by the user.

See also

- seekcamera_get_linear_agc_lock_mode
- seekcamera_set_linear_agc_lock_mode

seekcamera_histeq_agc_gain_limit_factor_mode_t

Declaration

```
typedef enum seekcamera_histeq_agc_gain_limit_factor_mode_t
{
    ...
} seekcamera_histeq_agc_gain_limit_factor_mode_t;
```

Overview

Enumerated type representing types of gain limit modes used by HistEQ AGC.

Cases

- SEEKCAMERA_HISTEQ_AGC_GAIN_LIMIT_FACTOR_MODE_AUTO
  - Gain limit factor mode type case for automatic gain limit factor control. The gain limit will be set by the user; all other gain limit factor settings will be controlled automatically.
- SEEKCAMERA_HISTEQ_AGC_GAIN_LIMIT_FACTOR_MODE_MANUAL
  - Gain limit factor mode type case for manual gain limit factor control. The gain limit will be set by the user; all other gain limit factor settings will also be controlled by the user.

See also

- seekcamera_get_histeq_agc_gain_limit_factor_mode
- seekcamera_set_histeq_agc_gain_limit_factor_mode

seekcamera_histeq_agc_plateau_redistribution_mode_t

Declaration

```
typedef enum seekcamera_histeq_agc_plateau_redistribution_mode_t
{
    ...
} seekcamera_histeq_agc_plateau_redistribution_mode_t;
```

Overview

Enumerated type representing types of plateau redistribution modes used by HistEQ AGC.

Cases

- SEEKCAMERA_HISTEQ_AGC_PLATEAU_REDISTRIBUTION_MODE_DISABLED
  - Plateau redistribution mode type used by HistEQ AGC. The default plateau redistribution mode. Pixels in any histogram bin that exceed the plateau threshold are discarded. For control of the plateau threshold, see seekcamera_set_histeq_agc_plateau.
- SEEKCAMERA_HISTEQ_AGC_PLATEAU_REDISTRIBUTION_MODE_ALL_BINS
  - Plateau redistribution mode type used by HistEQ AGC. Pixels in any histogram bin that exceed that plateau threshold are redistributed evenly among all bins in the histogram. For control of the plateau threshold, see seekcamera_set_histeq_agc_plateau.
- SEEKCAMERA_HISTEQ_AGC_PLATEAU_REDISTRIBUTION_MODE_ACTIVE_BINS_ONLY
  - Plateau redistribution mode type used by HistEQ AGC. Pixels in any histogram bin that exceed that plateau threshold are redistributed only among the active bins in the histogram. An active histogram bin is one which contains at least 1 pixel. For control of the plateau threshold, see seekcamera_set_histeq_agc_plateau.

See also

- seekcamera_get_histeq_agc_plateau
- seekcamera_set_histeq_agc_plateau
- seekcamera_get_histeq_agc_plateau_redistribution_mode
- seekcamera_set_histeq_agc_plateau_redistribution_mode

seekcamera_shutter_mode_t

Declaration

```
typedef enum seekcamera_shutter_mode_t
{
    ...
} seekcamera_shutter_mode_t;
```

Overview

Enumerated type representing types of shutter modes.

WARNING	Shutter mode is only applicable to Mosaic Cores.

Cases

- `SEEKCAMERA_SHUTTER_MODE_AUTO`
  - Shutter mode type case for automatic shuttering. When automatic shuttering is enabled, the user does not to need to manually trigger the shutter. This is the default shuttering mode.
- `SEEKCAMERA_SHUTTER_MODE_MANUAL`
  - Shutter mode type case for manual shuttering. When manual shuttering is enabled, the user is responsible for triggering the shutter.

See also

- seekcamera_get_shutter_mode
- seekcamera_set_shutter_mode

seekcamera_temperature_unit_t

Declaration

```
typedef enum seekcamera_temperature_unit_t
{
    ...
} seekcamera_temperature_unit_t;
```

Overview

Enumerated type representing types of temperature units.

Cases

- `SEEKCAMERA_TEMPERATURE_UNIT_CELSIUS`
  - Temperature unit type case for degrees Celsius. This is the default temperature unit.
- `SEEKCAMERA_TEMPERATURE_UNIT_FAHRENHEIT`
  - Temperature unit type case for degrees Fahrenheit.
- `SEEKCAMERA_TEMPERATURE_UNIT_KELVIN`
  - Temperature unit type case for Kelvin.

See also

- seekcamera_get_temperature_unit
- seekcamera_set_temperature_unit

seekcamera_usb_io_properties_t

Declaration

```
typedef struct seekcamera_usb_io_properties_t
{
    ...
} seekcamera_usb_io_properties_t;
```

Overview

Structure that contains properties of USB cameras.

Fields

- `bus_number: uint8_t`
  - USB bus number on which the camera is connected.
- `port_numbers: uint8_t[8]`
  - USB port numbers on which the camera is connected. Valid port numbers are indicated by a value strictly greater than zero.

See also

- seekcamera_io_properties_t
- seekcamera_get_io_properties

seekcamera_spi_io_properties_t

Declaration

```
typedef struct seekcamera_spi_io_properties_t
{
    ...
} seekcamera_spi_io_properties_t;
```

Overview

Structure that contains properties of SPI cameras.

Fields

- `bus_number: uint8_t`
  - SPI bus number on which the camera is connected. This field corresponds to the bus number set in the SPI configuration file.
- `cs_number: uint8_t`
  - SPI chip select number on which the camera is connected. This field corresponds to the chip select (cs) number set in the SPI configuration file.

See also

- seekcamera_io_properties_t
- seekcamera_get_io_properties

seekcamera_io_properties_t

Declaration

```
typedef struct seekcamera_io_properties_t
{
    ...
} seekcamera_io_properties_t;
```

Overview

Generic structure that contains the IO properties of the camera. The IO properties are stored as a union.

Fields

- `type: seekcamera_io_type_t`
  - IO type of the camera.
- `usb: seekcamera_usb_io_properties_t`
  - Structure that contains properties of USB cameras.
- `spi: seekcamera_spi_io_properties_t`
  - Structure that contains properties of SPI cameras.

See also

- seekcamera_get_io_properties
- seekcamera_io_type_t
- seekcamera_usb_io_properties_t
- seekcamera_spi_io_properties_t

seekcamera_filter_t

Declaration

```
typedef enum seekcamera_filter_t
{
    ...
} seekcamera_filter_t;
```

Overview

Enumerated type representing the controllable image processing filters.

Cases

- `SEEKCAMERA_FILTER_GRADIENT_CORRECTION`
  - Filter responsible for correcting image gradient on all data pipelines. It is triggered automatically on flat scenes.
- `SEEKCAMERA_FILTER_FLAT_SCENE_CORRECTION`

- Filter responsible for correcting non-uniformities on all data pipelines. It is stored explicitly by the user apriori.

See also

- seekcamera_filter_state_t
- seekcamera_set_filter_state
- seekcamera_get_filter_state

seekcamera_filter_state_t

Declaration

```
typedef enum seekcamera_filter_state_t
{
    ...
} seekcamera_filter_state_t;
```

Cases

- SEEKCAMERA_FILTER_STATE_DISABLED
  - Filter state type case for a disabled filter.
- SEEKCAMERA_FILTER_STATE_ENABLED
  - Filter state type case for an enabled filter.

See also

- seekcamera_filter_t
- seekcamera_set_filter_state
- seekcamera_get_filter_state

seekcamera_flat_scene_correction_id_t

Declaration

```
typedef enum seekcamera_flat_scene_correction_id_t
{
    ...
} seekcamera_flat_scene_correction_id_t;
```

Overview

Enumerated type representing a unique flat scene correction (FSC) identifier.

Cases

- SEEKCAMERA_FLAT_SCENE_CORRECTION_ID_0
  - The default FSC ID. If a FSC with this ID is stored, then it will be autoloaded and applied on startup.

See also

- seekcamera_store_flat_scene_correction
- seekcamera_delete_flat_scene_correction

seekcamera_get_io_type

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_io_type(
    seekcamera_t* camera,
    seekcamera_io_type_t* type);
```

Overview

Gets the IO type of the camera.

Parameters

- `camera`
  - Camera for which to get the IO type.
- `type`
  - IO type of the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_io_type_t

seekcamera_get_io_properties

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_io_properties(
    seekcamera_t* camera,
    seekcamera_io_properties_t* properties);
```

Overview

Gets the IO properties of the camera.

Parameters

- `camera`
  - Camera for which to get the IO properties.
- `properties`
  - IO properties of the camera. It is an output paramter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_io_properties_t

seekcamera_get_chipid

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_chipid(
    seekcamera_t* camera,
    seekcamera_chipid_t* chipid);
```

Overview

Gets the chip identifier (CID) of the camera.

Parameters

- `camera`
  - Camera for which to get the CID.
- `chipid`
  - CID of the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_chipid_it

seekcamera_get_serial_number

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_serial_number(
    seekcamera_t* camera,
    seekcamera_serial_number_t* serial_number);
```

Overview

Gets the serial number (SN) of the camera.

Parameters

- `camera`
  - Camera for which to get the SN.

- `serial_number`
  - SN of the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_serial_number_t

seekcamera_get_core_part_number

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_core_part_number(
    seekcamera_t* camera,
    seekcamera_core_part_number_t* core_part_number);
```

Overview

Gets the core part number (CPN) of the camera.

Parameters

- `camera`
  - Camera for which to get the CPN.
- `core_part_number`
  - CPN of the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_core_part_number_t

seekcamera_get_firmware_version

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_firmware_version(
    seekcamera_t* camera,
    seekcamera_firmware_version_t* version);
```

Overview

Gets the firmware version of the camera.

Parameters

- `camera`
  - Camera for which to get the firmware version.
- `version`
  - Firmware version of the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_firmware_version_t

seekcamera_get_thermography_window

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_thermography_window(
    seekcamera_t* camera,
    size_t* x0,
    size_t* y0,
    size_t* w,
    size_t* h);
```

Overview

Gets the thermography window of the camera. The thermography window is expressed in image coordinates. The global origin is the upper-left corner of the frame. Data outside the window is invalid.

Parameters

- `camera`
  - Camera for which to get the thermography window.
- `x0`
  - X-coordinate, in pixels, of the thermography window origin. It is an output parameter.
- `y0`
  - Y-coordinate, in pixels, of the thermography window origin. It is an output parameter.
- `w`
  - Width, in pixels, of the thermography window. It is an output parameter.
- `h`
  - Height, in pixels, of the thermography window. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_thermography_window

seekcamera_set_thermography_window

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_thermography_window(
    seekcamera_t* camera,
    size_t x0,
    size_t y0,
    size_t w,
    size_t h);
```

Overview

Sets the thermography window of the camera. The thermography window is expressed in image coordinates. The global origin is the upper-left corner of the frame. Data outside the window is invalid.

Parameters

- `camera`
  - Camera for which to set the thermography window.
- `x0`
  - X-coordinate, in pixels, of the thermography window origin.
- `y0`
  - Y-coordinate, in pixels, of the thermography window origin.
- `w`
  - Width, in pixels, of the thermography window.
- `h`
  - Height, in pixels, of the thermography window.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_thermography_window

seekcamera_update_firmware

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_update_firmware(
    seekcamera_t* camera,
    const char* update_file,
    seekcamera_memory_access_callback_t callback,
    void* user_data);
```

Overview

Updates the camera firmware using an input firmware file on the host OS. An optional callback can be used to provide progress updates.

WARNING	This function should not be called when a capture session is live.

NOTE	This function is not supported, and is not neccessary, for Micro Core SPI cores.

Parameters

- `camera`
  - Camera for which to update the firmware.
- `update_file`
  - Path to the firmware update file. It may be absolute or relative to the executable. It must exist on the host filesystem.
- `callback`
  - Optional memory access callback that provides progress updates.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_memory_access_callback_t

seekcamera_store_calibration_data

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_store_calibration_data(
    seekcamera_t* camera,
    const char* source_dir,
    seekcamera_memory_access_callback_t callback,
    void* user_data);
```

Overview

Stores calibration data and pairs the camera. An optional callback can be used to provide progress updates. Pairing refers to the process by which the sensor is associated with the host and the embedded processor. The pairing process and requirements are highly dependent on the characteristics of the camera and system.

WARNING	This function should not be called when a capture session is live. Pairing commands are only applicable to Micro Cores.

Parameters

- `camera`
  - Camera for which to upload calibration data.
- `source_dir`
  - If NULL, the calibration data is read from sensor flash. If non NULL, the calibration data is read from the host filesystem.
- `callback`
  - Optional memory access callback that provides progress updates.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_memory_access_callback_t

`seekcamera_store_flat_scene_correction`

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_store_flat_scene_correction(
    seekcamera_t* camera,
    seekcamera_flat_scene_correction_id_t id,
    seekcamera_memory_access_callback_t callback,
    void* user_data);
```

Overview

Stores a flat scene correction (FSC). The FSC is always stored to the host filesystem; it is also stored to the device if supported. An optional callback can be used to provide progress updates. Flat scene correction refers to the procedure used to correct non-uniformity in the thermal image introduced by the OEMs manufacturing process. See the FSC application note on the Seek Thermal Developers Portal for more information.

NOTE

1. The camera must start imaging to compute and store a FSC.
2. The camera must stop imaging to persistently save a FSC.

Parameters

- `camera`
  - Camera for which to compute and store the FSC.
- `id`
  - Unique enumerated ID of the FSC.
- `callback`
  - Optional memory access callback that provides progress updates.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_flat_scene_correction_id_t
- seekcamera_delete_flat_scene_correction
- seekcamera_memory_access_callback_t

seekcamera_delete_flat_scene_correction

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_delete_flat_scene_correction(
    seekcamera_t* camera,
    seekcamera_flat_scene_correction_id_t id,
    seekcamera_memory_access_callback_t callback,
    void* user_data);
```

Overview

Deletes a flat scene correction (FSC). The FSC will be deleted from any location it was stored. An optional callback can be used to provide progress updates. See the FSC application note on the Seek Thermal Developers Portal for more information.

NOTE === The camera must not be imaging to delete a FSC. ===

Parameters

- `camera`
  - Camera for which to delete the FSC.

- `id`
  - Unique enumerated ID of the FSC.
- `callback`
  - Optional memory access callback that provides progress updates.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_flat_scene_correction_id_t
- seekcamera_store_flat_scene_correction
- seekcamera_memory_access_callback_t

seekcamera_load_app_resources

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_load_app_resources(
    seekcamera_t* camera,
    seekcamera_app_resources_region_t region,
    void* data,
    size_t data_size,
    seekcamera_memory_access_callback_t callback,
    void* user_data);
```

Overview

Loads application resources into host memory. The source region may either be the camera internal memory or the SDK internal cache. Resources in each region must be ⇐ 64KB. An optional callback can be used to provide progress updates.

WARNING	This function should not be called when a capture session is live.

Parameters

- `camera`
  - Camera for which to load the application resources.
- `region`
  - Enumerated value indicating which region should be used to load the data.
- `data`
  - Pointer to host memory. It is where the device data will be loaded. It must point to valid

memory.

  - `data_size`
  - Total size of the buffer pointed to by `data`. It must be ⇐ 64KB.

- `callback`
  - Optional memory access callback that provides progress updates.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_app_resources_t
- seekcamera_memory_access_callback_t

`seekcamera_store_app_resources`

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_store_app_resources(
    seekcamera_t* camera,
    seekcamera_app_resources_region_t region,
    const void* data,
    size_t data_size,
    seekcamera_memory_access_callback_t callback,
    void* user_data);
```

Overview

Stores application resources to either the host or the device. The source region is host memory. The destination region may either be the camera internal memory or the SDK internal cache. Resources in each region must be ⇐ 64KB. The data is persistent. An optional callback can be used to provoide progress updates.

WARNING	This function should not be called when a capture session is live.

Parameters

- `camera`
  - Camera for which to store application resources.
- `region`
  - Enumerated value indicating which region shouldbe used to store the data.
- `data`
  - Pointer to host memory. It contains the user defined data to store. It must point to valid

memory.

- `data_size`
  - Total size of the buffer pointed to by `data`. It must be ⇐ 64KB.
- `callback`
  - Optional memory access callback that provides progress updates.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_app_resources_t
- seekcamera_memory_access_callback_t

seekcamera_capture_session_start

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_capture_session_start(
    seekcamera_t* camera,
    uint32_t frame_format);
```

Overview

Begins streaming frames of the specified output formats from the camera. Generally a frame available callback should be pre-registered in order to receive frames, but doing so is not required.

WARNING

Camera functions that interact with flash storage should not be called when a capture session is live.

Unsafe functions
seekcamera_update_firmware
seekcamera_store_calibration_data
seekcamera_delete_flat_scene_correction
seekcamera_load_app_resources
seekcamera_store_app_resources

Parameters

- `camera`
  - Camera for which to create the capture session.

- `frame_format`
    - Bitwise OR combination of the frame formats to output. The frame format types are specified by seekcamera_frame_format_t.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_capture_session_stop
- seekcamera_frame_available_callback_t
- seekcamera_frame_format_t

seekcamera_capture_session_stop

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_capture_session_stop(
    seekcamera_t* camera);
```

Overview

Stops streaming frames from the camera.

Parameters

- `camera`
    - Camera for which to stop the capture session.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_capture_session_start

seekcamera_register_frame_available_callback

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_register_frame_available_callback(
    seekcamera_t* camera,
    seekcamera_frame_available_callback_t callback,
    void* user_data);
```

Overview

Registers a user-defined frame available callback function with the camera. The callback is fired

every time a new frame is available. There can only be one registered frame callback at a time.

Parameters

- `camera`
  - Camera for which to register the frame available callback.
- `callback`
  - Callback function to register with the camera.
- `user_data`
  - Optional parameter pointing to user defined data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_frame_available_callback_t

seekcamera_get_color_palette

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_color_palette(
    seekcamera_t* camera,
    seekcamera_color_palette_t* palette);
```

Overview

Gets the active color palette. Color palettes are used to colorize the image. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to get the color palette.
- `palette`
  - Active color palette used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_color_palette_t
- seekcamera_set_color_palette

seekcamera_set_color_palette

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_color_palette(
    seekcamera_t* camera,
    seekcamera_color_palette_t palette);
```

Overview

Sets the active color palette. Color palettes are used to colorize the image. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to set the color palette.
- `palette`
  - New active color palette for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_color_palette_t
- seekcamera_get_color_palette

seekcamera_set_color_palette_data

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_color_palette_data(
    seekcamera_t* camera,
    seekcamera_color_palette_t palette,
    seekcamera_color_palette_data_t* palette_data);
```

Overview

Sets the color palette data for a particular color palette. Setting color palette data is only valid for user palettes. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to set the color palette data.
- `palette`
  - Enumerated type corresponding to the color palette for which to set the data.

- `palette_data`
- Color values used to colorize the thermal image.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_color_palette_t
- seekcamera_color_palette_data_t

seekcamera_get_agc_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_agc_mode(
    seekcamera_t* camera,
    seekcamera_agc_mode_t* mode);
```

Overview

Gets the active AGC mode. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `mode`
  - Active AGC mode used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_agc_mode_t
- seekcamera_set_agc_mode

seekcamera_set_agc_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_agc_mode(
    seekcamera_t* camera,
    seekcamera_agc_mode_t mode);
```

Overview

Sets the AGC mode. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `mode`
  - New active AGC mode for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_agc_mode_t
- seekcamera_get_agc_mode

`seekcamera_get_histeq_agc_plateau`

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_plateau(
    seekcamera_t* camera,
    float* plateau);
```

Overview

Gets the plateau value used for HistEQ AGC. Plateau value limits the percentage of pixels that can be in a single histogram bin. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `plateau`
  - Active plateau value used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_plateau

seekcamera_set_histeq_agc_plateau

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_plateau(
    seekcamera_t* camera,
    float plateau);
```

Overview

Sets the plateau value used for HistEQ AGC. Plateau value limits the percentage of pixels that can be in a single histogram bin. It may take on values in the closed interval [0, 1] with floating point precision. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `plateau`
  - New active plateau value for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_histeq_agc_plateau

seekcamera_get_histeq_agc_plateau_redistribution_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t
seekcamera_get_histeq_agc_plateau_redistribution_mode(
    seekcamera_t* camera,
    seekcamera_histeq_agc_plateau_redistribution_mode_t* mode);
```

Overview

Gets the plateau redistribution mode used for HistEQ AGC. The plateau value limits the percentage of pixels that can be in a single histogram bin. The plateau redistribution mode controls how pixels in a histogram bin that exceed the plateau value are redistributed to other bins in the histogram. Settings are refreshed between frames.

NOTE	This value only affects SEEKCAMERA_AGC_MODE_HISTEQ.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `mode`
  - Active plateau redistribution mode used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_plateau
- seekcamera_histeq_agc_plateau_redistribution_mode_t

`seekcamera_set_histeq_agc_plateau_redistribution_mode`

Declaration

```
SEEKCAMERA_API seekcamera_error_t
seekcamera_set_histeq_agc_plateau_redistribution_mode(
    seekcamera_t* camera,
    seekcamera_histeq_agc_plateau_redistribution_mode_t mode);
```

Overview

Sets the plateau redistribution mode used for HistEQ AGC. The plateau value limits the percentage of pixels that can be in a single histogram bin. The plateau redistribution mode controls how pixels in a histogram bin that exceed the plateau value are redistributed to other bins in the histogram. Settings are refreshed between frames.

NOTE	This value only affects SEEKCAMERA_AGC_MODE_HISTEQ.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `mode`
  - Plateau redistribution mode to be used by the camera.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_plateau

- seekcamera_histeq_agc_plateau_redistribution_mode_t

seekcamera_get_histeq_agc_gain_limit

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_gain_limit(
    seekcamera_t* camera,
    float* limit);
```

Overview

Gets the maximum gain limit value used for HistEQ AGC. Gain limit constrains the number of output bits assigned to a single bin. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `limit`
  - Active maximum gain limit value used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_gain_limit

seekcamera_set_histeq_agc_gain_limit

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_gain_limit(
    seekcamera_t* camera,
    float limit);
```

Overview

Sets the maximum gain limit value used for HistEQ AGC. Gain limit constrains the number of output bits assigned to a single bin. It may take on values in the closed interval [0, 256] with floating point precision. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `limit`
  - New active maximum gain limit value for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_histeq_agc_gain_limit

seekcamera_get_histeq_agc_gain_limit_factor_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_gain_limit_factor_mode(
    seekcamera_t* camera,
    seekcamera_histeq_agc_gain_limit_factor_mode_t* mode);
```

Overview

Gets the gain limit factor mode for HistEQ AGC. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC.

Parameters

- `camera`
  - Camera for which to set the gain limit factor mode.
- `mode`
  - Active HistEQ AGC gain limit factor mode used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_histeq_agc_gain_limit_factor_mode_t
- seekcamera_set_histeq_agc_gain_limit_factor_mode

seekcamera_set_histeq_agc_gain_limit_factor_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_gain_limit_factor_mode(
    seekcamera_t* camera,
    seekcamera_histeq_agc_gain_limit_factor_mode_t mode);
```

Overview

Sets the gain limit factor mode for HistEQ AGC. Settings are refreshed between frames.

> **NOTE** This value has effect when using HistEQ AGC.

Parameters

- `camera`
  - Camera for which to set the gain limit factor mode.
- `mode`
  - New active HistEQ AGC gain limit factor mode for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_histeq_agc_gain_limit_factor_mode_t
- seekcamera_get_histeq_agc_gain_limit_factor_mode

seekcamera_get_histeq_agc_gain_limit_factor_ymin

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_gain_limit_factor_ymin(
    seekcamera_t* camera,
    float* ymin);
```

Overview

Gets the gain limit factor ymin value used for HistEQ AGC. It is the minimum value of the gain limit factor. Settings are refreshed between frames.

> **NOTE** This value has effect when using HistEQ AGC. It has effect in Manual gain limit factor mode.

Parameters

- `camera`

    - Camera for which to set the AGC mode.
- `ymin`
    - Active gain limit factor ymin value used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_gain_limit_factor_ymin

seekcamera_set_histeq_agc_gain_limit_factor_ymin

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_gain_limit_factor_ymin(
    seekcamera_t* camera,
    float ymin);
```

Overview

Sets the gain limit factor ymin value used for HistEQ AGC. It may take on values in the closed interval [0, 1.0] with floating point precision. It is the minimum value of the gain limit factor. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in Manual gain limit factor mode.

Parameters

- `camera`
    - Camera for which to set the AGC mode.
- `ymin`
    - New active gain limit factor ymin for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_histeq_agc_gain_limit_factor_ymin

seekcamera_get_histeq_agc_gain_limit_factor_xmax

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_gain_limit_factor_xmax(
    seekcamera_t* camera,
    uint32_t* xmax);
```

Overview

Gets the gain limit factor xmax value used for HistEQ AGC. It is the width of the histogram at which the gain limit factor will be applied. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in Manual gain limit factor mode.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `xmax`
  - Active gain limit factor xmax used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_gain_limit_factor_xmax

seekcamera_set_histeq_agc_gain_limit_factor_xmax

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_gain_limit_factor_xmax(
    seekcamera_t* camera,
    uint32_t xmax);
```

Overview

Sets the gain limit factor xmax value used for HistEQ AGC. It may take on values in the closed interval [0, 65535] with integer precision. It is the width of the histogram at which the gain limit factor will be applied. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in Manual gain limit factor mode.

Parameters

- `camera`
  - Camera for which to set the AGC mode.

- xmax
  - New active gain limit factor xmax for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_histeq_agc_gain_limit_factor_xmax

seekcamera_get_histeq_agc_alpha_time_seconds

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_alpha_time_seconds(
    seekcamera_t* camera,
    float* time);
```

Overview

Gets the alpha time parameter used for HistEQ AGC. Alpha time is used to blend the current frame histogram with the previous frame histogram. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- camera
  - Camera for which to set the AGC mode.
- time
  - Active alpha time value used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_alpha_time_seconds

seekcamera_set_histeq_agc_alpha_time_seconds

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_alpha_time_seconds(
    seekcamera_t* camera,
    float time);
```

Overview

Sets the alpha time parameter used for HistEQ AGC. Alpha time is used to blend the current frame histogram with the previous frame histogram. It may take on values in the closed interval [0.0, 6.0] with floating point precision. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `time`
  - New active alpha time value for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_histeq_agc_alpha_time_seconds

seekcamera_get_histeq_agc_trim_left

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_trim_left(
    seekcamera_t* camera,
    float* trim);
```

Overview

Gets the value used to trim the left side of the histogram. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `trim`
  - Active left trim value used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_trim_left
- seekcamera_get_histeq_agc_trim_right

seekcamera_set_histeq_agc_trim_left

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_trim_left(
    seekcamera_t* camera,
    float trim);
```

Overview

Sets the value used to trim the left side of the histogram. It may take on values in the closed interval [0.0, 0.49] with floating point precision. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `trim`
  - New active left trim value for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_histeq_agc_trim_left
- seekcamera_set_histeq_agc_trim_right

seekcamera_get_histeq_agc_trim_right

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_histeq_agc_trim_right(
    seekcamera_t* camera,
    float* trim);
```

Overview

Gets the value used to trim the right side of the histogram. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `trim`
  - Active right trim value used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_histeq_agc_trim_right
- seekcamera_get_histeq_agc_trim_left

seekcamera_set_histeq_agc_trim_right

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_histeq_agc_trim_right(
    seekcamera_t* camera,
    float trim);
```

Overview

Sets the value used to trim the right side of the histogram. It may take on values in the closed interval [0.0, 0.49] with floating point precision. Settings are refreshed between frames.

NOTE	This value has effect when using HistEQ AGC. It has effect in both Auto or Manual gain limit factor modes.

Parameters

- `camera`
  - Camera for which to set the AGC mode.
- `trim`
  - New active right trim value for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t

- seekcamera_get_histeq_agc_trim_right
- seekcamera_set_histeq_agc_trim_left

seekcamera_get_linear_agc_lock_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_linear_agc_lock_mode(
    seekcamera_t* camera,
    seekcamera_linear_agc_lock_mode_t* mode);
```

Overview

Gets the lock mode used by Linear AGC. Settings are refreshed between frames.

> **NOTE** This value has effect when using Linear AGC.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `mode`
  - Active linear AGC lock mode used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_linear_agc_lock_mode_t
- seekcamera_set_linear_agc_lock_mode

seekcamera_set_linear_agc_lock_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_linear_agc_lock_mode(
    seekcamera_t* camera,
    seekcamera_linear_agc_lock_mode_t mode);
```

Overview

Sets the lock mode used by Linear AGC. Settings are refreshed between frames.

> **NOTE** This value has effect when using Linear AGC.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `mode`
  - New active linear AGC lock mode for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_linear_agc_lock_mode_t
- seekcamera_get_linear_agc_lock_mode

seekcamera_get_linear_agc_lock_min

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_linear_agc_lock_min(
    seekcamera_t* camera,
    uint32_t* lock_min);
```

Overview

Gets the value used as the minimum lock by Linear AGC. Settings are refreshed between frames.

NOTE	This value has effect when using Linear AGC. It has effect in both select semi-Manual and Manual lock modes.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `lock_min`
  - Active linear AGC lock minimum used by the camera. It is specified in counts. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_linear_agc_lock_min
- seekcamera_get_linear_agc_lock_max

seekcamera_set_linear_agc_lock_min

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_linear_agc_lock_min(
    seekcamera_t* camera,
    uint32_t lock_min);
```

Overview

Sets the value used as the minimum lock by Linear AGC. It may take on values in the closed interval [0, 65535] with integer precision. Settings are refreshed between frames.

> **NOTE** This value has effect when using Linear AGC. It has effect in both select semi-Manual and Manual lock modes.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `lock_min`
  - New active linear AGC lock minimum for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_linear_agc_lock_min_t
- seekcamera_get_linear_agc_lock_min
- seekcamera_set_linear_agc_lock_max

seekcamera_get_linear_agc_lock_max

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_linear_agc_lock_max(
    seekcamera_t* camera,
    uint32_t* lock_max);
```

Overview

Gets the value used as the maximum lock by Linear AGC. Settings are refreshed between frames.

> **NOTE** This value has effect when using Linear AGC. It has effect in both select semi-Manual and Manual lock modes.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `lock_max`
  - Active linear AGC lock maximum used by the camera. It is specified in counts. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_linear_agc_lock_max
- seekcamera_get_linear_agc_lock_max

seekcamera_set_linear_agc_lock_max

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_linear_agc_lock_max(
    seekcamera_t* camera,
    uint32_t lock_max);
```

Overview

Sets the value used as the maximum lock by Linear AGC. It may take on values in the closed interval [0, 65535] with integer precision. Settings are refreshed between frames.

NOTE	This value has effect when using Linear AGC. It has effect in both select semi-Manual and Manual lock modes.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `lock_max`
  - New active linear AGC lock maximum for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_linear_agc_lock_max
- seekcamera_set_linear_agc_lock_max

seekcamera_get_shutter_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_shutter_mode(
    seekcamera_t* camera,
    seekcamera_shutter_mode_t* mode);
```

Overview

Gets the active shutter mode. Settings are refreshed between frames.

WARNING	Shutter commands are only applicable to Mosaic Cores.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `mode`
  - Active shutter mode used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_shutter_mode_t
- seekcamera_set_shutter_mode

seekcamera_set_shutter_mode

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_shutter_mode(
    seekcamera_t* camera,
    seekcamera_shutter_mode_t mode);
```

Overview

Sets the shutter mode. Settings are refreshed between frames.

WARNING	Shutter commands are only applicable to Mosaic Cores.

Parameters

- `camera`
  - Camera for which to set the shutter mode.
- `mode`

- New active shutter mode for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_shutter_mode_t
- seekcamera_get_shutter_mode

seekcamera_get_temperature_unit

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_temperature_unit(
    seekcamera_t* camera,
    seekcamera_temperature_unit_t* unit);
```

Overview

Gets the active temperature unit. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to get the temperature unit.
- `unit`
  - Active temperature unit used by the camera. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_temperature_unit_t
- seekcamera_set_temperature_unit

seekcamera_set_temperature_unit

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_temperature_unit(
    seekcamera_t* camera,
    seekcamera_temperature_unit_t unit);
```

Overview

Sets the active temperature unit. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to set the temperature unit.
- `unit`
  - Temperature unit to be used by the camera.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_temperature_unit_t
- seekcamera_get_temperature_unit

seekcamera_shutter_trigger

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_shutter_trigger(
    seekcamera_t* camera);
```

Overview

Triggers the camera to shutter as soon as possible.

WARNING	Shutter commands are only applicable to Mosaic Cores.

Parameters

- `camera`
  - Camera for which to issue a shutter request.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_shutter_mode_t

seekcamera_get_scene_emissivity

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_scene_emissivity(
    seekcamera_t* camera,
    float* emissivity);
```

Overview

Gets the global scene emissivity. Emissivity is the measure of an objects ability to emit thermal radiation. It may take on values in the closed interval [0,1] with floating point precision. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to get the scene emissivity.
- `emissivity`
  - Active scene emissivity that was set by the user. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_scene_emissivity

`seekcamera_set_scene_emissivity`

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_scene_emissivity(
    seekcamera_t* camera,
    float emissivity);
```

Overview

Sets the global scene emissivity. Emissivity is the measure of an objects ability to emit thermal radiation. It may take on values in the closed interval [0,1] with floating point precision. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to set the scene emissivity.
- `emissivity`
  - New scene emissivity for the camera to use.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_scene_emissivity

seekcamera_get_thermography_offset

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_thermography_offset(
    seekcamera_t* camera,
    float* offset);
```

Overview

Gets the thermography offset. The thermography offset is a constant that is applied to every pixel in the thermography frame. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to get the thermography offset.
- `offset`
  - Active thermography offset applied to every pixel in the thermography frame.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_set_thermography_offset

seekcamera_set_thermography_offset

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_thermography_offset(
    seekcamera_t* camera,
    float offset);
```

Overview

Sets the thermography offset. The thermography offset is a constant that is applied to every pixel in the thermography frame. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to get the thermography offset.
- `offset`
  - Thermography offset to be applied to every pixel in the thermography frame.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_get_thermography_offset

seekcamera_set_filter_state

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_set_filter_state(
    seekcamera_t* camera,
    seekcamera_filter_t filter,
    seekcamera_filter_state_t state);
```

Overview

Sets the state of an image processing filter. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to set the filter state.
- `filter`
  - Enumerated type indicating the filter for which to set the state.
- `state`
  - Enumerated type indicating the state of the filter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_filter_t
- seekcamera_filter_state_t
- seekcamera_get_filter_state

seekcamera_get_filter_state

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_get_filter_state(
    seekcamera_t* camera,
    seekcamera_filter_t filter,
    seekcamera_filter_state_t* state);
```

Overview

Gets the state of an image processing filter. Settings are refreshed between frames.

Parameters

- `camera`
  - Camera for which to get the filter state.
- `filter`
  - Enumerated type indicating the filter for which to get the state.
- `state`
  - Enumerated type indicating the state of the filter. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_t
- seekcamera_filter_t
- seekcamera_filter_state_t
- seekcamera_set_filter_state

Seek Camera Frame API

The Seek Camera Frame API is an encapsulation layer that manages frames delivered via Seek camera devices. It is a part of the Seek Camera API interface and makes use of the generic Seek Frame API interface.

`seekcamera_frame_t`

Declaration

```
typedef struct seekcamera_frame_t seekcamera_frame_t;
```

Overview

Structure that encapsulates single or multiple generic seekframe_t structs.

`seekcamera_frame_format_t`

Declaration

```
typedef enum seekcamera_frame_format_t
{
    ...
} seekcamera_frame_format_t;
```

Overview

Types of output frame formats. Multiple frame formats can be captured simultaneously. However only one Display frame format and one Thermography frame format can exist in a capture session.

The exception is that grayscale can be captured along with another color Display format.

Please refer to the frame types overview section for more information on Display, Thermography, and Partially Processed frame types.

NOTE	All format types are little endian byte order.

Cases

- `SEEKCAMERA_FRAME_FORMAT_CORRECTED`
  - Corrected frame data format. It is the least processed format offerred. To output corrected data, the SDK performs only required processing steps. These include: flat field subtraction (for shuttered cores), gain and offset correction, bad pixel replacement, and a few other proprietary processing techniques.

Channels	Pixel Depth	Pixel Padding	Line Padding
1	16	0	0

- `SEEKCAMERA_FRAME_FORMAT_PRE_AGC`
  - Pre-AGC frame data format. It is the second least processed format offered. To output pre-AGC data, the SDK performs all processing steps required to output corrected data, as well as a limited number of proprietary processing filters.

Channels	Pixel Depth	Pixel Padding	Line Padding
1	16	0	0

- `SEEKCAMERA_FRAME_FORMAT_THERMOGRAPHY_FLOAT`
  - Thermography floating point frame data format. To output floating point thermography data, the SDK performs all processing steps required to output corrected and pre-AGC data, as well as proprietary thermography processing. Temperature units are in degrees Celsius.

Channels	Pixel Depth	Pixel Padding	Line Padding
1	32	0	0

- `SEEKCAMERA_FRAME_FORMAT_THERMOGRAPHY_FIXED_10_6`
  - Thermography fixed point (U10.6) frame data format. To output fixed point thermography data, the SDK performs all processing steps required to output corrected and pre-AGC data, as well as proprietary thermography processing.

To get temperature

```
temperature = (counts / 64) - 40
```

Channels	Pixel Depth	Pixel Padding	Line Padding
1	16	0	0

- SEEKCAMERA_FRAME_FORMAT_GRAYSCALE
  - Grayscale frame data format. To output grayscale data, the SDK performs all processing steps required to output corrected and pre-AGC data, as well as proprietary automatic gain control (AGC) processing.

Channels	Pixel Depth	Pixel Padding	Line Padding
1	8	0	0

- SEEKCAMERA_FRAME_FORMAT_COLOR_ARGB8888
  - Color ARGB8888 (BGRA32) frame data format. To output ARGB8888 display data, the SDK performs all processing steps required to output corrected and pre-AGC data, as well as proprietary image processing.

Channels	Pixel Depth	Pixel Padding	Line Padding
4	32	0	0

- SEEKCAMERA_FRAME_FORMAT_COLOR_RGB565
  - Color RGB565 frame data format. To output RGB565 display data, the SDK performs all processing steps required to output corrected and pre-AGC data, as well as proprietary image processing.

Channels	Pixel Depth	Pixel Padding	Line Padding
3	16	0	0

- SEEKCAMERA_FRAME_FORMAT_COLOR_AYUV
  - Color AYUV frame data format. To output AYUV display data, the SDK performs all processing steps required to output corrected and pre-AGC data, as well as proprietary image processing.

Channels	Pixel Depth	Pixel Padding	Line Padding
4	32	0	0

- SEEKCAMERA_FRAME_FORMAT_COLOR_YUY2
  - Color YUY2 frame data format. To output YUY2 display data, the SDK performs all processing steps required to output corrected and pre-AGC data, as well as proprietary image processing.

Channels	Pixel Depth	Pixel Padding	Line Padding
2	16	0	0

seekcamera_frame_header_t

Declaration

```
typedef struct seekcamera_frame_header_t
{
    ...
} seekcamera_frame_header_t;
```

Overview

Common header for the camera frame. It is a fixed size (2048 bytes) and byte aligned. It can be accessed in each individual seekframe_t struct.

Fields

- `sentinel: uint32_t`
  - Header sentinel
- `version: uint8_t`
  - Version of the frame header
- `type: uint32_t`
  - Enumerated frame type (seekcamera_frame_format_t)
- `width: uint16_t`
  - Number of pixels in horizontal dimension
- `height: uint16_t`
  - Number of pixels in vertical dimension
- `channels: uint8_t`
  - Number of image channels (e.g. 3 for RGB)
- `pixel_depth: uint8_t`
  - Number of bits per pixel
- `pixel_padding: uint8_t`
  - Number of padding bits per pixel
- `line_stride: uint16_t`
  - Number of padding bytes per line
- `line_padding: uint16_t`
  - Number of padding bytes per line row
- `header_size: uint16_t`
  - Number of bytes in the header including line padding
- `timestamp_utc_ns: uint64_t`
  - UTC timestamp in nanosecond resolution
- `chipid: char[16]`
  - CID of the camera (seekcamera_chipid_t)

- `serial_number: char[16]`
  - SN of the camera (seekcamera_serial_number_t)
- `core_part_number: char[32]`
  - CPN of the camera (seekcamera_core_part_number_t)
- `firmware_version: uint8_t[4]`
  - Firmware version (seekcamera_firmware_version_t)
- `io_type: uint8_t`
  - IO type of the camera (seekcamera_io_type_t)
- `fpa_frame_count: uint32_t`
  - Index of the frame as seen by the FPA
- `fpa_diode_count: uint32_t`
  - Uncalibrated sampling of the FPA temperature diode voltage
- `environment_temperature: float`
  - Estimated temperature based on the FPA in degrees Celsius
- `thermography_min_x: uint16_t`
  - Image coordinate (x-dimension) of the min thermography pixel
- `thermography_min_y: uint16_t`
  - Image coordinate (y-dimension) of the min thermography pixel
- `thermography_min_value: float`
  - Value of the min thermography pixel
- `thermography_max_x: uint16_t`
  - Image coordinate (x-dimension) of the max thermography pixel
- `thermography_max_y: uint16_t`
  - Image coordinate (y-dimension) of the max thermography pixel
- `thermography_max_value: float`
  - Value of the max thermography pixel
- `thermography_spot_x: uint16_t`
  - Image coordinate (x-dimension) of the 'spot' thermography pixel
- `thermography_spot_y: uint16_t`
  - Image coordinate (y-dimension) of the 'spot' thermography pixel
- `thermography_spot_value: float`
  - Value of the 'spot' thermography pixel
- `agc_mode: uint8_t`
  - AGC mode used to process the image (seekcamera_agc_mode_t)
- `histeq_agc_num_bins: uint16_t`

    - Number of bins in the HistEQ AGC histogram.
- `histeq_agc_bin_width: uint16_t`
    - Number of counts per bin in the HistEQ AGC histogram.
- `histeq_agc_gain_limit_factor: float`
    - Multiplier of the HistEQ gain limit.
- `histeq_agc_reserved: uint8_t[64]`
    - Reserved bytes for future HistEQ AGC metrics.
- `linear_agc_min: uint32_t`
    - Minimum count value in the frame when using Linear AGC.
- `linear_agc_max: uint32_t`
    - Maximum count value in the frame when using Linear AGC.
- `linear_agc_reserved: uint8_t[32`
    - Reserved bytes for future Linear AGC metrics.
- `gradient_correction_filter_state: uint8_t`
    - State of the gradient correction filter (seekcamera_filter_state_t)
- `flat_scene_correction_filter_state: uint8_t`
    - State of the flat scene correction filter (seekcamera_filter_state_t)
- `reserved: uint8_t[1798]`
    - Reserved bytes to guarantee this struct is 2048 bytes

See also

- seekframe_t
- seekframe_get_header
- seekframe_get_header_size

seekcamera_frame_get_frame_by_format

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_frame_get_frame_by_format(
    const seekcamera_frame_t* camera_frame,
    seekcamera_frame_format_t format,
    seekframe_t** frame);
```

Overview

Gets an individual frame from the camera frame structure according to format.

NOTE	The format must have been specified in the capture session flags.

Parameters

- `camera_frame`
  - Camera frame wrapper for which to get the generic frame data.
- `format`
  - Format of the frame to retrieve.
- `frame`
  - Pointer to the generic frame storage. It should be NULL on input. If successful, it will be non-NULL and point to valid data on output; otherwise it will remain NULL. It is an output parameter.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_frame_t
- seekcamera_frame_format_t
- seekframe_t

`seekcamera_frame_lock`

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_frame_lock(
    seekcamera_frame_t* camera_frame);
```

Overview

Locks a camera frame for exclusive access to the underlying frame data. This allows the application to safely access the frame outside the context of the frame callback.

NOTE

1. There is no need to use this API (or the corresponding unlock) if all processing of the frame and frame data is done in the context of the frame callback.
2. If the camera frame is already locked, calling this API multiple times has no effect.
3. The user is responsible for unlocking a camera frame when exclusive access to the frame data is no longer needed.
4. If seekcamera_capture_session_stop is called while the camera frame is locked, exclusive access to the frame data is no longer guaranteed.

Parameters

- `camera_frame`
  - Camera frame wrapper for the generic frame data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_frame_t

seekcamera_frame_unlock

Declaration

```
SEEKCAMERA_API seekcamera_error_t seekcamera_frame_unlock(
    seekcamera_frame_t* camera_frame);
```

Overview

Unlocks a camera frame that was previously locked by seekcamera_frame_lock.

NOTE

1. This API should only be used outside the context of the frame callback. There is no need for this API (or the corresponding lock API) if all processing of the frame (and frame data) is done in the context of the frame callback.
2. If the camera frame is already unlocked, calling this API multiple times has no effect.

Parameters

- `camera_frame`
  - Camera frame wrapper for the generic frame data.

Return value

An enumerated error code indicating pass or fail.

See also

- seekcamera_frame_t

Seek Camera Error API

The Seek Camera Error API is a error handling interface which the rest of the Seek Camera interfaces uses for error handling.

seekcamera_error_t

Declaration

```
typedef enum seekcamera_error_t
{
    ...
} seekcamera_error_t;
```

Overview

Enumerated type representing types of events used by the camera manager.

Cases

- `SEEKCAMERA_SUCCESS`
  - Status that is returned when there is no error.
- `SEEKCAMERA_ERROR_DEVICE_COMMUNICATION`
  - Status that is returned when device communication fails.
- `SEEKCAMERA_ERROR_INVALID_PARAMETER`
  - Status that is returned when an invalid parameter is received.
- `SEEKCAMERA_ERROR_PERMISSIONS`
  - Status that is returned when there are insufficient permissions to access a resource.
- `SEEKCAMERA_ERROR_NO_DEVICE`
  - Status that is returned where there is no device.
- `SEEKCAMERA_ERROR_DEVICE_NOT_FOUND`
  - Status that is returned when no device is found.
- `SEEKCAMERA_ERROR_DEVICE_BUSY`
  - Status that is returned when the device is busy.
- `SEEKCAMERA_ERROR_TIMEOUT`
  - Status that is returned when a device timeout occurs.
- `SEEKCAMERA_ERROR_OVERFLOW`
  - Status that is returned when overflow is detected.
- `SEEKCAMERA_ERROR_UNKNOWN_REQUEST`
  - Status that is returned when an unknown request is made.
- `SEEKCAMERA_ERROR_INTERRUPTED`
  - Status taht is returned when an operation is interrupted.
- `SEEKCAMERA_ERROR_OUT_OF_MEMORY`
  - Status that is returned when the system is out of memory.
- `SEEKCAMERA_ERROR_NOT_SUPPORTED`
  - Status that is returned when an operation is not supported.
- `SEEKCAMERA_ERROR_OTHER`
  - Status that is return when the source of the error is unknown.
- `SEEKCAMERA_ERROR_CANNOT_PERFORM_REQUEST`
  - Status that is returned when the request cannot be performed.
- `SEEKCAMERA_ERROR_FLASH_ACCESS_FAILURE`
  - Status that is returned when flash access fails.
- `SEEKCAMERA_ERROR_IMPLEMENTATION_ERROR`
  - Status that is returned when there is an implementation error.

- SEEKCAMERA_ERROR_REQUEST_PENDING
  - Status that is returned when a request is already pending.
- SEEKCAMERA_ERROR_INVALID_FIRMWARE_IMAGE
  - Status that is returned when the firmware image is invalid.
- SEEKCAMERA_ERROR_INVALID_KEY
  - Status that is returned when an invalid key is encountered.
- SEEKCAMERA_ERROR_SENSOR_COMMUNICATION
  - Status that is returned when sensor communication fails.
- SEEKCAMERA_ERROR_OUT_OF_RANGE
  - Status that is returned when a value is out of range.
- SEEKCAMERA_ERROR_VERIFY_FAILED
  - Status that is returned verification fails.
- SEEKCAMERA_ERROR_SYSCALL_FAILED
  - Status that is returned when a system call fails.
- SEEKCAMERA_ERROR_FILE_DOES_NOT_EXIST
  - Status that is returned when a file does not exist.
- SEEKCAMERA_ERROR_DIRECTORY_DOES_NOT_EXIST
  - Status that is returned when a directory does not exist.
- SEEKCAMERA_ERROR_FILE_READ_FAILED
  - Status that is returned when a file read fails.
- SEEKCAMERA_ERROR_FILE_WRITE_FAILED
  - Status that is returned when a file write fails.
- SEEKCAMERA_ERROR_NOT_IMPLEMENTED
  - Status that is returned when a method is not implemented.
- SEEKCAMERA_ERROR_NOT_PAIRED
  - Status that is returned when a camera connects in an unpaired state.

See also

- seekcamera_error_get_str

seekcamera_error_get_str

Declaration

```
SEEKCAMERA_API const char* seekcamera_error_get_str(
    seekcamera_error_t status);
```

Overview

Gets the description of a camera error type. The error strings are NULL terminated C strings.

Parameters

- `status`
  - An enumerated error code indicating pass or fail.

Return value

The description of a camera error type.

See also

- seekcamera_error_t

Seek Camera Version API

The Seek Camera Version API is an interface used by the SDK and user applications to query the Seek Camera API version. The version number scheme is [MAJOR].[MINOR].[PATCH].[INTERNAL]_[QUALIFIER].

`seekcamera_version_get_major`

Declaration

```
SEEKCAMERA_API uint32_t seekcamera_version_get_major();
```

Overview

Returns the major version number of the seekcamera API.

Return value

The major version number of the seekcamera API.

See also

- seekcamera_version_get_minor
- seekcamera_version_get_patch
- seekcamera_version_get_internal
- seekcamera_version_get_qualifier

`seekcamera_version_get_minor`

Declaration

```
SEEKCAMERA_API uint32_t seekcamera_version_get_minor();
```

Overview

Returns the minor version number of the seekcamera API.

Return value

The minor version number of the seekcamera API.

See also

- seekcamera_version_get_major
- seekcamera_version_get_patch
- seekcamera_version_get_internal
- seekcamera_version_get_qualifier

seekcamera_version_get_patch

Declaration

```
SEEKCAMERA_API uint32_t seekcamera_version_get_patch();
```

Overview

Returns the patch version number of the seekcamera API.

Return value

The patch version number of the seekcamera API.

See also

- seekcamera_version_get_major
- seekcamera_version_get_minor
- seekcamera_version_get_internal
- seekcamera_version_get_qualifier

seekcamera_version_get_internal

Declaration

```
SEEKCAMERA_API uint32_t seekcamera_version_get_internal();
```

Overview

Returns the internal version number of the seekcamera API.

Return value

The internal version number of the seekcamera API.

See also

- seekcamera_version_get_major
- seekcamera_version_get_minor
- seekcamera_version_get_patch
- seekcamera_version_get_qualifier

seekcamera_version_get_qualifier

Declaration

```
SEEKCAMERA_API const char* seekcamera_version_get_qualifier();
```

Overview

Returns the qualifier version string of the seekcamera API.

Return value

The qualifier version string of the seekcamera API.

See also

- seekcamera_version_get_major
- seekcamera_version_get_minor
- seekcamera_version_get_patch
- seekcamera_version_get_internal

Seek Frame API

The Seek Frame API is an generic frame representation interface used to represent all Seek frames.

seekframe_t

Declaration

```
typedef struct seekframe_t seekframe_t;
```

Overview

Structure that represents an arbitrary frame.

seekframe_get_width

Declaration

```
SEEKFRAME_API size_t seekframe_get_width(
    const seekframe_t* frame);
```

Overview

Gets the width of the frame in image coordinates.

Parameters

- frame
    - Frame for which to get the width.

Return value

On success, the frame width as an unsigned integer type. On failure, 0.

See also

- seekframe_t

seekframe_get_height

Declaration

```
SEEKFRAME_API size_t seekframe_get_height(
    const seekframe_t* frame);
```

Overview

Gets the height of the frame in image coordinates.

Parameters

- `frame`
  - Frame for which to get the height.

Return value

On success, the frame height as an unsigned integer type. On failure, 0.

See also

- seekframe_t

seekframe_get_channels

Declaration

```
SEEKFRAME_API size_t seekframe_get_channels(
    const seekframe_t* frame);
```

Overview

Gets the number of image channels of the frame. Channels are stored contiguously.

Parameters

- `frame`
  - Frame for which to get the number of channels.

Return value

On success, the number of channels as an unsigned integer type. On failure, 0.

See also

- seekframe_t

seekframe_get_pixel_depth

Declaration

```
SEEKFRAME_API size_t seekframe_get_pixel_depth(
    const seekframe_t* frame);
```

Overview

Gets the pixel depth of the frame in bits. Pixel depth refers to the non-padded bit depth of each pixel.

Parameters

- `frame`
  - Frame for which to get the pixel depth.

Return value

On success, the pixel depth as an unsigned integer type. On failure, 0.

See also

- seekframe_t
- seekframe_get_pixel_padding

seekframe_get_pixel_padding

Declaration

```
SEEKFRAME_API size_t seekframe_get_pixel_padding(
    const seekframe_t* frame);
```

Overview

Gets the pixel padding of the frame in bits. Pixel padding refers padding stored between pixels.

Parameters

- `frame`
  - Frame for which to get the pixel padding.

Return value

On success, the pixel padding as an unsigned integer type. On failure, 0.

See also

- seekframe_t
- seekframe_get_pixel_depth

seekframe_get_line_stride

Declaration

```
SEEKFRAME_API size_t seekframe_get_line_stride(
    const seekframe_t* frame);
```

Overview

Gets the line stride of the frame in bytes. Line stride refers to the total width of each image line/row. It includes line padding.

Parameters

- `frame`
    - Frame for which to get the line stride.

Return value

On success, the line stride as an unsigned integer type. On failure, 0.

See also

- seekframe_t
- seekframe_get_line_padding

seekframe_get_line_padding

Declaration

```
SEEKFRAME_API size_t seekframe_get_line_padding(
    const seekframe_t* frame);
```

Overview

Gets the line padding of the frame in bytes. Line padding refers to padding stored at the end of each line/row.

Parameters

- `frame`
    - Frame for which to get the line padding.

Return value

On success, the line padding as an unsigned integer type. On failure, 0.

See also

- seekframe_t
- seekframe_get_line_padding

seekframe_get_data_size

Declaration

```
SEEKFRAME_API size_t seekframe_get_data_size(
    const seekframe_t* frame);
```

Overview

Gets the total size of the frame pixel data in bytes.

Parameters

- `frame`
  - Frame for which to get the data size.

Return value

On success, the data size as an unsigned integer type. On failure, 0.

See also

- seekframe_t
- seekframe_get_data

seekframe_get_data

Declararion

```
SEEKFRAME_API void* seekframe_get_data(
    const seekframe_t* frame);
```

Overview

Gets the pointer to the pixel data of the frame.

Parameters

- `frame`
  - Frame for which to get the pixel data pointer.

Return value

On success, the data pointer as a non NULL pointer. On failure, NULL.

See also

- seekframe_t
- seekframe_get_data_size
- seekframe_get_row
- seekframe_get_pixel

seekframe_get_row

Declarartion

```
SEEKFRAME_API void* seekframe_get_row(
    const seekframe_t* frame,
    size_t y);
```

Overview

Gets the pointer to a row of pixel data of the frame.

Parameters

- `frame`
  - Frame for which to get the pixel data row pointer.

Return value

On success, the row data pointer as a non NULL pointer. On failure, NULL.

See also

- seekframe_t
- seekframe_get_data
- seekframe_get_pixel

seekframe_get_pixel

Declaration

```
SEEKFRAME_API void* seekframe_get_pixel(
    const seekframe_t* frame,
    size_t x,
    size_t y);
```

Overview

Gets the pointer to an individual pixel of the frame.

Parameters

- `frame`
  - Frame for which to get the individual pixel data pointer.

Return value

On success, the individual pixel pointer as a non NULL pointer. On failure, NULL.

See also

- seekframe_t
- seekframe_get_data

- seekframe_get_row

seekframe_is_empty

Declaration

```
SEEKFRAME_API bool seekframe_is_empty(
    const seekframe_t* frame);
```

Overview

Checks if the frame does not contain any data.

Parameters

- `frame`
    - Frame to check for being empty.

Return value

If the frame is empty, true. If the frame is not empty, false.

See also

- seekframe_t

seekframe_get_header_size

Declaration

```
SEEKFRAME_API size_t seekframe_get_header_size(
    const seekframe_t* frame);
```

Overview

Gets the total size of the frame header data in bytes.

Parameters

- `frame`
    - Frame for which to get the header size.

Return value

On success, the header size as an unsigned integer type. On failure, 0.

See also

- seekframe_t
- seekframe_get_header
- seekcamera_frame_header_t

`seekframe_get_header`

Declaration

```
SEEKFRAME_API void* seekframe_get_header(
    const seekframe_t* frame);
```

Overview

Gets the pointer to the header data of the frame.

Parameters

- `frame`
  - Frame for which to get the header data pointer.

Return value

On success, the header pointer as a non NULL pointer. On failure, NULL.

See also

- seekframe_t
- seekframe_get_header_size
- seekcamera_frame_header_t