

Brightway Vision Camera API Command Reference.

For API version 3.1.20

1 Module Documentation	3
1.1 Connection module	3
1.1.1 Detailed Description	4
1.1.2 Typedef Documentation	4
1.1.2.1 connected_cb_t	4
1.1.2.2 device_discovered_cb_t	4
1.1.2.3 disconnected_cb_t	4
1.1.3 Enumeration Type Documentation	5
1.1.3.1 connection_type_t	5
1.1.4 Function Documentation	5
1.1.4.1 bwv_api_add_connect_callback()	5
1.1.4.2 bwv_api_add_device_discovered_callback()	6
1.1.4.3 bwv_api_add_disconnect_callback()	6
1.1.4.4 bwv_api_connect()	6
1.1.4.5 bwv_api_deinit()	7
1.1.4.6 bwv_api_disconnect()	7
1.1.4.7 bwv_api_get_connected_device()	7
1.1.4.8 bwv_api_get_device()	8
1.1.4.9 bwv_api_get_device_count()	8
1.1.4.10 bwv_api_init()	8
1.1.4.11 bwv_api_is_connected()	8
1.1.4.12 bwv_api_remove_device_discovered_callback()	9
1.1.4.13 bwv_api_remove_disconnect_callback()	9
1.2 Version module	9
1.2.1 Detailed Description	10
1.2.2 Typedef Documentation	10
1.2.2.1 version_callback_t	10
1.2.3 Function Documentation	10
1.2.3.1 bwv_api_get_version()	10
1.2.3.2 bwv_api_version_add_callback_cb()	11
1.2.3.3 bwv_api_version_add_callback_cu()	11
1.2.3.4 bwv_api_version_add_callback_fx3()	12
1.2.3.5 bwv_api_version_get_cb()	12
1.2.3.6 bwv_api_version_get_cu()	12
1.2.3.7 bwv_api_version_get_fx3()	13
1.3 Gating module	13
1.3.1 Detailed Description	14
1.3.2 Macro Definition Documentation	14
1.3.2.1 GATING_OVERPOWER_NONE	15
1.3.2.2 GATING_OVERPOWER_NORMAL	15
1.3.3 Typedef Documentation	15
1.3.3.1 external_trigger_callback_t	15

1.3.3.2 gating_callback_t	15
1.3.4 Function Documentation	15
1.3.4.1 bwv_api_gating_add_callback_alert()	16
1.3.4.2 bwv_api_gating_add_callback_external_trigger()	16
1.3.4.3 bwv_api_gating_add_callback_overpower()	16
1.3.4.4 bwv_api_gating_disable_external_trigger()	17
1.3.4.5 bwv_api_gating_enable_external_trigger()	17
1.3.4.6 bwv_api_gating_read_alert()	17
1.3.4.7 bwv_api_gating_read_external_trigger()	18
1.3.4.8 bwv_api_gating_read_overpower()	18
1.3.4.9 bwv_api_gating_write_overpower()	18
1.3.5 Variable Documentation	19
1.3.5.1 addr	19
1.3.5.2 alert	19
1.3.5.3 ctype	19
1.3.5.4 current_progress	19
1.3.5.5 current_total	19
1.3.5.6 device	19
1.3.5.7 ethernet [1/2]	19
1.3.5.8 ethernet [2/2]	20
1.3.5.9 fps	20
1.3.5.10 frame_type	20
1.3.5.11 function	20
1.3.5.12 fwu_port	20
1.3.5.13 height	20
1.3.5.14 illuminator_en	20
1.3.5.15 ip	20
1.3.5.16 is_8bpp	21
1.3.5.17 limit_type	21
1.3.5.18 limit_value	21
1.3.5.19 link_speed	21
1.3.5.20 local_ip	21
1.3.5.21 location	21
1.3.5.22 lost_frames	21
1.3.5.23 name	21
1.3.5.24 number_of_frames	22
1.3.5.25 overall_progress	22
1.3.5.26 overall_total	22
1.3.5.27 overpower	22
1.3.5.28 pad	22
1.3.5.29 rawimage	22
1.3.5.30 rtp_port	22

1.3.5.31 rx_port	22
1.3.5.32 serial [1/2]	23
1.3.5.33 serial [2/2]	23
1.3.5.34 series	23
1.3.5.35 series_en	23
1.3.5.36 success	23
1.3.5.37 time	23
1.3.5.38 tx_port	23
1.3.5.39 type [1/2]	23
1.3.5.40 type [2/2]	24
1.3.5.41 width	24
1.4 Firmware update module	24
1.4.1 Detailed Description	24
1.4.2 Typedef Documentation	24
1.4.2.1 update_done_callback_t	24
1.4.2.2 update_progress_callback_t	25
1.4.3 Function Documentation	25
1.4.3.1 bwv_api_finish()	25
1.4.3.2 bwv_api_update()	25
1.4.3.3 bwv_api_update_add_callback_done()	26
1.4.3.4 bwv_api_update_add_callback_progress()	26
1.5 Capture module	27
1.5.1 Detailed Description	27
1.5.2 Typedef Documentation	27
1.5.2.1 capture_callback_capture_finished_t	27
1.5.2.2 capture_callback_data_t	28
1.5.2.3 capture_callback_end_t	28
1.5.2.4 capture_callback_start_t	28
1.5.3 Enumeration Type Documentation	29
1.5.3.1 capture_limit_type_t	29
1.5.4 Function Documentation	29
1.5.4.1 bwv_api_capture_add_callback_capture_finished()	29
1.5.4.2 bwv_api_capture_add_callback_data()	29
1.5.4.3 bwv_api_capture_add_callback_end()	30
1.5.4.4 bwv_api_capture_add_callback_error()	30
1.5.4.5 bwv_api_capture_add_callback_start()	31
1.5.4.6 bwv_api_capture_get_lost_frames()	31
1.5.4.7 bwv_api_capture_remove_callback_capture_finished()	31
1.5.4.8 bwv_api_capture_remove_callback_data()	32
1.5.4.9 bwv_api_capture_remove_callback_end()	32
1.5.4.10 bwv_api_capture_remove_callback_start()	33
1.5.4.11 bwv_api_capture_reset_lost_frames()	33

1.5.4.12 bwv_api_capture_start_capture()	33
1.5.4.13 bwv_api_capture_start_limited_capture()	34
1.5.4.14 bwv_api_capture_start_preview()	34
1.5.4.15 bwv_api_capture_stop_capture()	34
1.5.4.16 bwv_api_capture_stop_preview()	35
2 Data Structure Documentation	37
2.1 capture_data_t Struct Reference	37
2.2 capture_finish_info_t Struct Reference	37
2.3 connection_t Struct Reference	37
2.4 connection_type_t Struct Reference	38
2.5 ethernet_addr_t Struct Reference	38
2.6 gating_data_t Struct Reference	38
2.7 serial_addr_t Struct Reference	38
2.8 update_data_t Struct Reference	39
2.9 version_data_t Struct Reference	39
Index	41

Chapter 1

Module Documentation

1.1 Connection module

Data Structures

- struct **serial_addr_t**
- struct **ethernet_addr_t**
- struct **connection_type_t**
- struct **connection_t**

Typedefs

- typedef void(* **device_discovered_cb_t**) (device_t device, void *param)
- typedef void(* **connected_cb_t**) (void *param)
- typedef void(* **disconnected_cb_t**) (device_t device, void *param)

Functions

- int **bwv_api_add_device_discovered_callback** (**device_discovered_cb_t** callback, void *param)
- int **bwv_api_remove_device_discovered_callback** (**device_discovered_cb_t** callback)
- int **bwv_api_get_device_count** ()
- int **bwv_api_get_device** (int index, device_t *device)
- void * **bwv_api_init** ()
- void **bwv_api_deinit** (void *api)
- int **bwv_api_add_connect_callback** (void *api, **connected_cb_t** callback, void *param)
- int **bwv_api_add_disconnect_callback** (void *api, **disconnected_cb_t** callback, void *param)
- void **bwv_api_remove_disconnect_callback** (void *api, int index)
- bool **bwv_api_connect** (void *api, device_t device)
- void **bwv_api_disconnect** (void *api)
- bool **bwv_api_is_connected** (void *api)
- device_t **bwv_api_get_connected_device** (void *api)

1.1.1 Detailed Description

This module provides connectivity routines such as device discovery, device connection/disconnection handling

Usage sequence (in case only one system is connected to PC):

1. Start device discovery by calling to *api_start_device_discovery*
2. Wait till there is at least 1 device (*api_get_device_count* > 0)
3. Connect to system by calling *api_connect* (0, &device_info)

1.1.2 Typedef Documentation

1.1.2.1 connected_cb_t

```
typedef void(* connected_cb_t) (void *param)
```

API callback which is called when API is connected to system.

Parameters

<i>param</i>	User data
--------------	-----------

1.1.2.2 device_discovered_cb_t

```
typedef void(* device_discovered_cb_t) (device_t device, void *param)
```

API callback which is called when Device was discovered by API.

Parameters

<i>device</i>	Device connection description
<i>param</i>	User data

1.1.2.3 disconnected_cb_t

```
typedef void(* disconnected_cb_t) (device_t device, void *param)
```

API callback which is called when API was disconnected.

Parameters

<i>param</i>	User data
--------------	-----------

1.1.3 Enumeration Type Documentation

1.1.3.1 connection_type_t

```
enum connection_type_t : unsigned char
```

Enumerator

CT_UNKNOWN	0x00 unknown type
CT_SERIAL_CTRL_BOX	0x01 Connection to Control Box through serial port
CT_SERIAL_COMM_BOX	0x02 Connection to Communication Box through serial port
CT_ETHERNET	0x03 Connection to Ethernet Camera unit

1.1.4 Function Documentation

1.1.4.1 bwv_api_add_connect_callback()

```
int bwv_api_add_connect_callback (
    void * api,
    connected_cb_t callback,
    void * param )
```

Register callback for API connected state.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

internal callback index or -1 if failed

1.1.4.2 bwv_api_add_device_discovered_callback()

```
int bwv_api_add_device_discovered_callback (
    device_discovered_cb_t callback,
    void * param )
```

Add API callback which is called when Device was discovered by API.

Parameters

<i>callback</i>	Callback function
<i>param</i>	User data

1.1.4.3 bwv_api_add_disconnect_callback()

```
int bwv_api_add_disconnect_callback (
    void * api,
    disconnected_cb_t callback,
    void * param )
```

Register callback for API disconnected state.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

internal callback index or -1 if failed

1.1.4.4 bwv_api_connect()

```
bool bwv_api_connect (
    void * api,
    device_t device )
```

Connect to selected device.

Parameters

<i>api</i>	Handle to API instance
<i>device</i>	A device to connect to

Returns

1 on success, 0 on failure

1.1.4.5 bwv_api_deinit()

```
void bwv_api_deinit (
    void * api )
```

Close API instance.

Parameters

<i>handle</i>	to API instance
---------------	-----------------

1.1.4.6 bwv_api_disconnect()

```
void bwv_api_disconnect (
    void * api )
```

Disconnect from device.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.1.4.7 bwv_api_get_connected_device()

```
device_t bwv_api_get_connected_device (
    void * api )
```

Get information about device that is currently used by API.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

Returns

Device information

1.1.4.8 **bwv_api_get_device()**

```
int bwv_api_get_device (
    int index,
    device_t * device )
```

Get device connection description from internal list of devices that were discovered.

Parameters

<i>index</i>	0-based array index
<i>device</i>	preallocated buffer for storing device connection description

Returns

0 on success, -1 if index is out of bounds

1.1.4.9 **bwv_api_get_device_count()**

```
int bwv_api_get_device_count ( )
```

Retrieve number of devices discovered so far.

Returns

number of devices

1.1.4.10 **bwv_api_init()**

```
void * bwv_api_init ( )
```

Initialize API instance.

Returns

handle to API instance

1.1.4.11 **bwv_api_is_connected()**

```
bool bwv_api_is_connected (
    void * api )
```

Check if API is connected.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

Returns

1 when connected, 0 otherwise

1.1.4.12 **bwv_api_remove_device_discovered_callback()**

```
int bwv_api_remove_device_discovered_callback (
    device_discovered_cb_t callback )
```

Remove device discovery callback.

Parameters

<i>callback</i>	Callback function
-----------------	-------------------

1.1.4.13 **bwv_api_remove_disconnect_callback()**

```
void bwv_api_remove_disconnect_callback (
    void * api,
    int index )
```

Register callback for API disconnected state.

Parameters

<i>api</i>	Handle to API instance
<i>index</i>	Internal callback index received from <code>api_add_disconnect_callback</code>

1.2 Version module

Data Structures

- struct **version_data_t**

Typedefs

- typedef void(* **version_callback_t**) (**version_data_t** data, void *param)

Functions

- int **bwv_api_version_add_callback_cu** (void *api, **version_callback_t** callback, void *param)
- int **bwv_api_version_add_callback_cb** (void *api, **version_callback_t** callback, void *param)
- int **bwv_api_version_add_callback_fx3** (void *api, **version_callback_t** callback, void *param)
- **version_data_t** **bwv_api_get_version** (void *api)
- void **bwv_api_version_get_cu** (void *api)
- void **bwv_api_version_get_cb** (void *api)
- void **bwv_api_version_get_fx3** (void *api)

1.2.1 Detailed Description

This module provides functions for reading firmware version information of different system units

1.2.2 Typedef Documentation

1.2.2.1 version_callback_t

```
typedef void(* version_callback_t) ( version_data_t data, void *param)
```

Version information service.

Parameters

<i>data</i>	Version information structure
<i>param</i>	User data

1.2.3 Function Documentation

1.2.3.1 bwv_api_get_version()

```
version_data_t bwv_api_get_version (  
    void * api )
```

Get API version.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

Returns

Version information

1.2.3.2 bwv_api_version_add_callback_cb()

```
int bwv_api_version_add_callback_cb (
    void * api,
    version_callback_t callback,
    void * param )
```

Add API callback for Control Box/Communication Box version.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.2.3.3 bwv_api_version_add_callback_cu()

```
int bwv_api_version_add_callback_cu (
    void * api,
    version_callback_t callback,
    void * param )
```

Add API callback for Camera unit version.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.2.3.4 bwv_api_version_add_callback_fx3()

```
int bwv_api_version_add_callback_fx3 (
    void * api,
    version_callback_t callback,
    void * param )
```

Add API callback for FX3 USB peripheral IC.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.2.3.5 bwv_api_version_get_cb()

```
void bwv_api_version_get_cb (
    void * api )
```

Read Control Box/Communication Box version.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.2.3.6 bwv_api_version_get_cu()

```
void bwv_api_version_get_cu (
    void * api )
```

Read Camera unit version.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.2.3.7 bwv_api_version_get_fx3()

```
void bwv_api_version_get_fx3 (
    void * api )
```

Read FX3 USB peripheral IC version.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.3 Gating module

Modules

- Firmware update module
- Capture module

Data Structures

- struct **gating_data_t**

Macros

- #define **GATING_OVERPOWER_NONE** 0
- #define **GATING_OVERPOWER_NORMAL** 1

Typedefs

- typedef void(* **external_trigger_callback_t**) (unsigned char data, void *param)
- typedef void(* **gating_callback_t**) (**gating_data_t** data, void *param)

Functions

- int **bwv_api_gating_add_callback_alert** (void *api, **gating_callback_t** callback, void *param)
- void **bwv_api_gating_read_alert** (void *api)
- int **bwv_api_gating_add_callback_overpower** (void *api, **gating_callback_t** callback, void *param)
- void **bwv_api_gating_read_overpower** (void *api)
- void **bwv_api_gating_write_overpower** (void *api, **gating_data_t** data, unsigned int code)
- int **bwv_api_gating_add_callback_external_trigger** (void *api, **external_trigger_callback_t** callback, void *param)
- void **bwv_api_gating_read_external_trigger** (void *api)
- void **bwv_api_gating_enable_external_trigger** (void *api)
- void **bwv_api_gating_disable_external_trigger** (void *api)

Variables

- char **addr** [16]
- unsigned short **location** [16]
- unsigned long **ip**
- unsigned long **link_speed**
- unsigned short **rx_port**
- unsigned short **tx_port**
- unsigned short **fwu_port**
- unsigned short **rtp_port**
- unsigned long **local_ip**
- unsigned char **pad** [16]
- **connection_type_t** type
- unsigned char **ctype**
- char **name** [64]
- **serial_addr_t** serial
- **ethernet_addr_t** ethernet
- uint64_t **time**
- unsigned char **type**
- unsigned char **series**
- unsigned char **function** [4]
- unsigned char **illuminator_en**
- unsigned char **series_en**
- unsigned char **overpower**
- unsigned int **alert**
- unsigned char **device**
- unsigned char **overal_progress**
- unsigned char **overall_total**
- unsigned int **current_progress**
- unsigned int **current_total**
- uint8_t **frame_type**
- uint32_t **lost_frames**
- int **width**
- int **height**
- unsigned short * **rawimage**
- bool **is_8bpp**
- uint8_t **fps**
- **capture_limit_type_t** limit_type
- uint32_t **limit_value**
- bool **success**
- uint32_t **number_of_frames**

1.3.1 Detailed Description

This module provides functions for working with Camera unit gating parameters

1.3.2 Macro Definition Documentation

1.3.2.1 GATING_OVERPOWER_NONE

```
#define GATING_OVERPOWER_NONE 0
```

No illuminator override

1.3.2.2 GATING_OVERPOWER_NORMAL

```
#define GATING_OVERPOWER_NORMAL 1
```

Normal illuminator override

1.3.3 Typedef Documentation

1.3.3.1 external_trigger_callback_t

```
typedef void(* external_trigger_callback_t) (unsigned char data, void *param)
```

API callback signature for reading external trigger state.

Parameters

<i>data</i>	External trigger status (0 - disabled, 1 - enabled)
<i>param</i>	User data

1.3.3.2 gating_callback_t

```
typedef void(* gating_callback_t) ( gating_data_t data, void *param)
```

API callback signature for gating related commands.

Parameters

<i>data</i>	Gating data structure. Only relevant fields are filled when callback arrives.
<i>param</i>	User data

1.3.4 Function Documentation

1.3.4.1 bwv_api_gating_add_callback_alert()

```
int bwv_api_gating_add_callback_alert (
    void * api,
    gating_callback_t callback,
    void * param )
```

Add API callback for retrieve camera unit alert value (gating_read_alert)

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance
<i>callback</i>	Callback function. Note - only relevant fields of callback data structure are filled when callback arrives.
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.3.4.2 bwv_api_gating_add_callback_external_trigger()

```
int bwv_api_gating_add_callback_external_trigger (
    void * api,
    external_trigger_callback_t callback,
    void * param )
```

Add API callback for read external trigger status.

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance
<i>callback</i>	Callback function
<i>param</i>	User data to pass to callback

Returns

0 on success, -1 on failure

1.3.4.3 bwv_api_gating_add_callback_overpower()

```
int bwv_api_gating_add_callback_overpower (
    void * api,
    gating_callback_t callback,
    void * param )
```

Add API callback for read overpower value (gating_read_overpower)

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance
<i>callback</i>	Callback function. Note - only relevant fields of callback data structure are filled when callback arrives.
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.3.4.4 **bwv_api_gating_disable_external_trigger()**

```
void bwv_api_gating_disable_external_trigger (
    void * api )
```

Disable external trigger.

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance

1.3.4.5 **bwv_api_gating_enable_external_trigger()**

```
void bwv_api_gating_enable_external_trigger (
    void * api )
```

Enable external trigger.

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance

1.3.4.6 **bwv_api_gating_read_alert()**

```
void bwv_api_gating_read_alert (
    void * api )
```

Read alert status.

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance

1.3.4.7 bwv_api_gating_read_external_trigger()

```
void bwv_api_gating_read_external_trigger (
    void * api )
```

Read external trigger status.

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance

1.3.4.8 bwv_api_gating_read_overpower()

```
void bwv_api_gating_read_overpower (
    void * api )
```

Read overpower.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.3.4.9 bwv_api_gating_write_overpower()

```
void bwv_api_gating_write_overpower (
    void * api,
    gating_data_t data,
    unsigned int code )
```

Write overpower.

Parameters

<i>api</i>	Handle to API instance
<i>api</i>	Pointer to API instance
<i>data</i>	structure with all relevant members filled
<i>code</i>	Protection code. (Should be requested from BWV)

1.3.5 Variable Documentation

1.3.5.1 addr

```
char addr[16]
```

Serial port path (E.g. COM3, /dev/ttyUSB0, /dev/ttyACM0, etc...)

1.3.5.2 alert

```
unsigned int alert
```

Alert code

1.3.5.3 ctype

```
unsigned char ctype
```

reserved

1.3.5.4 current_progress

```
unsigned int current_progress
```

Current progress of current update stage

1.3.5.5 current_total

```
unsigned int current_total
```

Total progress of current update stage

1.3.5.6 device

```
unsigned char device
```

Device id being updated (Camera unit/Control box/Communication box)

1.3.5.7 ethernet [1/2]

```
ethernet_addr_t ethernet
```

Ethernet connection description

1.3.5.8 ethernet [2/2]

`ethernet_addr_t` ethernet

Ethernet connection description

1.3.5.9 fps

`uint8_t` fps

Current FPS value

1.3.5.10 frame_type

`uint8_t` frame_type

Current frame type

1.3.5.11 function

`unsigned char` function[4]

Current function

1.3.5.12 fwu_port

`unsigned short` fwu_port

Number of Camera unit FW update port

1.3.5.13 height

`int` height

Frame height in pixels

1.3.5.14 illuminator_en

`unsigned char` illuminator_en

Illuminator status. 0x00 - disabled, 0x01 - enabled

1.3.5.15 ip

`unsigned long` ip

IP address of Camera unit

1.3.5.16 is_8bpp

```
bool is_8bpp
```

Image pixel format. True - 8BPP, False - 10BPP

1.3.5.17 limit_type

```
capture_limit_type_t limit_type
```

Capture limit type

1.3.5.18 limit_value

```
uint32_t limit_value
```

Capture limit value

1.3.5.19 link_speed

```
unsigned long link_speed
```

Current link layer speed

1.3.5.20 local_ip

```
unsigned long local_ip
```

PC's local IP

1.3.5.21 location

```
unsigned short location[16]
```

Serial port device location (VID/PID or USB path)

1.3.5.22 lost_frames

```
uint32_t lost_frames
```

Number of lost frames

1.3.5.23 name

```
char name[64]
```

Displayable device name

1.3.5.24 number_of_frames

```
uint32_t number_of_frames
```

Total number of frames captured

1.3.5.25 overal_progress

```
unsigned char overal_progress
```

Current progress of firmware update operation

1.3.5.26 overall_total

```
unsigned char overall_total
```

Overall total percentage of firmware update operation

1.3.5.27 overpower

```
unsigned char overpower
```

Illuminator override status

1.3.5.28 pad

```
unsigned char pad[16]
```

reserved

1.3.5.29 rawimage

```
unsigned short* rawimage
```

Raw frame data. Note - this is static pointer. Do not attempt to free.

1.3.5.30 rtp_port

```
unsigned short rtp_port
```

Number of Camera unit RTP broadcast port

1.3.5.31 rx_port

```
unsigned short rx_port
```

Number of Camera unit receiving port

1.3.5.32 serial [1/2]

`serial_addr_t serial`

Serial port description

1.3.5.33 serial [2/2]

`serial_addr_t serial`

Serial port description

1.3.5.34 series

`unsigned char series`

Frame type series

1.3.5.35 series_en

`unsigned char series_en`

Series status. 0x00 - disabled, 0x01 - enabled

1.3.5.36 success

`bool success`

True when capture was finished by user, or by limit value, false if error occurred

1.3.5.37 time

`uint64_t time`

Discovery time

1.3.5.38 tx_port

`unsigned short tx_port`

Number of Camera unit transmitting port

1.3.5.39 type [1/2]

`connection_type_t type`

Device connection type

1.3.5.40 type [2/2]

unsigned char type

Frame type

1.3.5.41 width

int width

Frame width in pixels

1.4 Firmware update module

Data Structures

- struct **update_data_t**

Typedefs

- typedef void(* **update_progress_callback_t**) (**update_data_t** data, void *param)
- typedef void(* **update_done_callback_t**) (unsigned char local_error, unsigned char system_error, void *param)

Functions

- int **bwv_api_update_add_callback_progress** (void *api, **update_progress_callback_t** callback, void *param)
- int **bwv_api_update_add_callback_done** (void *api, **update_done_callback_t** callback, void *param)
- int **bwv_api_update** (void *api, char *filename, int filename_len)
- void **bwv_api_finish** (void *api)

1.4.1 Detailed Description

This module provides control functions for work with Camera unit's FPN module

1.4.2 Typedef Documentation

1.4.2.1 update_done_callback_t

```
typedef void(* update_done_callback_t) (unsigned char local_error, unsigned char system_error, void *param)
```

API callback signature for update done event.

Parameters

<i>local_error</i>	API error. 0x00 - no error
<i>system_error</i>	System (Camera unit/Control box/Communication box) error. 0x00 - no error
<i>param</i>	User data

1.4.2.2 update_progress_callback_t

```
typedef void(* update_progress_callback_t) ( update_data_t data, void *param)
```

API callback signature for update progress indication.

Parameters

<i>data</i>	Update information data
<i>param</i>	User data

1.4.3 Function Documentation

1.4.3.1 bwv_api_finish()

```
void bwv_api_finish (
    void * api )
```

Finalize firmware update process.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.4.3.2 bwv_api_update()

```
int bwv_api_update (
    void * api,
    char * filename,
    int filename_len )
```

Start firmware update process.

Parameters

<i>api</i>	Handle to API instance
<i>filename</i>	FW file name
<i>filename_len</i>	Length in bytes of file name

Returns

0 on successful start, -1 on failure starting process

1.4.3.3 bwv_api_update_add_callback_done()

```
int bwv_api_update_add_callback_done (
    void * api,
    update_done_callback_t callback,
    void * param )
```

Add APi callback for update done event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.4.3.4 bwv_api_update_add_callback_progress()

```
int bwv_api_update_add_callback_progress (
    void * api,
    update_progress_callback_t callback,
    void * param )
```

Add APi callback for progress indication.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.5 Capture module

Data Structures

- struct **capture_data_t**
- struct **capture_finish_info_t**

Typedefs

- typedef void(* **capture_callback_start_t**) (int width, int height, void *param)
- typedef void(* **capture_callback_data_t**) (**capture_data_t** data, void *param)
- typedef void(* **capture_callback_end_t**) (void *param)
- typedef void(* **capture_callback_capture_finished_t**) (**capture_finish_info_t** info, void *param)

Functions

- int **bwv_api_capture_add_callback_start** (void *api, **capture_callback_start_t** callback, void *param)
- int **bwv_api_capture_add_callback_end** (void *api, **capture_callback_end_t** callback, void *param)
- int **bwv_api_capture_add_callback_error** (void *api, **capture_callback_end_t** callback, void *param)
- int **bwv_api_capture_add_callback_capture_finished** (void *api, **capture_callback_capture_finished_t** callback, void *param)
- int **bwv_api_capture_add_callback_data** (void *api, **capture_callback_data_t** callback, void *param)
- int **bwv_api_capture_remove_callback_start** (void *api, **capture_callback_start_t** callback)
- int **bwv_api_capture_remove_callback_end** (void *api, **capture_callback_end_t** callback)
- int **bwv_api_capture_remove_callback_data** (void *api, **capture_callback_data_t** callback)
- int **bwv_api_capture_remove_callback_capture_finished** (void *api, **capture_callback_capture_finished_t** callback)
- int **bwv_api_capture_start_preview** (void *api)
- void **bwv_api_capture_stop_preview** (void *api)
- int **bwv_api_capture_start_capture** (void *api, const char *filename)
- int **bwv_api_capture_start_limited_capture** (void *api, const char *filename, **capture_limit_type_t** type, uint32_t value)
- void **bwv_api_capture_stop_capture** (void *api)
- unsigned int **bwv_api_capture_get_lost_frames** (void *api)
- void **bwv_api_capture_reset_lost_frames** (void *api)

1.5.1 Detailed Description

This module provides video capture related functions

1.5.2 Typedef Documentation

1.5.2.1 **capture_callback_capture_finished_t**

```
typedef void(* capture_callback_capture_finished_t) ( capture_finish_info_t info, void *param)
```

API callback for capture finished event. Called when limited capture is done.

Parameters

<i>info</i>	Capture finish information
<i>param</i>	User data

1.5.2.2 capture_callback_data_t

```
typedef void(* capture_callback_data_t) ( capture_data_t data, void *param)
```

API callback for frame capture event.

Parameters

<i>data</i>	Capture status/information structure
<i>param</i>	User data

1.5.2.3 capture_callback_end_t

```
typedef void(* capture_callback_end_t) (void *param)
```

API callback for capture end event.

Parameters

<i>param</i>	User data
--------------	-----------

1.5.2.4 capture_callback_start_t

```
typedef void(* capture_callback_start_t) (int width, int height, void *param)
```

API callback for capture start event.

Parameters

<i>width</i>	Frame width
<i>height</i>	Frame height
<i>param</i>	User data

1.5.3 Enumeration Type Documentation

1.5.3.1 `capture_limit_type_t`

enum `capture_limit_type_t`

Limited capture mode.

Enumerator

<code>CM_UNLIMITED</code>	Unlimited capture
<code>CM_LIMIT_FRAMES</code>	Capture is limited by number of frames
<code>CM_LIMIT_TIME</code>	Capture is limited by time
<code>CM_LIMIT_FILESIZE</code>	Capture is limited by file size

1.5.4 Function Documentation

1.5.4.1 `bwv_api_capture_add_callback_capture_finished()`

```
int bwv_api_capture_add_callback_capture_finished (
    void * api,
    capture_callback_capture_finished_t callback,
    void * param )
```

Add API callback for capture finished event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.5.4.2 `bwv_api_capture_add_callback_data()`

```
int bwv_api_capture_add_callback_data (
    void * api,
```

```
capture_callback_data_t callback,  
void * param )
```

Add API callback for capture frame event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.5.4.3 bwv_api_capture_add_callback_end()

```
int bwv_api_capture_add_callback_end (  
    void * api,  
    capture_callback_end_t callback,  
    void * param )
```

Add API callback for capture end event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.5.4.4 bwv_api_capture_add_callback_error()

```
int bwv_api_capture_add_callback_error (  
    void * api,  
    capture_callback_end_t callback,  
    void * param )
```

Add API callback for preview failure event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.5.4.5 bwv_api_capture_add_callback_start()

```
int bwv_api_capture_add_callback_start (
    void * api,
    capture_callback_start_t callback,
    void * param )
```

Add API callback for capture start event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function
<i>param</i>	User data

Returns

0 on success, -1 on failure

1.5.4.6 bwv_api_capture_get_lost_frames()

```
unsigned int bwv_api_capture_get_lost_frames (
    void * api )
```

Get number of lost frames.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.5.4.7 bwv_api_capture_remove_callback_capture_finished()

```
int bwv_api_capture_remove_callback_capture_finished (
    void * api,
    capture_callback_capture_finished_t callback )
```

Remove API callback for capture finished event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function

Returns

0 on success, -1 on failure

1.5.4.8 bwv_api_capture_remove_callback_data()

```
int bwv_api_capture_remove_callback_data (
    void * api,
    capture_callback_data_t callback )
```

Remove API callback for capture frame event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function

Returns

0 on success, -1 on failure

1.5.4.9 bwv_api_capture_remove_callback_end()

```
int bwv_api_capture_remove_callback_end (
    void * api,
    capture_callback_end_t callback )
```

Remove API callback for capture end event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function

Returns

0 on success, -1 on failure

1.5.4.10 bwv_api_capture_remove_callback_start()

```
int bwv_api_capture_remove_callback_start (
    void * api,
    capture_callback_start_t callback )
```

Remove API callback for capture start event.

Parameters

<i>api</i>	Handle to API instance
<i>callback</i>	Callback function

Returns

0 on success, -1 on failure

1.5.4.11 bwv_api_capture_reset_lost_frames()

```
void bwv_api_capture_reset_lost_frames (
    void * api )
```

Reset lost frames counter.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.5.4.12 bwv_api_capture_start_capture()

```
int bwv_api_capture_start_capture (
    void * api,
    const char * filename )
```

Start simple capture to file.

Parameters

<i>api</i>	Handle to API instance
<i>filename</i>	target filename

Returns

0 on success, -1 on failure

1.5.4.13 bwv_api_capture_start_limited_capture()

```
int bwv_api_capture_start_limited_capture (
    void * api,
    const char * filename,
    capture_limit_type_t type,
    uint32_t value )
```

Start limited capture.

Parameters

<i>api</i>	Handle to API instance
<i>filename</i>	target filename
<i>type</i>	type of limited capture
<i>value</i>	limit value

Returns

0 on success, -1 on failure

1.5.4.14 bwv_api_capture_start_preview()

```
int bwv_api_capture_start_preview (
    void * api )
```

Initialize video channel and start image preview.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

Returns

0 on success, -1 on failure

1.5.4.15 bwv_api_capture_stop_capture()

```
void bwv_api_capture_stop_capture (
    void * api )
```

Stop capture.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

1.5.4.16 bwv_api_capture_stop_preview()

```
void bwv_api_capture_stop_preview (
    void * api )
```

Stop image preview.

Parameters

<i>api</i>	Handle to API instance
------------	------------------------

Chapter 2

Data Structure Documentation

2.1 `capture_data_t` Struct Reference

```
#include <bwv_api.h>
```

Data Fields

- `uint8_t frame_type`
- `uint32_t lost_frames`
- `int width`
- `int height`
- `unsigned short * rawimage`
- `bool is_8bpp`
- `uint8_t fps`
- `capture_limit_type_t limit_type`
- `uint32_t limit_value`

2.2 `capture_finish_info_t` Struct Reference

```
#include <bwv_api.h>
```

Data Fields

- `bool success`
- `uint32_t number_of_frames`

2.3 `connection_t` Struct Reference

```
#include <bwv_api.h>
```

2.4 connection_type_t Struct Reference

```
#include <bwv_api.h>
```

2.5 ethernet_addr_t Struct Reference

```
#include <bwv_api.h>
```

Data Fields

- unsigned long **ip**
- unsigned long **link_speed**
- unsigned short **rx_port**
- unsigned short **tx_port**
- unsigned short **fwu_port**
- unsigned short **rtp_port**
- unsigned long **local_ip**
- unsigned char **pad** [16]

2.6 gating_data_t Struct Reference

```
#include <bwv_api.h>
```

Data Fields

- unsigned char **type**
- unsigned char **series**
- unsigned char **function** [4]
- unsigned char **illuminator_en**
- unsigned char **series_en**
- unsigned char **overpower**
- unsigned int **alert**

2.7 serial_addr_t Struct Reference

```
#include <bwv_api.h>
```

Data Fields

- char **addr** [16]
- unsigned short **location** [16]

2.8 `update_data_t` Struct Reference

```
#include <bwv_api.h>
```

Data Fields

- unsigned char **device**
- unsigned char **overal_progress**
- unsigned char **overall_total**
- unsigned int **current_progress**
- unsigned int **current_total**

2.9 `version_data_t` Struct Reference

```
#include <bwv_api.h>
```


Index

addr
 Gating module, 19

alert
 Gating module, 19

bwv_api_add_connect_callback
 Connection module, 5

bwv_api_add_device_discovered_callback
 Connection module, 5

bwv_api_add_disconnect_callback
 Connection module, 6

bwv_api_capture_add_callback_capture_finished
 Capture module, 29

bwv_api_capture_add_callback_data
 Capture module, 29

bwv_api_capture_add_callback_end
 Capture module, 30

bwv_api_capture_add_callback_error
 Capture module, 30

bwv_api_capture_add_callback_start
 Capture module, 31

bwv_api_capture_get_lost_frames
 Capture module, 31

bwv_api_capture_remove_callback_capture_finished
 Capture module, 31

bwv_api_capture_remove_callback_data
 Capture module, 32

bwv_api_capture_remove_callback_end
 Capture module, 32

bwv_api_capture_remove_callback_start
 Capture module, 32

bwv_api_capture_reset_lost_frames
 Capture module, 33

bwv_api_capture_start_capture
 Capture module, 33

bwv_api_capture_start_limited_capture
 Capture module, 33

bwv_api_capture_start_preview
 Capture module, 34

bwv_api_capture_stop_capture
 Capture module, 34

bwv_api_capture_stop_preview
 Capture module, 35

bwv_api_connect
 Connection module, 6

bwv_api_deinit
 Connection module, 7

bwv_api_disconnect
 Connection module, 7

bwv_api_finish
 Firmware update module, 25

bwv_api_gating_add_callback_alert
 Gating module, 15

bwv_api_gating_add_callback_external_trigger
 Gating module, 16

bwv_api_gating_add_callback_overpower
 Gating module, 16

bwv_api_gating_disable_external_trigger
 Gating module, 17

bwv_api_gating_enable_external_trigger
 Gating module, 17

bwv_api_gating_read_alert
 Gating module, 17

bwv_api_gating_read_external_trigger
 Gating module, 18

bwv_api_gating_read_overpower
 Gating module, 18

bwv_api_gating_write_overpower
 Gating module, 18

bwv_api_get_connected_device
 Connection module, 7

bwv_api_get_device
 Connection module, 7

bwv_api_get_device_count
 Connection module, 8

bwv_api_get_version
 Version module, 10

bwv_api_init
 Connection module, 8

bwv_api_is_connected
 Connection module, 8

bwv_api_remove_device_discovered_callback
 Connection module, 9

bwv_api_remove_disconnect_callback
 Connection module, 9

bwv_api_update
 Firmware update module, 25

bwv_api_update_add_callback_done
 Firmware update module, 26

bwv_api_update_add_callback_progress
 Firmware update module, 26

bwv_api_version_add_callback_cb
 Version module, 11

bwv_api_version_add_callback_cu
 Version module, 11

bwv_api_version_add_callback_fx3
 Version module, 11

bwv_api_version_get_cb
 Version module, 12

bwv_api_version_get_cu	bwv_api_connect, 6
Version module, 12	bwv_api_deinit, 7
bwv_api_version_get_fx3	bwv_api_disconnect, 7
Version module, 12	bwv_api_get_connected_device, 7
	bwv_api_get_device, 7
Capture module, 27	bwv_api_get_device_count, 8
bwv_api_capture_add_callback_capture_finished,	bwv_api_init, 8
29	bwv_api_is_connected, 8
bwv_api_capture_add_callback_data, 29	bwv_api_remove_device_discovered_callback, 9
bwv_api_capture_add_callback_end, 30	bwv_api_remove_disconnect_callback, 9
bwv_api_capture_add_callback_error, 30	connected_cb_t, 4
bwv_api_capture_add_callback_start, 31	connection_type_t, 5
bwv_api_capture_get_lost_frames, 31	CT_ETHERNET, 5
bwv_api_capture_remove_callback_capture_finished,	CT_SERIAL_COMM_BOX, 5
31	CT_SERIAL_CTRL_BOX, 5
bwv_api_capture_remove_callback_data, 32	CT_UNKNOWN, 5
bwv_api_capture_remove_callback_end, 32	device_discovered_cb_t, 4
bwv_api_capture_remove_callback_start, 32	disconnected_cb_t, 4
bwv_api_capture_reset_lost_frames, 33	connection_t, 37
bwv_api_capture_start_capture, 33	connection_type_t, 38
bwv_api_capture_start_limited_capture, 33	Connection module, 5
bwv_api_capture_start_preview, 34	CT_ETHERNET
bwv_api_capture_stop_capture, 34	Connection module, 5
bwv_api_capture_stop_preview, 35	CT_SERIAL_COMM_BOX
capture_callback_capture_finished_t, 27	Connection module, 5
capture_callback_data_t, 28	CT_SERIAL_CTRL_BOX
capture_callback_end_t, 28	Connection module, 5
capture_callback_start_t, 28	CT_UNKNOWN
capture_limit_type_t, 29	Connection module, 5
CM_LIMIT_FILESIZE, 29	ctype
CM_LIMIT_FRAMES, 29	Gating module, 19
CM_LIMIT_TIME, 29	current_progress
CM_UNLIMITED, 29	Gating module, 19
capture_callback_capture_finished_t	current_total
Capture module, 27	Gating module, 19
capture_callback_data_t	
Capture module, 28	device
capture_callback_end_t	Gating module, 19
Capture module, 28	device_discovered_cb_t
capture_callback_start_t	Connection module, 4
Capture module, 28	disconnected_cb_t
capture_data_t, 37	Connection module, 4
capture_finish_info_t, 37	
capture_limit_type_t	ethernet
Capture module, 29	Gating module, 19
CM_LIMIT_FILESIZE	ethernet_addr_t, 38
Capture module, 29	external_trigger_callback_t
CM_LIMIT_FRAMES	Gating module, 15
Capture module, 29	
CM_LIMIT_TIME	Firmware update module, 24
Capture module, 29	bwv_api_finish, 25
CM_UNLIMITED	bwv_api_update, 25
Capture module, 29	bwv_api_update_add_callback_done, 26
connected_cb_t	bwv_api_update_add_callback_progress, 26
Connection module, 4	update_done_callback_t, 24
Connection module, 3	update_progress_callback_t, 25
bwv_api_add_connect_callback, 5	
bwv_api_add_device_discovered_callback, 5	fps
bwv_api_add_disconnect_callback, 6	Gating module, 20
	frame_type

Gating module, 20	Gating module, 15
function	gating_data_t, 38
Gating module, 20	GATING_OVERPOWER_NONE
fwu_port	Gating module, 14
Gating module, 20	GATING_OVERPOWER_NORMAL
	Gating module, 15
Gating module, 13	height
addr, 19	Gating module, 20
alert, 19	illuminator_en
bwv_api_gating_add_callback_alert, 15	Gating module, 20
bwv_api_gating_add_callback_external_trigger, 16	ip
bwv_api_gating_add_callback_overpower, 16	Gating module, 20
bwv_api_gating_disable_external_trigger, 17	is_8bpp
bwv_api_gating_enable_external_trigger, 17	Gating module, 20
bwv_api_gating_read_alert, 17	limit_type
bwv_api_gating_read_external_trigger, 18	Gating module, 21
bwv_api_gating_read_overpower, 18	limit_value
bwv_api_gating_write_overpower, 18	Gating module, 21
ctype, 19	link_speed
current_progress, 19	Gating module, 21
current_total, 19	local_ip
device, 19	Gating module, 21
ethernet, 19	location
external_trigger_callback_t, 15	Gating module, 21
fps, 20	lost_frames
frame_type, 20	Gating module, 21
function, 20	name
fwu_port, 20	Gating module, 21
gating_callback_t, 15	number_of_frames
GATING_OVERPOWER_NONE, 14	Gating module, 21
GATING_OVERPOWER_NORMAL, 15	overall_progress
height, 20	Gating module, 22
illuminator_en, 20	overall_total
ip, 20	Gating module, 22
is_8bpp, 20	overpower
limit_type, 21	Gating module, 22
limit_value, 21	pad
link_speed, 21	Gating module, 22
local_ip, 21	rawimage
location, 21	Gating module, 22
lost_frames, 21	rtp_port
name, 21	Gating module, 22
number_of_frames, 21	rx_port
overall_progress, 22	Gating module, 22
overall_total, 22	serial
overpower, 22	Gating module, 22, 23
pad, 22	serial_addr_t, 38
rawimage, 22	series
rtp_port, 22	Gating module, 23
rx_port, 22	series_en
serial, 22, 23	Gating module, 23
series, 23	success
series_en, 23	
success, 23	
time, 23	
tx_port, 23	
type, 23	
width, 24	
gating_callback_t	

- Gating module, 23
- time
 - Gating module, 23
- tx_port
 - Gating module, 23
- type
 - Gating module, 23
- update_data_t, 39
- update_done_callback_t
 - Firmware update module, 24
- update_progress_callback_t
 - Firmware update module, 25
- Version module, 9
 - bwv_api_get_version, 10
 - bwv_api_version_add_callback_cb, 11
 - bwv_api_version_add_callback_cu, 11
 - bwv_api_version_add_callback_fx3, 11
 - bwv_api_version_get_cb, 12
 - bwv_api_version_get_cu, 12
 - bwv_api_version_get_fx3, 12
 - version_callback_t, 10
- version_callback_t
 - Version module, 10
- version_data_t, 39
- width
 - Gating module, 24