LED-Panel API

C++

Version 1.0.0

Documentation

Created on Fri Jul 7 2017

Contents

1 LED-Panel API documentation

The LED-Panel API offers an C++ interface for controlling LED-Panel V3/V4 devices by Image Engineering GmbH & Co. KG.

Please note that all functions are only available with LED-Panel V4.

Older versions may lack of hardware outputs that can be controlled by the API and/or the firmware cannot be updated.

We recommend to use this API with the latest version of the LED-Panel.

2 Examples

The examples section contains code samples for the usage of the LED-Panel API.

To get started we need to create LEDPanelController object(s):

```
* int error = 0;
* std::map<std::string, iQ::LEDPanel::LEDPanelController *> devices =
      iQ::LEDPanel::LEDPanelController::createObjects(error);
*
```

If everything worked correctly the error should be 0 which means "No Error occured". If an error occure error codes can be converted to plain text for debuggin purpose or inform the user that something went wrong:

See Also

error.h

```
* std::string errMsg = iQ::LEDPanel::LEDPanelController::getErrorMessage
      (error);
*
```

To get a specific device from the devices std::map we have to use the device serial number:

```
* iQ::LEDPanel::LEDPanelController *device = devices["LP40000"];
*
```

Start, stop and reset the the LEDs to initial state:

```
* device->start();
* device->stop();
* device->reset();
*
```

Changing measurment mode and frequency/period time:

See Also

iQ::LEDPanel::Mode

```
* device->setMode(iQ::LEDPanel::Mode::ResponseTime);
* device->setFrequency(0.01) //Period time 0.01s = 100Hz
*
```

If you use a iQ-Trigger athe LED-Panel starts running if you trigger the divice:

```
* device->setLedPostRollTime(2000); //With setPostRollTime() the LEDs keep running even
      if the release time of the iQ-Trigger is expired.
* device->setIQTriggerStatus(iQ::LEDPanel::iQTrigger::Mode::high, 200) //Release
      iQ-Trigger wiht full force for 200ms
*
```

3 Class Index

3.1 Class List

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

4 Class Documentation

4.1 iQ::LEDPanel::LEDPanelController Class Reference

Public Member Functions

- ∼LEDPanelController ()

  iQ::LEDPanel::LEDPanelController::∼LEDPanelController Destructor of the iQ::LEDPanelController object.
- void start (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::start Starts the LEDs of the device
- void stop (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::stop Stops the LEDs of the device
- void reset (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::reset Resets the LEDs to the initial state
- bool isRunning (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::isRunning Returns true if the LEDs of the device are running
- void setSleepMode (const bool &on, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setSleepMode Sets or wakes the device to/from sleep mode.
- void setDisplayBrightness (const int &value, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setDisplayBrightness Adjusts the display brightness.
- int getDisplayBrightness (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getDisplayBrightness Returns the current display brightness
- void setMode (LEDPanel::Mode::Mode mode, int &errorCode=default_)

iQ::LEDPanel::LEDPanelController::setMode Sets the measurement mode of the LED-Panel

- void setMode (int mode, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setMode This is an overloaded function, see iQ::LEDPanel::LEDPanelController-::setMode above.
- void setDirectionSingle (LEDPanel::Direction::LedSingle::Direction direction, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setDirectionSingle Sets the LED moving direction in the modes were only one LED lights up at a time.
- void setDirectionSingle (const int &direction, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setDirectionSingle This is a overloaded function, see iQ::LEDPanel::LEDPanel-Controller::setDirectionSingle above.
- void setDirectionMulti (LEDPanel::Direction::LedLine::Direction direction, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setDirectionMulti Sets the LED moving direction in the modes were more than one LED lights up at a time.
- void setDirectionMulti (const int &direction, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setDirectionMulti This is a overloaded function, see iQ::LEDPanel::LEDPanel-Controller::setDirectionMulti above.
- int getMode (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getMode Returns the current set mode of the LED-Panel.
- int getDirectionSingle (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getDirectionSingle Returns the current set direction of the single LED mode.
- int getDirectionMulti (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getDirectionMulti Returns the current set direction of the multi LED mode.
- void setTime (const double &time, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setTime Sets the time/frequency for the current mode.
- void setTime (LEDPanel::Time::ExposureTimeValues time, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setTime Sets the time/frequency ExposureTime mode.
- void setTime (const int &time, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setTime This is an overlaoded function, see iQ::LEDPanel::LEDPanelController-::setTime above.
- double getTime (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getTime Returns the time in seconds of the current set mode.
- void setTriggerMode (iQ::LEDPanel::Trigger::Mode mode, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setTriggerMode Sets the LED-Panel in Continuous or External mode.
- void setTriggerMode (const int &mode, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setTriggerMode An overloaded function, see iQ::LEDPanel::LEDPanelController-::setTriggerMode above.
- int getTriggerMode (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getTriggerMode Returns the selected trigger mode.
- void enableTrigger (bool enabled, iQ::LEDPanel::Trigger::Type type, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::enableTrigger Enables/Disables the selected hardware input (CameraTrigger and/or StopTrigger).
- bool isTriggerEnabled (iQ::LEDPanel::Trigger::Type type, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::isTriggerEnabled Returns the state of the specific hardware input.
- bool getTriggerState (iQ::LEDPanel::Trigger::Type type, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getTriggerState TO DOOOOO
- bool getIQTriggerStatus (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getDigitusStatus Returns the current set digitus output state.

- void setIQTriggerStatus (iQ::LEDPanel::iQTrigger::Mode mode, const int &duration, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setIQTriggerStatus Sets the mode of the iQ-Trigger output.
- void setIQTriggerStatus (const int &mode, const int &duration, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setIQTriggerStatus Overloaded function, see iQ::LEDPanel::LEDPanelController-::setIQTriggerStatus above.
- double getIQTriggerTime (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getIQTriggerTime Returns the set time for the iQ-Trigger.
- void setIQTriggerAutoRelease (bool enabled, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setIQTriggerAutoRelease Enables/Disables the iQ-Trigger automatic release.
- bool getIQTriggerAutoRelease (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getIQTriggerAutoRelease Returns if the automatic iQ-Trigger release is enabled or disabled.
- void setLedPostRollTime (const int &duration, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setLedPostRollTime Sets the time of how long after the release of the iQ-Trigger the LED-Panel keeps running.
- int getLedPostRollTime (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getLedPostRollTime Returns the set post roll time
- void setDefocusTime (const int &duration, int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::setDefocusTime Sets the time in milliseconds of how long the iQ-Defocus should be in released state.
- int getDefocusTime (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::getDefocusTime Returns the set defocus time in milliseconds
- std::string getSerialNumber (int &errorCode=default_)

  iQ::LEDPanelController::getSerialNumber returns the serial number of the connected LED-Panel.
- std::string getFirmwareVersion (int &errorCode=default_)

  iQ::LEDPanelController::getFirmwareVersion returns the current firmware version of the device.

Static Public Member Functions

- static std::map< std::string,
  LEDPanelController ∗ > createObjects (int &errorCode=default_)

  iQ::LEDPanel::LEDPanelController::createObjects Scans for connected LED-Panel and creates a map containing the device serial as key and a pointer to the LEDPanelController object as value for every LED-Panel.
- static std::string getErrorMessage (int &errorCode)

  iQ::LEDPanelController::getErrorMessage returns a plain text error message that corresponds to the given error code.
  Almost every function in this API provides the possibility to catch errors that occur during processing.
  As the returned error codes are simple integer values this function returns a detailed description of the underlying value.
  For example:

4.1.1 Member Function Documentation

4.1.1.1 std::map< std::string, iQ::LEDPanel::LEDPanelController * > iQ::LEDPanel::LEDPanelController::createObjects (int & *errorCode* = `default_`) [static]

iQ::LEDPanel::LEDPanelController::createObjects Scans for connected LED-Panel and creates a map containing the device serial as key and a pointer to the LEDPanelController object as value for every LED-Panel.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

Returns

std::map<std::string, iQ::LEDPanel::LEDPanelController∗> A std::map with LED-Panel serials as keys and the controller object as value.

```
* int error = 0;
* std::map<std::string, iQ::LEDPanel::LEDPanelController *> devices =
      iQ::LEDPanel::LEDPanelController::createObjects(error);
*
```

4.1.1.2 void iQ::LEDPanel::LEDPanelController::enableTrigger (bool *enabled*, iQ::LEDPanel::Trigger::Type *type*, int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::enableTrigger Enables/Disables the selected hardware input (CameraTrigger and/or StopTrigger).

Parameters

A	bool which enables/disables the selected input.
type	An enumeration value from iQ::LEDPanel::Trigger::Type.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

4.1.1.3 int iQ::LEDPanel::LEDPanelController::getDefocusTime (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getDefocusTime Returns the set defocus time in milliseconds

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

Returns

An int representing the time in milliseconds.

4.1.1.4 int iQ::LEDPanel::LEDPanelController::getDirectionMulti (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getDirectionMulti Returns the current set direction of the multi LED mode.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

Returns

A int representing the current set direction.

4.1.1.5 int iQ::LEDPanel::LEDPanelController::getDirectionSingle (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getDirectionSingle Returns the current set direction of the single LED mode.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A int representing the current set direction.

4.1.1.6 int iQ::LEDPanel::LEDPanelController::getDisplayBrightness (int & *errorCode* = default_)

iQ::LEDPanel::LEDPanelController::getDisplayBrightness Returns the current display brightness

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A int representing the current display brightness value.

4.1.1.7 std::string iQ::LEDPanel::LEDPanelController::getErrorMessage (int & *errorCode*) [static]

iQ::LEDPanelController::getErrorMessage returns a plain text error message that corresponds to the given error code.

Almost every function in this API provides the possibility to catch errors that occur during processing.

As the returned error codes are simple integer values this function returns a detailed description of the underlying value.

For example:

```
*  // the variable that holds the error code
*  int error = 0; \n
*  // Use any API function with error code parameter. Your error variable is passed by reference so it can
      be changed inside the function.
*  LEDPanelControllerObject.someFunction(error) \n
*  // if you pass this error code to getErrorMessage() it will return the plain text error message that
      belongs to the error code
*  std::string errorMessage = LEDPanelControllerObject.getErrorMessage(error);
*
```

_

Parameters

errorCode	Error code catched from a function call.

Returns

Plain text error message that corresponds to the given error code.

4.1.1.8 std::string iQ::LEDPanel::LEDPanelController::getFirmwareVersion (int & *errorCode* = default_)

iQ::LEDPanelController::getFirmwareVersion returns the current firmware version of the device.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

std::string Firmware version of the device.

4.1.1.9 bool iQ::LEDPanel::LEDPanelController::getIQTriggerAutoRelease (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getIQTriggerAutoRelease Returns if the automatic iQ-Trigger release is enabled or disabled.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

enabled A bool which enables or disables the automatic release.

4.1.1.10 bool iQ::LEDPanel::LEDPanelController::getIQTriggerStatus (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getDigitusStatus Returns the current set digitus output state.

If true the output is activated.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A bool indicating the state.

4.1.1.11 double iQ::LEDPanel::LEDPanelController::getIQTriggerTime (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getIQTriggerTime Returns the set time for the iQ-Trigger.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A double representing the time in seconds.

4.1.1.12 int iQ::LEDPanel::LEDPanelController::getLedPostRollTime (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getLedPostRollTime Returns the set post roll time

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

An int with the set post roll time in milliseconds.

4.1.1.13 int iQ::LEDPanel::LEDPanelController::getMode (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getMode Returns the current set mode of the LED-Panel.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A int representing the current set mode

4.1.1.14 std::string iQ::LEDPanel::LEDPanelController::getSerialNumber (int & *errorCode* = `default_`)

iQ::LEDPanelController::getSerialNumber returns the serial number of the connected LED-Panel.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

std::string The serial number of the LED-Panel.

4.1.1.15 double iQ::LEDPanel::LEDPanelController::getTime (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getTime Returns the time in seconds of the current set mode.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.16 int iQ::LEDPanel::LEDPanelController::getTriggerMode (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getTriggerMode Returns the selected trigger mode.

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A int corresponding to a enumeration value from iQ::LEDPanel::Trigger::Mode.

4.1.1.17 bool iQ::LEDPanel::LEDPanelController::getTriggerState (iQ::LEDPanel::Trigger::Type *type*, int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::getTriggerState TO DOOOOO

Parameters

type	An enumeration value from iQ::LEDPanel::Trigger::Type.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A bool indicating the state.

4.1.1.18 bool iQ::LEDPanel::LEDPanelController::isRunning (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::isRunning Returns true if the LEDs of the device are running

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A bool representing the current state of the LEDs.

4.1.1.19 bool iQ::LEDPanel::LEDPanelController::isTriggerEnabled (iQ::LEDPanel::Trigger::Type *type*, int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::isTriggerEnabled Returns the state of the specific hardware input.

Returns true if the selected input is enabled.

Parameters

type	An enumeration value from iQ::LEDPanel::Trigger::Type.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A bool indicating the state.

4.1.1.20 void iQ::LEDPanel::LEDPanelController::reset (int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::reset Resets the LEDs to the initial state

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.21 void iQ::LEDPanel::LEDPanelController::setDefocusTime (const int & *duration*, int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setDefocusTime Sets the time in milliseconds of how long the iQ-Defocus should be in released state.

Parameters

duration	The time in milliseconds.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.22 void iQ::LEDPanel::LEDPanelController::setDirectionMulti (LEDPanel::Direction::LedLine::Direction *direction*, int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setDirectionMulti Sets the LED moving direction in the modes were more than one LED lights up at a time.

E.g. Rolling shutter mode.

Parameters

direction	Sets the LED moving direction, enums are: LeftToRight = 1, RightToLeft = 2, TopToBottom = 3, BottomToTop = 4. Example: LeftToRight means that the LEDs light up column wise from left to right In the TopToBottom direction the LEDs light up row wise from top to bottom.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.23 void iQ::LEDPanel::LEDPanelController::setDirectionSingle (LEDPanel::Direction::LedSingle::Direction *direction,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setDirectionSingle Sets the LED moving direction in the modes were only one LED lights up at a time.

Parameters

direction	Sets the LED moving direction, enums are: TopLeft_LeftToRight = 1, TopRight_RightToLeft = 2, BottomLeft_LeftToRight = 3, Bottom-Right_RightToLeft = 4, TopLeft_TopToBottom = 5, TopRight_TopToBottom = 6, BottomLeft_BottomToTop = 7, BottomRight_BottomToTop = 8 Example: TopLeft_LeftToRight means that in the initial state the LED in the top left corner is on and the LED array will light up from left to right.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.24 void iQ::LEDPanel::LEDPanelController::setDisplayBrightness (const int & *value,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setDisplayBrightness Adjusts the display brightness.

Parameters

value	Can be set as int from 0 (off) to 100 (max. brightness)
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.25 void iQ::LEDPanel::LEDPanelController::setIQTriggerAutoRelease (bool *enabled,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setIQTriggerAutoRelease Enables/Disables the iQ-Trigger automatic release.

/n This is used to measure negative shutterlag/shootinglag. The LED-Panel starts running, if the 50th LED lights up /n the iQ-Trigger will be released automaticaly.

Parameters

enabled	A bool which enables or disables the automatic release.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.26 void iQ::LEDPanel::LEDPanelController::setIQTriggerStatus (iQ::LEDPanel::iQTrigger::Mode *mode,* const int & *duration,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setIQTriggerStatus Sets the mode of the iQ-Trigger output.

There are three different modes: off, low power, heigh power.

In heigh power mode the iQ-Trigger operates with maximum force.

Parameters

mode	An enumeration value from iQ::LEDPanel::iQTrigger::Mode.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

Returns

A bool indicating the state.

4.1.1.27 void iQ::LEDPanel::LEDPanelController::setIQTriggerStatus (const int & *mode,* const int & *duration,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setIQTriggerStatus Overloaded function, see iQ::LEDPanel::LEDPanel-Controller::setIQTriggerStatus above.

•

4.1.1.28 void iQ::LEDPanel::LEDPanelController::setLedPostRollTime (const int & *duration,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setLedPostRollTime Sets the time of how long after the release of the iQ-Trigger the LED-Panel keeps running.

If there is no time set the LED-Panel stops running immediately after the iQ-Trigger release time is elapsed.

Parameters

duration	The time in milliseconds
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.29 void iQ::LEDPanel::LEDPanelController::setMode (LEDPanel::Mode::Mode *mode,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setMode Sets the measurement mode of the LED-Panel

Parameters

mode	Sets the desired measurement mode, enums are: ResponseTime = 1, ExposureTime = 2, Framerate = 3, RollingShutter = 4, AllOn = 5
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.30 void iQ::LEDPanel::LEDPanelController::setSleepMode (const bool & *on,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setSleepMode Sets or wakes the device to/from sleep mode.

Parameters

on	Sets the mode to the device.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

4.1.1.31 void iQ::LEDPanel::LEDPanelController::setTime (const double & *time,* int & *errorCode* = `default_`)

iQ::LEDPanel::LEDPanelController::setTime Sets the time/frequency for the current mode.

Parameters

time	
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

4.1.1.32 void iQ::LEDPanel::LEDPanelController::setTime (LEDPanel::Time::ExposureTimeValues *time,* int & *errorCode =* `default_`)

iQ::LEDPanel::LEDPanelController::setTime Sets the time/frequency ExposureTime mode.

Parameters

time	
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

4.1.1.33 void iQ::LEDPanel::LEDPanelController::setTriggerMode (iQ::LEDPanel::Trigger::Mode *mode,* int & *errorCode =* `default_`)

iQ::LEDPanel::LEDPanelController::setTriggerMode Sets the LED-Panel in Continuous or External mode.

The External mode is used if an external event triggers the LED-Panel

e.g. a plugged remote release or a software controlling the LED-Panel.

The Continuous mode can be set if the LED-Panel is used as 'stand alone' device.

Parameters

mode	An enumeration value from iQ::LEDPanel::Trigger::Mode. Sets the selected trigger mode.
errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

4.1.1.34 void iQ::LEDPanel::LEDPanelController::start (int & *errorCode =* `default_`)

iQ::LEDPanel::LEDPanelController::start Starts the LEDs of the device

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LEDPanelController::getErrorMessage(int).

```
* //Start single device
* devices[<serialNumber>]->start();
*
* //Start all devices
*  std::map<std::string, iQ::LEDPanel::LEDPanelController *>::iterator itr;
*  for(itr = devices.begin(); itr != devices.end(); ++itr) {
*       itr->second->start();
*  }
*
* //Start all devices range based (C++11)
* for(auto device : devices.second) {
*     device.second->start();
* }
*
```

4.1.1.35 void iQ::LEDPanel::LEDPanelController::stop (int & *errorCode =* `default_`)

iQ::LEDPanel::LEDPanelController::stop Stops the LEDs of the device

Parameters

errorCode	Describes the error that occurred during processing. 0 denotes no error. You can receive a corresponding plain text error message by calling iQ::LEDPanel::LED-PanelController::getErrorMessage(int).

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

- include/LedpanelController.h
- LedpanelController.h
- LedpanelController.cpp

4.2 iQ::Version Class Reference

Static Public Member Functions

- static int getMajor ()

    iQ::Version::getMajor returns the major version number.
- static int getMinor ()

    iQ::Version::getMajor returns the minor version number.
- static int getPatch ()

    iQ::Version::getMajor returns the patch version number.
- static std::string getVersion ()

    iQ::Version::getMajor returns the API version number.
    The string is build of major, minor and patch version number seperated by a period.

4.2.1 Member Function Documentation

4.2.1.1 int iQ::Version::getMajor ( ) `[static]`

iQ::Version::getMajor returns the major version number.

E.g. API version 1.0.0 means:

- major 1
- minor 0
- patch 0

Returns

The major version number of the used API.

4.2.1.2 int iQ::Version::getMinor ( ) `[static]`

iQ::Version::getMajor returns the minor version number.

E.g. API version 1.0.0 means:

- major 1
- minor 0
- patch 0

Returns

The minor version number of the used API.

4.2.1.3 int iQ::Version::getPatch ( ) [static]

iQ::Version::getMajor returns the patch version number.

E.g. API version 1.0.0 means:

- major 1
- minor 0
- patch 0

Returns

The patch version number of the used API.

4.2.1.4 std::string iQ::Version::getVersion ( ) [static]

iQ::Version::getMajor returns the API version number.

The string is build of major, minor and patch version number seperated by a period.

Returns

The API version number.

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

- include/version.h
- version.h
- version.cpp

Index