TEMAS Documentation

Release 0.0.1

Rubu GmbH

Jul 17, 2025

TEMAS LIBRARIES

This is the official documentation for **TEMAS**, the modular 3D sensor platform by Rubu GmbH.

TEMAS integrates real-time image processing, object detection, distance measurement and control via a unified Python-based library. Whether you're using it in education, robotics, research, or prototyping – here you'll find everything to get started.

CHAPTER

ONE

GENERAL DOCUMENTATION FOR TEMAS

- **Library Reference** All available functions, classes, and modules of the TEMAS Python library
- **Temas Use Cases** Real-world applications in research, robotics, education and logistics

CHAPTER

TWO

LIBRARY MODULES

- **temas.connect** – This class alows the Temas connection.
- **temas.camera** – This class allows the control of camera using urllib.
- **temas.control** – This class alows the control of motors and laser.
- **temas.common** – This class included the common settings of Temas.

2.1 TEMAS Libraries

Connect	This class alows the Temas connection.
Camera	This class allows the control of camera using urllib.
Control	This class alows the control of motors and laser.
Common	This class included the common settings of Temas

2.1.1 temas.Connect

class Connect(*hostname='temas'*, *ip_address=''*)

Bases: object

This class alows the Temas connection.

__init__(*hostname='temas'*, *ip_address=''*)

Initial values of class Connect.

Methods

__init__([hostname, ip_address])	Initial values of class Connect.
get_ip_from_hostname(hostname[, max_attempts])	Attempts to determine the IP address of the hostname.

get_ip_from_hostname(*hostname*, *max_attempts=4*)

Attempts to determine the IP address of the hostname.

2.1.2 temas.Camera

class Camera(*port=8081*, *ip_address=''*)

Bases: object

This class allows the control of camera using urllib.

__init__(*port=8081*, *ip_address=''*)

Initializes the Camera class

Methods

__init__([port, ip_address])	Initializes the Camera class
get_frame()	Returns the most recent frame
set_brightness(brightness_percent)	Set brightness of visual camera 0-100 :return: Brightness :rtype: String
set_contrast(contrast_percent)	Set contrast of visual camera 0-100 :return: Contrast :rtype: String
set_exposure_time(microseconds)	Set exposure time of visual camera 50-33000µs :return: Exposure time :rtype: String
set_gain(gain_value)	Set gain of visual camera 1.123-16.0 :return: Gain :rtype: String
set_lens_position(lens_position_percent)	Set lens position of visual camera 0-100 :return: Lens position :rtype: String
set_saturation(saturation_percent)	Set saturation of visual camera 0-100 :return: Saturation :rtype: String
start_thread()	Starts the thread to fetch frames.
stop_thread()	Stops the thread.
update()	Continuously fetches frames from the MJPEG stream.

update()

Continuously fetches frames from the MJPEG stream.

start_thread()

Starts the thread to fetch frames.

stop_thread()

Stops the thread.

get_frame()

Returns the most recent frame

Returns
Frame

Return type
numpy.ndarray or None

set_exposure_time(*microseconds*)

Set exposure time of visual camera 50-33000µs :return: Exposure time :rtype: String

set_brightness(*brightness_percent*)

Set brightness of visual camera 0-100 :return: Brightness :rtype: String

set_saturation(*saturation_percent*)

Set saturation of visual camera 0-100 :return: Saturation :rtype: String

set_contrast(*contrast_percent*)

Set contrast of visual camera 0-100 :return: Contrast :rtype: String

set_lens_position(*lens_position_percent*)

Set lens position of visual camera 0-100 :return: Lens position :rtype: String

set_gain(*gain_value*)

Set gain of visual camera 1.123-16.0 :return: Gain :rtype: String

2.1.3 temas.Control

class Control(*port=8082*, *ip_address=''*)

Bases: `object`

This class alows the control of motors and laser.

__init__(*port=8082*, *ip_address=''*)

Methods

__init__([port, ip_address])	
distance()	Laser distance measurment
get_pcl(path)	Retrieve the latest point cloud from the laser scan (e.g., located at C:/temp/).
get_point_cloud_scan_percent()	Point cloud scan status in perecent
get_pos()	Get motor positions
mean_distance()	Mean laser distance measurement
move(direction)	Sends a movement command to the motor.
move_down()	Move motor down in 1° steps
move_down_fine()	Move motor down in 0.5° steps
move_home()	Move to home position
move_left()	Move motor left in 1° steps
move_left_fine()	Move motor left in 0.5° steps
move_pos(phi, theta)	Move to the specified pan and tilt angles in degrees (e.g., 12.5, 11) with a step size of 0.5°.
move_right()	Move motor right in 1° steps
move_right_fine()	Move motor right in 0.5° steps
move_threaded(direction)	Starts the movement in a separate thread.
move_up()	Move motor up in 1° steps
move_up_fine()	Move motor up in 0.5° steps
start_point_cloud_scan(theta1, theta2, phi1, ...)	Starts a point cloud scan between the specified pan and tilt angle ranges.
stop_point_cloud_scan()	Stop point cloud scan

get_pcl(*path*)

Retrieve the latest point cloud from the laser scan (e.g., located at C:/temp/).

Returns
.ply file

`distance()`

Laser distance measurment

Returns
Value of the distance [cm]

Return type
String

`mean_distance()`

Mean laser distance measurement

Returns
Value of the mean_distance [cm]

Return type
String

`move_pos(phi, theta)`

Move to the specified pan and tilt angles in degrees (e.g., 12.5, 11) with a step size of 0.5°.

Returns
Status message indicating if the move was successful. Returns True if successful.

Return type
String

`get_pos()`

Get motor positions

Returns
pan, tilt positions [°]

Return type
String

`move_home()`

Move to home position

`move(direction)`

Sends a movement command to the motor.

`move_threaded(direction)`

Starts the movement in a separate thread.

`move_right()`

Move motor right in 1° steps

`move_left()`

Move motor left in 1° steps

`move_up()`

Move motor up in 1° steps

`move_down()`

Move motor down in 1° steps

move_right_fine()

Move motor right in 0.5° steps

move_left_fine()

Move motor left in 0.5° steps

move_up_fine()

Move motor up in 0.5° steps

move_down_fine()

Move motor down in 0.5° steps

start_point_cloud_scan(*theta1*, *theta2*, *phi1*, *phi2*, *color=1*)

Starts a point cloud scan between the specified pan and tilt angle ranges.

Parameters

- **theta1** – Minimum elevation angle in degrees (-30 to 90, step size 0.5°)
- **theta2** – Maximum elevation angle in degrees (-30 to 90, step size 0.5°)
- **phi1** – Minimum azimuth angle in degrees (-60 to 60, step size 0.5°)
- **phi2** – Maximum azimuth angle in degrees (-60 to 60, step size 0.5°)
- **color** – Color mode (default = 1)

Returns

Status message indicating if the scan was started successfully. Returns True if successful.

Return type

String

stop_point_cloud_scan()

Stop point cloud scan

Returns

Status message indicating if the scan was stopped successfully. Returns True if successful.

Return type

String

get_point_cloud_scan_percent()

Point cloud scan status in perecent

Returns

Point cloud scan status [%]

Return type

String

2.1.4 temas.Common

class Common(*port=8083*, *ip_address=''*)

Bases: object

This class included the common settings of Temas

__init__(*port=8083*, *ip_address=''*)

Initial values of class Common

Methods

__init__([port, ip_address])	Initial values of class Common
get_fw_version()	Get firmware version of Temas
get_hostname()	Get hostname of Temas
get_ip()	Get ip address of Temas
get_laser_x_los()	Get the x value los of Laser in pixels relative to the visual frame.
get_laser_y_los()	Get the y value los of Laser in pixels relative to the visual frame.
get_mac()	Get the mac address of Temas
get_port_camera()	Get port of Temas visual camera
get_port_common()	Get port of Temas common settings
get_port_control()	Get port of control device
get_port_tof_camera()	Get port of Temas tof camera
get_sn()	Get serial number of Temas
get_temperature()	Read Temperature of Temas
near_mode_off()	tof near mode off
near_mode_on()	tof near mode on
restart()	Restart Temas
set_direct_static_ip(ip)	static ip /24 without dhcp :param ip: IP address
set_hostname(hostname)	Set hostname of Temas
set_static_ip(ip, gateway, dns)	static ip /24 dhcp :param ip: IP address :param gateway: Default gateway for the network :param dns: DNS server address
shutdown()	Shutdown Temas

shutdown()

Shutdown Temas

Returns
Status message indicating if the shutdown was successful. Returns True if successful.

Return type
String

restart()

Restart Temas

Returns
Status message indicating if the restart was successful. Returns True if successful.

Return type
String

get_temperature()

Read Temperature of Temas

Returns
Temperature Temas [°C]

Return type
String

get_ip()

Get ip address of Temas

Returns
Ip address

Return type
String

`get_sn()`
Get serial number of Temas

Returns
Serial number is returned

Return type
String

`get_fw_version()`
Get firmware version of Temas

Returns
Firmware version is returned

Return type
String

`get_hostname()`
Get hostname of Temas

Returns
Hostname is returned

Return type
String

`set_hostname(hostname)`
Set hostname of Temas

`get_port_camera()`
Get port of Temas visual camera

Returns
Port from visual camera

Return type
String

`get_port_tof_camera()`
Get port of Temas tof camera

Returns
Port from tof camera

Return type
String

`get_port_common()`
Get port of Temas common settings

Returns
Port from Temas common settings

Return type
String

`get_port_control()`

Get port of control device

Returns
Port from control device

Return type
String

`get_mac()`

Get the mac address of Temas

Returns
Returns the mac address

Return type
String

`get_laser_x_los()`

Get the x value los of Laser in pixels relative to the visual frame.

Returns
Returns the x value los (px)

Return type
String

`get_laser_y_los()`

Get the y value los of Laser in pixels relative to the visual frame.

Returns
Returns the y value los

Return type
String

`set_static_ip(ip, gateway, dns)`

static ip /24 dhcp :param ip: IP address :param gateway: Default gateway for the network :param dns: DNS server address

`set_direct_static_ip(ip)`

static ip /24 without dhcp :param ip: IP address

`near_mode_on()`

tof near mode on

`near_mode_off()`

tof near mode off

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex

INDEX