Inovo API

version

Inovo

March 19, 2021

Contents

Welcome to Inovo API’s documentation!

Inovo API

Submodules

gripper_client

class commander_api.gripper_client.**GripperClient** (ns='default')
Class to controll gripper connected to the robot

Parameters: **ns** (*string, optional*) – namespace, to define the name of the gripper, defaults to ‘default’

activate ()
Activate the gripper before moving it

Raises: **Exception** – Gripper activation error

get_position ()
Return current position of the gripper

Returns: gripper position
Return type: float

get_state ()
Return information about activation state of the gripper

Returns: activated
Return type: bool

move (target_position**,** effort**)**
Move the gripper to the given goal, locks till target is reached or gripper is stalled

Parameters:
- **target_position** (*float*) – the desired position of the gripper, from 0 to 1
- **effort** (*float*) – the effort used to move the fingers, from 0 to 1

Raises: **Exception** – Gripper action error
Returns: True if the target position is not reached, False if the target is reached
Return type: Bool

io_client

class commander_api.io_client.**IOClient**
Class used to control the digital input/output on the robot writst and PSU box

psu_analog_read (channel**)**
Get the value of the PSU analog input

Parameters: **channel** (*int*) – channel to read
Returns: Analog input value
Return type: float

psu_analog_write (channel**,** value**)**
Write to the PSU analog output

Parameters:
- **channel** (*int*) – channel to write
- **value** (*float*) – output value

psu_digital_read (channel)
Get the value of the PSU digital input

Parameters: **channel** (*int*) – channel to read
Raises: **Exception** – [description]
Returns: value read by the PSU
Return type: int

psu_digital_write (channel, value)
Write to the PSU digital output

Parameters:
- **channel** (*int*) – channel to write
- **value** (*int*) – output value

Raises: **Exception** – PSU digital write error

wrist_digital_read (channel)
Get the read of the wrist digital input

Parameters: **channel** (*int*) – channel to read
Raises: **Exception** – Wrist digital read error
Returns: value read by the wrist
Return type: int

wrist_digital_write (channel, value)
Write to the wrist digital output

Parameters:
- **channel** (*int*) – channel to write
- **value** (*int*) – output value

Raises: **Exception** – Wrist digital write error

motion_control_client

class commander_api.motion_control_client.**MotionControlClient**
Class used to send motion commands to the robot

get_joint_angles ()
Get the current joint positions

Raises: **Exception** – Failed to get the joint angles
Returns: Joint positions (rad)
Return type: list(float)

get_tcp_pose ()
Get the current position of the tcp in base frame

Returns: tcp position
Return type: TransformFrame

movej (target, spd=0.25, acc=0.5)
Joint move to a cartesian target, locks till the target is reached

Parameters:
- **target** (*TransformFrame*) – target position x, y, z in meters and rotation rx, ry, rz in radians
- **spd** (*float, optional*) – Tool speed in m/s, defaults to 0.25
- **acc** (*float, optional*) – Tool acceleration in m/s/s, defaults to 0.5

Raises: **RuntimeError** – Unable to execute the move

movej_angle (joint_angles**,** joint_velocity=0.4**,** joint_acceleration=0.75**)**
Move to a joint space target, locks till the target is reached

Parameters:
- **joint_angles** (*list(float)*) – joint angles in radians
- **joint_velocity** (*float, optional*) – maximum joint velocity in rad/s, defaults to 0.4
- **joint_acceleration** (*float, optional*) – maximum joint acceleration in rad/s/s, defaults to 0.75

Raises: **RuntimeError** – Unable to execute the move

movel (target**,** spd=0.25**,** acc=0.5**)**
Linear move to a cartesian target, locks till the target is reached

Parameters:
- **target** (*TransformFrame*) – target position x, y, z in meters and rotation rx, ry, rz in radians
- **spd** (*float, optional*) – Tool speed in m/s, defaults to 0.25
- **acc** (*float, optional*) – Tool acceleration in m/s/s, defaults to 0.5

Raises: **RuntimeError** – Unable to execute the move

psu_client

class commander_api.psu_client.**PSUClient**
Class used to control the PSU box

get_bus_current ()
Get the current of the bus

Returns: bus current
Return type: float

get_bus_state ()
Get the bus power status

Raises: **Exception** – Unknown psu state
Returns: state, bus on = 1, bus off = 0
Return type: int

get_bus_voltage ()
Get the voltage of the bus

Returns: bus voltage
Return type: float

power_off ()
Robot bus power off

Raises: **Exception** – Error powering off the robot

power_on ()

Robot bus power on

Raises: **Exception** – Error powering on the robot

shut_down ()
Turn off the PSU

robot_client

class commander_api.robot_client.**RobotClient**
Robot client class used for basic controll of the robot functionality

clear_errors ()
Clear all errors

Raises: **Exception** – Failed to clear errors

disable ()
Disable the robot

Raises: **Exception** – Failed to disable the robot

enable ()
Enable the robot when it is powered on

Raises: **Exception** – Failed to enable the robot

get_errors ()
Print the latest error message

Returns: Error message
Return type: ErrorMessage

sequencer_client

class commander_api.sequencer_client.**SequencerClient**
Class to control the sequencer

get_sequencer_state ()
Get the current state of the sequencer

Returns: Sequencer state, Idle = 0, Running = 1, Paused = 2
Return type: int

load_project (name**)**
Load an existing project

Parameters: **name** (*string*) – project name
Raises: **Exception** – Sequence failed to load a project

pause ()
Pause the running sequence

Raises: **Exception** – Sequence failed to pause

start (name='', variable_names=[], variable_values=[]**)**
Start the sequence or function

Parameters:
- **name** (*str, optional*) – function name, defaults to the start block
- **variable_names** (*list, optional*) – arguments to pass in the function, defaults to []
- **variable_values** (*list, optional*) – values of the arguments passed to the function, defaults to []

Raises: **Exception** – Sequencer failed to start

stop ()
Stop the running sequence

Raises: **Exception** – Sequence failed to stop

Datatypes

class commander_api.custom_datatypes.**ErrorMessage**
Error message class with error code, source and message attributes

class commander_api.custom_datatypes.**TransformFrame**
Transform frame class with x,y,z position and rx, ry,rz rotation atributes

from_quaternion (q1**,** q2**,** q3**,** q4**)**

get_quaternion ()

Guide

Example guide

Indices and tables

- **genindex**
- **modindex**
- **search**

Index

R

S

T

W

Python Module Index

c