Robot Web Services

This section covers the communication between Python and *RobotWare*. A package, rwsuis, is pip installable and includes all functions provided in this section.

RWS Class

Take full control of ABB robots through HTTP requests, made easy with the RWS class. Robot operating mode should be automatic.

```
>>> robot = RWS.RWS(base_url='robot_IP', username='user', password='pass')
>>> robot.request_mastership()
>>> robot.motors_on()
Robot motors turned on
>>> robot.start_rapid()
RAPID execution started from main
```

class **RWS**(*base_url*, *username*, *password*)

motors_on(*self*)

Sends a request to turn the robot's motors on. Mastership is required. Prints a message to the console stating whether or not the motors were in fact turned on.

motors_off(*self*)

Sends a request to turn the robot's motors off. Mastership is required. Prints a message to the console stating whether or not the motors were in fact turned off.

request_mastership(*self*)

Requests mastership over controller in automatic mode. For mastership in manual mode, see **request_rmmp()**.

release_mastership(*self*)

Releases mastership over controller.

request_rmmp(*self*)

Requests RMMP (Request Manual Mode Privileges). The request needs to be accepted within 10 seconds on controller. For mastership in automatic mode, see **request_mastership()**.

cancel_rmmp(*self*)

Cancels held or requested RMMP.

reset_pp(*self*)

Resets RAPID program pointer to main procedure. Prints a message to the console stating whether or not the request was successful.

start_RAPID(*self*)

Resets RAPID program pointer to main procedure, and starts RAPID execution. Prints a message to the console stating whether or not the request was successful.

stop_RAPID(*self*)

Stops RAPID execution. Prints a message to the console stating whether or not the request was successful.

get_rapid_variable(*self*, *var*)

Get the raw value of any variable in RAPID.

v: latest

Returns: A number if RAPID variable is ‘num’
Returns: A string if RAPID variable is not ‘num’

set_rapid_variable(*self*, *var*, *value*)

Sets the value of any variable in RAPID. Unless the variable is ‘num’, value has to be a string.

Parameters:
- **var** (*str*) – Name of variable as declared in RAPID
- **value** (*int, float or str*) – Desired variable value

set_robtarget_translation(*self*, *var*, *trans*)

Sets only the translational data of a robtarget variable in RAPID.

Parameters:
- **var** (*str*) – Name of robtarget variable as declared in RAPID
- **trans** (*int[]*) – Translational data [x,y,z]

set_robtarget_rotation_z_degrees(*self*, *var*, *rotation_z_degrees*)

Updates the orientation of a robtarget variable in RAPID by rotation about the z-axis in degrees. 0 degrees gives the Quaternion [0,1,0,0].

Parameters:
- **var** (*str*) – Name of robtarget variable as declared in RAPID
- **rotation_z_degrees** (*int*) – Rotation in degrees

set_robtarget_rotation_quaternion(*self*, *var*, *rotation_quaternion*)

Updates the orientation of a robtarget variable in RAPID by a Quaternion.

Parameters:
- **var** (*str*) – Name of robtarget variable as declared in RAPID
- **rotation_quaternion** (*tuple*) – Wanted robtarget orientation. Must be a Quaternion (tuple of length 4)

get_robtarget_variables(*self*, *var*)

Gets translational and rotational data of a robtarget variable in RAPID

Parameters: **var** (*str*) – Name of robtarget variable as declared in RAPID
Returns: Translational data of robtarget [x,y,z]
Returns: Rotational data of robtarget (Quaternion: [w,x,y,z]).

get_gripper_position(*self*)

Gets translational and rotational of the UiS tool ‘tGripper’ with respect to the work object ‘wobjTableN’.

Returns: Translational data of gripper [x,y,z]
Returns: Rotational data of gripper (Quaternion: [w,x,y,z])

get_gripper_height(*self*)

Uses **get_gripper_position()** to get the height of the UiS tool ‘tGripper’ above the work object ‘wobjTableN’.

set_rapid_array(*self*, *var*, *value*)

Sets the values of a num array variable in RAPID. The length of the num array must match the length of the array from Python.

Parameters:
- **var** (*str*) – Name of variable as declared in RAPID.
- **value** (*int[]*) – Array to be sent to RAPID.

wait_for_rapid(*self*, *var='ready_flag'*)

Polls a boolean variable in RAPID every 0.1 seconds. When the variable is TRUE, Python resets it and continues.

Parameters: **var** (*str*) – Name of boolean variable as declared in RAPID.

v: latest

set_zonedata(*self*, *var*, *zonedata*)

Set the value for a zonedata variable in RAPID. Mastership is required.

Parameters:
- **var** (*str*) – Name of variable as declared in RAPID.
- **zonedata** (*int*) – desired zonedata value.

set_speeddata(*self*, *var*, *speeddata*)

Set the value [int] for a speeddata variable in RAPID. Mastership is required.

Parameters:
- **var** (*str*) – Name of variable as declared in RAPID.
- **speeddata** (*int*) – Desired speeddata value.

set_speed_ratio(*self*, *speed_ratio*)

Set the speed ratio of the robot. Mastership is required. speed_ratio: desired speed ratio in percent [1-100].

is_running(*self*)

Uses `get_execution_state()` to check if RAPID execution is running or stopped. Returns True if running and False if stopped.

get_execution_state(*self*)

Polls the RAPID execution state.

Returns: ‘running’ or ‘stopped’

v: latest