

ElephantROBOTICS

**Tutorial of Communicate with Elephant Robot through
Socket**

1. Overview

Elephant robot allow users to control the robot from remote, one way is use socket. We use tcp protocol to communicate between the client and the robot, you can send the formatted string through tcp to get or set some property/state of the robot, the format for each function are introduced as bellow.

2. Socket String format rules

2.1 get current angles of robot

Socket string format: `get_angles()`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `get_angles:[0.174058, 0.520382, -0.07874, 0.092855, 0.0, 0.030356]`. If any error occurred, `InvalidAngles()`(defined as `[-1.0, -2.0, -3.0, -4.0, -1.0, -1.0]`) will be returned.

2.2 set the angles of robot

Socket string format: `set_angles(joint1_angle, joint2_angle, joint3_angle, joint4_angle, joint5_angle, joint6_angle, speed)`

Example: `set_angles(10.0,11.0,12.2,12.3,.11.1,16.0,500)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `set_angles:[ok]`. If any error occurred, you will get `set_angles:error_message`.

2.3 set the angle of one joint

Socket string format: `set_angle(joint,angle, ,speed)`

example: `set_angle(J1,50.5,500)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `set_angle:[ok]`. If any error occurred, you will get `set_angle:error_message`.

2.4 get current coordinates of robot

Socket string format: `get_coords()`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `get_coords:[0.174058, 0.520382, -0.07874, 0.092855, 0.0, 0.030356]`. If any error occurred, `InvalidCoords()`(defined as `[-1.0, -2.0, -3.0, -4.0, -1.0, -1.0]`) will be returned.

2.5 set the coordinates of robot

Socket string format: `set_coords(axis_x_coord, axis_y_coord, axis_z_coord, axis_rx_coord, axis_ry_coord, axis_rz_coord, speed)`

Example: `set_coords(10.0,11.0,12.2,12.3,.11.1,16.0,500)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `set_coords:[ok]`. If any error occurred, you will get `set_coords:error_message`.

2.6 set the coordinate of one axis

Socket string format: `set_coord(axis,coordinate ,speed)`

example: `set_coord(x,50.5,500)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `set_coord:[ok]`. If any error occurred, you will get `set_coord:error_message`.

2.7 get the signal of digital out pin

Socket string format: `get_digital_out(pin_number)`

example: `get_digital_out(1)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `get_digital_out:1`. If any error occurred, you will get `get_digital_out:error_message`.

2.8 set the signal of digital out pin

Socket string format: `set_digital_out(pin_number,signal)`

example: `set_digital_out(1,1)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `set_digital_out:[ok]`. If any error occurred, you will get `set_digital_out:error_message`.

2.9 get the signal of digital in pin

Socket string format: `get_digital_in(pin_number)`

example: `get_digital_in(1)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `get_digital_in:1`. If any error occurred, you will get `get_digital_in:error_message`.

2.10 set the signal of analog out pin

Socket string format: `set_analog_out(pin_number,signal)`

example: `set_digital_out(1,1.5)`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `set_analog_out:[ok]`. If any error occurred, you will get `set_analog_out:error_message`.

2.11 change the coordinate of one axis in one direction continuously

Socket string format: `jog_coord(axis,direction,speed)`

example: `jog_coord('x', 1, 500)`

The direction can be -1, 0, 1, -1 means in negative direction, 1 means positive direction, 0 means stop.

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `jog_coord:[ok]`. If any error occurred, you will get `jog_coord:error_message`.

2.12 change the angle of one joint in one direction continuously

Socket string format: `jog_angle(joint,direction,speed)`

example: `jog_coord('J1', 1, 500)`

The direction can be -1, 0, 1, -1 means in negative direction, 1 means positive direction, 0 means stop.

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `jog_angle:[ok]`. If any error occurred, you will get `jog_angle:error_message`.

2.13 enable the system

Socket string format: `state_on()`

example: `state_on()`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `state_on:[ok]`. If any error occurred, you will get `state_on:error_message`.

2.14 disable the system

Socket string format: `state_off()`

example: `state_off()`

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `state_off:[ok]`. If any error occurred, you will get `state_off:error_message`.

2.15 stop the task

Socket string format: task_stop ()

example: task_stop()

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: task_stop:[ok]. If any error occurred, you will get state_off:error_message.

2.16 set feed rate

Socket string format: set_feed_rate(speed)

example: set_feed_rate(50.0)

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success, it will return: set_feed_rate: 0. otherwise it means failed.

2.17 make the robot 'sleep' for some seconds

Socket string format: wait(seconds)

example: wait(10.5)

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success, it will return: wait:[ok]. This function will make the robot 'sleep ' for given seconds, seem likes use sleep in your code.

2.18 mount the robot upside down

Socket string format: set_upside_down(up_dn)

example: set_upside_down(1)

1 means upside down, 0 means not.

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: set_upside_down:[ok]. If any error occurred, you will get set_upside_down:error_message.

2.19 power on the robot

Socket string format: power_on()

example: power_on()

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: power_on:[ok]. If any error occurred, you will get power_on:error_message

2.20 power off the robot

Socket string format: power_off()

example: power_off()

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: power_off:[ok]. If any error occurred, you will get power_off:error_message

2.21 get the speed the robot

Socket string format: get_speed()

example: get_speed()

the speed unit is mm/s

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: get_speed:500. If any error occurred, you will get get_speed:error_message

2.22 check the state of the robot

Socket string format: state_check()

example: state_check()

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if the robot is in normal state, you will get:

state_check:1, if the robot is not in normal state, you will get: state_check:0. If any error occurred, you will get get_speed:error_message

2.23 check if the robot is running

Socket string format: check_running()
example: check_running()

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if the robot is running, you will get: check_running:1, if the robot is not running, you will get: check_running:0. If any error occurred, you will get get_speed:error_message

2.24 set the torque limit of the robot

Socket string format: set_torque_limit(axis,torque)
example: set_torque_limit(x,10.0)
the axis can be x,y or z
the unit of torque is N

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: set_torque_limit:[ok]. If any error occurred, you will get set_torque_limit:error_message.

2.25 open a g_code formatted text file

Socket string format: program_open(file_path_name)
example: program_open(/usr/a.txt)

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: program_open:0. If any error occurred, you will get program_open:error_message.

2.26 run a g_code formatted text file from the given line

Socket string format: program_run(line_number)
example: program_run(0)

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: program_run:0. If any error occurred, you will get program_run:error_message.

2.26 get the robot error

Socket string format: read_next_error()
example: read_next_error()

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: read_next_error:message.

2.27 set the payload of the robot

Socket string format: set_payload(payload)
example: set_payload(5.0)
the unit of payload is kg

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: set_payload:[ok]. If any error occurred, you will get set_payload:error_message.

2.28 set the acceleration of the robot

Socket string format: set_acceleration(acc)
example: set_acceleration(50)

the acceleration must be an integer, the unit of acceleration is mm/s².

the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: set_acceleration:[ok]. If any error occurred, you will get set_acceleration:error_message.

2.29 get the acceleration of the robot

Socket string format: `get_acceleration()`
example: `get_acceleration()`
the unit of acceleration is mm/s^2
the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: `get_acceleration:50`.

2.30 assign variable

Socket string format: `assign_variable('variable_name',value)`
example: `assign_variable('A',10)` or `assign_variable('B',"ABC")`
the variable name need to be quoted with single quote mark(''), if the value is a string,
need to be quoted with double quotation marks("").
the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: `assign_variable:[ok]`. If any error occurred, you will get `assign_variable:error_message`.

2.31 get the value of a variable

Socket string format: `get_variable('variable_name')`
example: `get_variable('A',10)`
the variable name need to be quoted with single quote mark('')
the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, like this: `get_variable:10`. If any error occurred, you will get `get_variable:error_message`.

2.32 wait for command done

Socket string format: `wait_command_done()`
example: `wait_command_done()`
this function will wait until the previous command finish.
the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: `wait_command_done:0`. If any error occurred, you will get `wait_command_done:error_message`.

2.32 pause the program

Socket string format: `pause_program()`
example: `pause_program()`
this function will pause the running program.
the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: `pause_program:[ok]`. If any error occurred, you will get `pause_program:error_message`.

2.33 resume the program

Socket string format: `resume_program()`
example: `resume_program()`
this function will resume the paused program.
the return string is formatted in key-value pair, the key is the function name, the value is the value from robot, if success: `resume_program:[ok]`. If any error occurred, you will get `resume_program:error_message`.