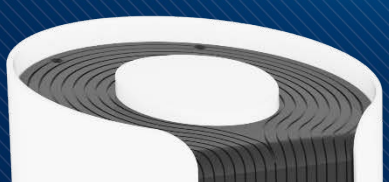
Z-Arm PC Terminal
Dynamic Link Library

Huiling-tech Robotic Co., Ltd



PC Terminal Dynamic Link Library of Z-Arm

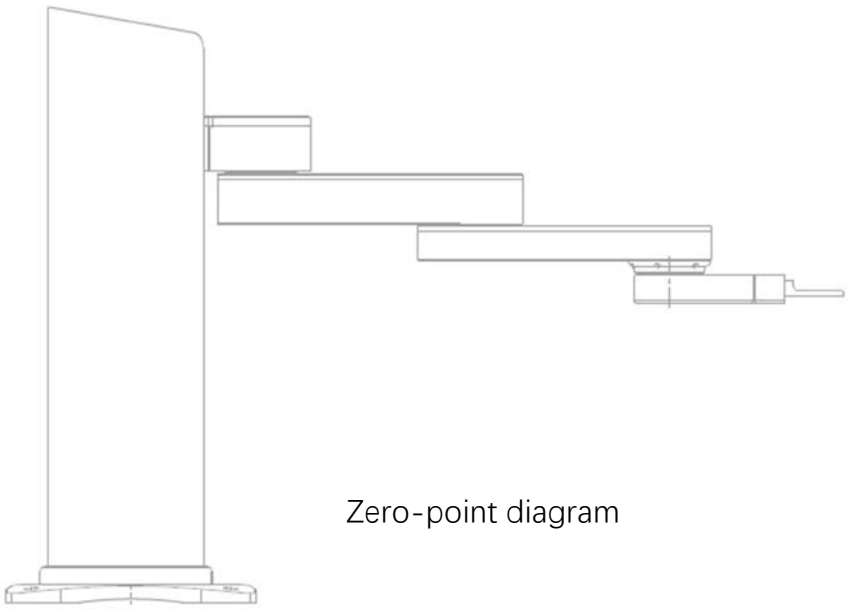
Technical support mailbox: hitbot@hitbot.cc Reply in 24 hours

Before you start, you need to know the basic information below:

When it is controlled by PC, the coordinate system of the robotic arm is defined as follows:

Axis 1 of this robotic arm is defined as robotic arm joint 1, and positive direction is counterclockwise seen from the top downward, as shown in the figure; axis 2 is defined as joint 2, and positive direction is counterclockwise seen from the top downward; axis 3 is defined as joint 3, and the positive direction is vertically upward; the distance from joint1 to joint 2 is 200mm, the distance from joint 2 to the end axis is 200mm, and the upper and lower stroke of axis 3 is 210/310mm; axis 4 is defined as joint 4.

The X axis positive direction of robotic arm is defined as the direction of the robotic arm body to the front, as shown in the figure; the Y axis positive direction is defined as facing the robotic arm body, towards the right direction from the body, as shown in Figure 1; the Z axis positive direction is vertically upward of the joint 1 axis direction, as shown in the figure; the Z axis zero point is defined in the position that the robotic arm runs to the highest position, namely the position after initialization. (If it is the first initialization of the robotic arm after being electrified exactly, the robotic arm axis 1 and axis 2 are zero, and the arm is stretched, so the coordinates of the arm end is (400,0,0)



Zero-point diagram

Figure 1

1. Preparations

This version of the dynamic link library is applicable to the PC terminal to control the robotic arm through network cables, which contains the contents of cable connection communication and robotic arm motion interpolation control; the user can use the host-computer programs produced by himself to directly control the tasks such as initialization, setting and motion control, etc. of the robotic arm through the interface functions we defined.

The user needs to prepare the following matters for the control of the robotic arm by this dynamic link library:

(1) In scenario of single robotic arm control, the PC end is directly connected to the robotic arm through the network cable, and the IP address of the PC end needs to be set as parameters in the figure below(Figure 2); if it is inconvenient to change the IP parameters at the PC end, you can search through the mobile phone side and change the IP parameters of the robotic arm (Figure 3)

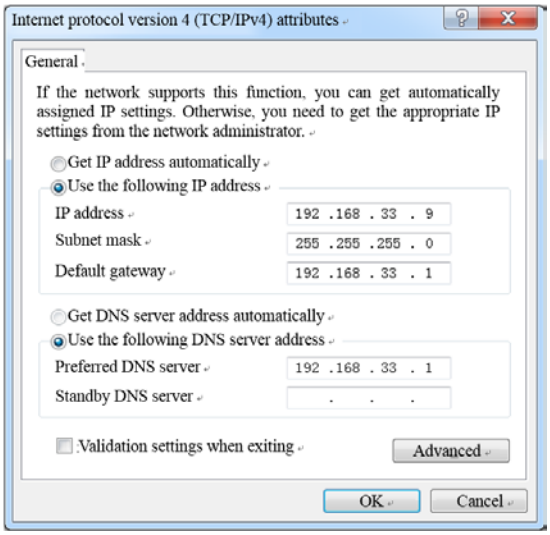


图 2

图 3

(2) In scenario of simultaneous control of multiple robotic arms, you need to control through routing;connect the computer and routing through cables, and connect the routing and the robotic arm through cables; at this time, you need to query and change IP parameters of the robotic arm through the mobile phone terminal (arm A in Figure 4, arm B in Figure 5), and the IP of different robot arms should be set under the same gateway, and remain no conflict.

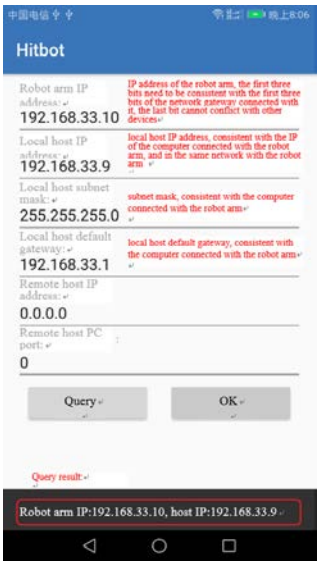


Figure 5

Figure 6

(3) Close the firewall on the PC end.

2.Function overview

Multi machine dynamic link library can connect with multiple robotic arms simultaneously, and achieve the function of one PC notifying and controlling multiple robotic arms, including two library files and one exe executable file.

small_scara_interface.dll

small_scara_interface.lib

small_scara_interface.h

share.dll

server.exe

3.Function description in Small_scara_interface.dll library

No.	Function Declaration	Incoming Parameter	Return Value
1	int initial(int generation, float z_travel) Initial parameters corresponding to the model Default J1 joint arm length is 200mm Default J2 joint arm length is 200mm.	1) int generation. =1 Z-Arm low configuration =2 Z-Arm high configuration 2) float z_travel Set the up and down stroke to be 210 or 310 according to the actual model, and you need to pass positive value;	=0 communication has not yet been established,this initialization is unsuccessful; =1 initializing; =2 generation parameter error; =3 encoder value error; =11 controlled by the mobile terminal, this initialization is not successful =12 z_travel transmission error

Following :

No.	Function Declaration	Incoming Parameter	Return Value
2	void get_scara_param(float *x,float *y,float*z, float *angle1, float *angle2, float *rotation, bool *communicate_success, bool *initial_finish, bool *servo_off_flag, bool *move_flag)	1) float *x, coordinate value of x (mm) 2) float *y, coordinate value of y (mm) 3) float *z, coordinate value of z (mm) 4) float *angle1, angle value of joint 1 (deg) 5) float *angle2, angle value of joint 2 (deg) 6) float * rotation, angle value of joint 4 (deg) 7) bool *communicate_success, =0 communication has not been connected =1 communication has been established 8) bool *initial_finish, =0 initialization successful =1 initialization unsuccessful 9) bool *servo_off_flag, =0 servo not closed =1 servo closed 10) bool *move_flag, =0 the robot arm is in standby state =1 the robot arm is in motion state	No Return Value
3	void net_port_initial_auto() initialize server	No Incoming Parameter	No Return Value
4	void set_arm_length(float l1, float l2) Set J1 J2 joint arm length	float l1 J1 joint length, reserved parameters, you must introduce 200, float l2 J1 joint length, introduce 200 generally for those with J4 rotation joint, and set according to the actual situation for those with no J4 rotation joint	No Return Value
5	int change_attitude(float speed) Change attitude	float speed The joint speed (deg/s) when transforming attitude, and the difference of the joint angles between the two attitudes will be judged. At the same time, divided by speed to get the motion time of each joint, and the longer time is the final movement time	=0 the robot arm is running other instructions, this command is invalid =1 this command goes into effect, and the robot arm begins to move =2 the incoming speed is less than or equal to 0 =3 not initialized yet =4 can't reach by the other attitude =6 servo not opened =11 mobile terminal is controlling

Following :

No.	Function Declaration	Incoming Parameter	Return Value
6	int single_axis_move(int axis, float distance, float speed, bool interpolation); Uniaxial motion	int axis Enter 1 or 2 or 3 or 4, corresponding to joint 1- joint 4 respectively float distance Moving distance relative to current position, When axis=3, distance unit is mm;when axis=1 or 2 or 4, distance unit is deg float speed Moving speed, When axis=3, speed unit is mm/s;when axis=1 or 2 or 4, speed unit is deg/s bool interpolation, Interpolation method, =0, polynomial interpolation =1, linear interpolation	=0 the robotic arm is running other instructions, this command is invalid =1 this command goes into effect, and the robotic arm begins to move =2 setting speed is less than or equal to zero =3 not initialized yet =4 cannot reach the target position =5 output shaft number parameter error =6 the robotic arm servo not opened =11 mobile terminal is controlling
7	inttrail_move(intpoint_number, float *x, float *y, float *z, float *r, float speed); Represent four degrees of freedom x(mm),y(mm),z(mm),r(deg) of all the point coordinates in a section of trajectory with four float arrays, and indicate the total number of points and running speed, introduce into the trail_move function; Note: the linear distance between two adjacent points in the trajectory should be equal to 1mm	intpoint_number Number of points to be executed float *x The first address of x coordinate array, and the unit of data in the array is mm float *y The first address of y coordinate array, and the unit of data in the array is mm float *z The first address of z coordinate array, and the unit of data in the array is mm float *r The first address of r coordinate array, and the unit of data in the array is deg float speed Running speed	=0 the robotic arm is running other instructions, this command is invalid =1 this command goes into effect, and the robotic arm begins to move =2 setting speed is less than or equal to zero =3 not initialized yet =4 the first point in the trajectory goes beyond bounds =6 the robotic arm servo not opened =11 mobile terminal in controlling

续上表：

No.	Function Declaration	Incoming Parameter	Return Value
8	intset_angle_move(float angle1, float angle2, float z, float rotation, float speed);	float angle1 The absolute angle of the target point joint 1, unit is deg float angle2 The absolute angle of the target point joint 2, unit is deg float z The absolute coordinate of target point joint 3, unit is mm float rotation The absolute coordinate of target point joint 4, unit is deg float speed Running speed unit Judge the difference of the joint angles between the current position and the target point, divided by speed at the same time, to get the movement time of each joint, and take the longer time as the final movement time, and then inversely calculate the actual running speed of each joint	=0 the robotic arm is running other instructions, this command is invalid =1 this command goes into effect, and the robotic arm begins to move =2 setting speed is less than or equal to zero =3 not initialized yet =4 the position point goes beyond bounds =6 the robotic arm servo not opened =11 mobile terminal is controlling
9	intset_position_move(float goal_x, float goal_y, float goal_z, float rotation, float speed, float acceleration, int interpolation, intmove_mode); Move to the target point from the current position attitude	float goal_x X coordinate value of the target point, unit is mm float goal_y Y coordinate value of the target point, unit is mm float goal_z Z coordinate value of the target point, unit is mm float goal rotation z J4 angle value of the target point, unit is deg float speed running speed mm/s float acceleration acceleration value in T shape interpolation, valid only when interpolation=2; int interpolation 1 is s curve interpolation, and 2 is T curve interpolation intmove_mode =1 is MOVEJ The trajectory from the current position to the target position is a straight line (if it can arrive) =2 is MOVEL Each joint moves from the current position to the target position, and the intermediate movement trajectory is generally not a straight line	=0 the robotic arm is running other instructions, this command is invalid =1 this command goes into effect, and the robot arm begins to move =2 setting speed is less than or equal to zero =3 not initialized yet =4 in the MOVEL movement, the intermediate process points go out of bounds and it cannot arrive, and the robotic arm will stop moving =6 robotic arm servo not opened =7 in the MOVEL movement, any intermediate process point cannot arrive by the robotic arm's current attitude (attitude), and the robotic arm will stop moving =8 setting acceleration is less than or equal to zero =9 interpolation mode parameter error =10 move_mode move mode error =11 mobile terminal is controlling

Following :

No.	Function Declaration	Incoming Parameter	Return Value
10	intxyz_move(int direction, float distance, float speed) Motion of x, y, z single axis	int direction =1 x axis direction motion =2 y axis direction motion =3 z axis direction motion float distance Offset in the direction of direction relative to the current position float speed Unit is mm/s	=0 the robotic arm is running other instructions, this command is invalid =1 this command goes into effect, and the robotic arm begins to move =2 setting speed is less than or equal to zero =3 not initialized yet =4 process point cannot reach =5 direction parameter error =6 robotic arm servo not opened =7 any intermediate process point cannot arrive by the robotic arm's current attitude (attitude), and the robotic arm will stop moving =11 mobile terminal is controlling
11	booljudge_in_range(float x, float y, float z, float ratation) Judge whether the output position point can arrive	float x X axis coordinate value mm float y Y axis coordinate value mm float z Z axis coordinate value mm float rotation J4 joint angle deg	=0 it cannot arrive =1 it can arrive
12	void stop_move() Stop the robot arm and stop all movement	No Incoming Parameter	Cannot Return Value
13	void servo_off() Turn off the servo	No Incoming Parameter	Cannot Return Value
14	bool servo_on() Trun on the servo	No Incoming Parameter	=0 not initialized or initialization not completed =1 settings successful
15	bool set_digital_out(intio_number, bool value) Set io output	intio_number Output io port number, value range is 0-2 containing 0 and 2, which can be 0-2 currently; others are reserved bool value Set the output value of io_number =0 corresponds to the disconnect state of the two pins of io =1 corresponds to the conducting state of the two pins of io The connection of IO port output pin is shown in Appendix 1.2	=0 io_number parameter error =1 settings successful

Following：

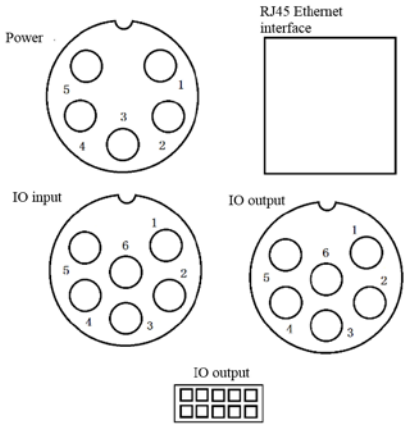
No.	Function Declaration	Incoming Parameter	Return Value
16	bool set_card_number(int n) Set up the robotic arm that needs to be controlled	int n The fourth bit of the robotic arm's IP address Value range is 0-255	=0 not connected to the specified robotic arm =1 settings successful
17	int card_number_is_connect (int n) Query whether the specified robotic arm is connected	int n The fourth bit of the robotic arm's IP address Value range is 0-255	=0 not connected =1 connected =2 parameter n error
18	int unlock_position(int n); Unlock function, before you control the movement of the robotic arm, you must unlock first	1) int n The fourth bit of the robotic arm's IP address	=0 not connected =1 connected =2 parameter n error
19	int get_digital_in(intio_in_number) Get the state value of the output IO	int io_in_number Input the io port number 0-2, including 0 and 2; Specific pin connection mode is shown in Appendix 1.3;	=0 24 v signal input =1 not connected, or no signal input =2 parameter io_in_number error
20	int set_efg_state(int type, float distance) Aims to control efg-20 motor-driven gripper(effective stroke is 20mm, which is unadjustable) and efg-8 motor-driven gripper(effective stroke is 8mm, which is unadjustable) Notice: every time after the mechanical arm is powered up, the controlling type can't be changed.	1） int type Type of motor-driven grippers: 20 for efg-20 and 8 for efg-8 2） float distance If type=20, distance for gripping position, data range (0,20), accurate to 0.1 If type=8, distance=0, stretch, Distance=1, clamp.	=1 Controls parameter changed =0 Type parameter error =1 Set ok
21	int get_efg_state(int *type, float *distance) Acquire the controlling type and the actual postion of the motor-driven gripper	1） int *type Shift to int pointer type. Assigning after function reference. type=0 controlling type unselected type=8 controlling type is efg-8 type=20 controlling type is efg-20 2) Shift to float pointer type. Assigning after function reference.	=1 Function reference ends

Following：

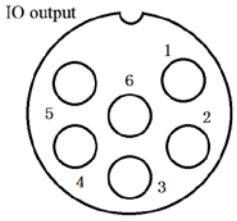
No.	Function Declaration	Incoming Parameter	Return Value
22	int get_digital_out(int io_out_num); Obtain the state of io output interface	1) int io_out_number the serial number of io interfaces.	=-1 io_out_num parameter error =0 output state of io interface is off =1 output state of io interface is on
23	void set_cooperation_fun_state(bool state) Set whether to turn on or turn off the coordination function (if model supports)	1) bool state =false off =true on	No returned value
24	bool get_cooperation_fun_state() Query whether the coordination function is on	No input parameter	=false off =true on
25	bool is_collision() Query whether the coordination function is triggered (collided)	No input parameter	=false no =true yes

Appendixes

1. Schematic diagram of interface panel:



2. Output IO port connection diagram:

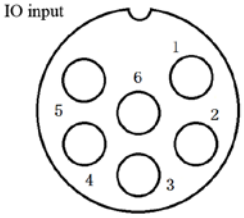


1-2 are connection ports of output IO0, 1 is connected to- high level, 2 is connected to - low level

3-4 are connection ports of output IO1, 3 is connected to - high level, 4 is connected to - low level

5-6 are connection ports of output IO2, 5 is connected to - high level, 6 is connected to - low level

3.Input IO port connection diagram:



1-2 are connection ports of output IO0, 1, 2 are connected to both ends of the signal line

3-4 are connection ports of output IO1, 3, 4are connected to both ends of the signal line

5-6 are connection ports of output IO2, 5, 6are connected to both ends of the signal line

Huiling-tech Robotic Co., Ltd

Tel 0755-36382405

Email hitbot@hitbot.cc

Website www.hitbot.cc

Address #503-505, G Building, Baoan Zhigu science and Technology Park, Yintian Road 4, Xixiang, Shenzhen, Guangdong, PR China