Z-Arm SDK instruction

I C#

Operation Environment: vs2010 runtime libraries, net framework 4.0 or above installed

Development Environment: vs2010 or above installed

1, Preparation

- Set a new C# program, copy the appropriate libraries to the relevant file, included ClassLibrary_ControlBean.dll, share.dll, server.exe, small_scara_interface.dll, and copy the 32bit application to bin\debug, the 64bit to bin\x64\Debug.
- Add ClassLibrary_ControlBean.dll to references with the program.
 Add using TcpserverExDll; using ControlBeanExDll to the main program.
- 3, First, call the TcpserverEx.net_port_initial() to initialize the network.
- Call TcpserverEx.card_number_connect(int card_number) to check the equipment connected or not.
- 5, Call ControlBeanEx robot= TcpserverEx.get_robot(int card_number) to select the Z-arm.
- 6, Call robot.initial(int generation,float z_travel) to initialize the Z-Arm.
- Call robot. set_arm_length(float 11,float 12) to set the first joint and second joint with Z-Arm, the default is 11=200,12=200;
- 8, After initialization, call robot.unlock_position() to unlock Z-Arm.
- 9, Z-Arm could be controlled with other control libraries when it was unlock.

2. Libraries instruction

1) TcpserverEx Type:

Member function:

No.	Function declaration	Incoming Parameter	Return Value
1	static void net_port_initial()		

	Explain: initialize the server		
2	static int card_number_connect(int card_number); Explain: Check the Z-Arm connected or	1)card_number (The number of Z-Arm, the fourth bit of the IP	=0 disconnected =1 connected =2 incoming parameter error
	not	address)	=101 incoming parameter NAN
3	static ControlBeanEx get_robot(int card_number); Explain: Select the Z-Arm with number	1)card_number (The number of Z-Arm, the fourth bit of the IP address)	Callback from the selected Z-Arm
4	static void close_tcpserver(); Explain: close the tcpserver		

2) ControlBeanEx Type

Member function:

float x;//Set the coordinate with X axes of Z-Arm (mm)

float y; // Set the coordinate with Y axes of Z-Arm (mm)

float z; // Set the coordinate with Z axes of Z-Arm (mm)

float angle1; // Set the coordinate witch first joint of Z-Arm(deg)

float angle2; // Set the coordinate with second joint of Z-Arm (deg)

float rotation;// // Set the coordinate with fourth joint (deg)

bool communicate_success;//state of host computer and Z-Arm, true (connected), false(disconnected)

bool initial_finish;//state of initialization, true(initialized), false(not initialized)

bool move_flag;//state of Z-Arm, true(running), false(stop)

bool servo_off_flag;//state of Z-Arm, true(servo on), false, (servo off)

-- The above variants should be update after calling get_scara_param()

bool isReach_after_judge;//

-- The above variants should call judge_position_gesture() and get callback with true before update.

float angle1_after_judge;// With judge_position_gesture() libraries, x and y are the relative coordinates

of first joint.

float angle2_after_judge; // With judge_position_gesture() libraries, x and y are the relative coordinates of second joint.

-- The above variants should call judge_position_gesture() and angle2_after_judge get callback with true before update.

int efg_type;

float efg_distance;

-- The above variants should call get_efg_state() and get callback with 1 before update.

Function libraries:

No.	Function declaration	Incoming Parameter	Return Value
		1) int generation.	
		=1 Z-Arm low	=0 communication has not yet
		configuration	been established, this initialization
	int initial(=2 Z-Arm high	is unsuccessful;
	int generation,	configuration	=1 initializing;
1	float z_travel)	2) float z_travel	=2 generation parameter error;
Т	Initial parameters corresponding to the model	Set the up and down	=3 encoder value error;
	Default J1 joint arm length is 200mm	stroke to be 210 or	=11 controlled by the mobile
	Default J2 joint arm length is 200mm.	310 according to the	terminal, this initialization is not
		actual model, and you	successful
		need to pass positive	=12 z_travel transmission error
		value;	
	void get_scara_param(float *x,float *y,float*z, float *angle1, float *angle2,	 float *x, coordinate value of x (mm) float *y, coordinate value of y (mm) float *z, coordinate 	
2	float *rotation, bool *communicate_success, bool *initial_finish, bool *servo_off_flag, bool *move_flag)	value of z (mm) 4) float *angle1, angle value of joint 1 (deg) 5) float *angle2, angle value of joint 2	No Return Value

		(deg)	
		6) float * rotation,	
		angle value of joint 4	
		(deg)	
		7) bool	
		*communicate_succe	
		ss,	
		=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 alosad	
		=0 serve alocad	
		=1 servo ciosed	
		10) bool *move_flag,	
		=0 the robot arm is in	
		standby state	
		=1 the robot arm is in	
		motion state	
		float 11	
		J1 joint length,	
		reserved parameters,	
		you must introduce J1	
	void set_arm_length(joint length, introduce	
3	float 11,	160 or 200 generally	No Return Value
-	float 12)	for those with J4	
	Set J1 J2 joint arm length	rotation joint, and set	
		according to the	
		actual situation for	
		those with no J4	
		rotation joint	
	int unlock_position(1) int n	
А	int n);	The fourth bit of the	=U not connected
4	Unlock function, before you control the movement of the	robotic arm's IP	=1 connected
	robotic arm, you must unlock first	address	=2 parameter n error
5	bool is_connected();		=true connected

	Explain: check the Z-Arm connected or not		=false disconnected
6	int get_card_num(); Explain: Get the number of Z-Arm, and get invoked after initialization		Callback the number of Z-Arm
7	int get_joint_state(int joint_num); Explain: Get state of Z-Arm, available after initialization	1)joint_num Joint number	 =0 reset the joint and need to be initialized again =1 joint could move regular =2 incoming parameter error =3 joint is not initialized =4 fail to get joint state (Only new version support) =5 in collision; =6 Drag-teaching mode;
8	bool set_drag_teach(bool state); Explain: Only Z-Arm support, drag-teaching mode on, the other joint could be drag except joint 3, available after initialization	1)state True on False off	=true setting success =false setting fail
9	bool get_drag_teach(); Explain: Only Z-Arm support, to check drag-teaching mode is on or off, available after initialization		=false drag-teaching mode off =true drag-teaching mode on
10 11 11 12	bool set_cooperation_fun_state(bool state); Explain: Only Z-Arm support, to check collision protect mode is on or off, available after initialization bool get_cooperation_fun_state() Query whether the coordination function is on bool is_collision() Query whether the coordination function is triggered	1)state true collision protect mode on false collision protect mode off No input parameter No input parameter	=true setting success =false setting fail =false off =true on =false no =true yes

Movement libraries

No.	Function declaration	Incoming Parameter	Return Value
		float goal_x	=0 the robotic arm is
		X coordinate value of the target point, unit is	running other instructions,
		mm	this command is invalid
		float goal_y	=1 this command goes into
		Y coordinate value of the target point, unit is	effect, and the robot arm
		mm	begins to move
		float goal_z	=2 setting speed is less than
		Z coordinate value of the target point, unit is	or equal to zero
	int set_position_move(mm	=3 not initialized yet
	float goal_x,	float goal rotation z	=4 in the MOVEL
	float goal_y,	J4 angle value of the target point, unit is deg	movement, the intermediate
	float goal_z, float rotation, float speed, float acceleration, int interpolation, intmove_mode); Move to the target point from the current position attitude	float speed	process points go out of
		running speed mm/s	bounds and it cannot arrive,
4		float acceleration	and the robotic arm will
		acceleration value in T shape interpolation,	stop moving
		valid only when interpolation=2;	=6 robotic arm servo not
		int interpolation	opened
		1 is s curve interpolation, and 2 is T curve	=7 in the MOVEL
		interpolation	movement, any
		intmove_mode	intermediate process point
		=1 is MOVEJ	cannot arrive by the robotic
		The trajectory from the current position to the	arm's current attitude
		target position is a straight line (if it can	(attitude), and the robotic
		arrive)	arm will stop moving
		=2 is MOVEL	=8 setting acceleration is
		Each joint moves from the current position to	less than or equal to zero
		the target position, and the intermediate	=9 interpolation mode

		movement trajectory is generally not a	parameter error
		straight line	=10 move_mode move
			mode error
			=11 mobile terminal is
			controlling
			=101 Incoming parameter
			NAN
			102 In collicion, could not
			-102 In comsion, could not
			move
			=103 joint was reset and
			need to be initialized again
2	int set_angle_move(float angle1, float angle2, float z, float rotation, float speed);	float angle 1 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 =101 Incoming parameter NAN =102 In collision, could not move =103 joint was reset and
		int direction	=0 the robotic arm is
	int xyz_move(=1 x axis direction motion	running other instructions.
	int direction,	=2 v axis direction motion	this command is invalid
3	float distance,	=3 z axis direction motion	=1 this command goes into
	float speed)	float distance	affact and the relatio
	Motion of x, y, z single axis		the robotic afm
		Offset in the direction of direction relative to	begins to move

		the current position	=2 setting speed is less than
		float speed	or equal to zero
		Unit is mm/s	=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
			=101 Incoming parameter
			NAN
			=102 In collision, could not
			move
			=103 joint was reset and
			need to be initialized again
		1)axis	=0 Z-Arm is in state of
	int single joint move(Input 1 to 4, matching with joint 1 to joint 4	other implementation, the
4	int suic	2)distance	present implementation is
	float dictance	Moving distance from the present position,	unavailable.
	float speed):	When axis=3, unit of distance is mm, when	=1 present implementation
	noa specu),	axis=1 or 2 or 3, unit of distance is deg	is available, Z-Arm begin
		3)speed	moving

		Moving speed,	=2 set the speed less or
		When axis=3, unit of speed is mm/s	equal to 0
		When axis=1 or 2 or 4, unit of speed is deg/s	=3 Not initialized yet
			=4 Could not reach the
			position
			=5 parameter error of the
			number of output axis
			=6 Z-Arm servo off
			=11 Controlling by APP
			=101 Incoming parameter
			NAN
			=102 In collision, could not
			move
			=103 joint was reset and
			need to be initialized again
	int trail_move(intpoint_number	=0 the robotic arm is
	intpoint_number,	Number of points to be executed	running other instructions,
	float *x,	float *x	this command is invalid
	float *y,	The first address of x coordinate array, and	=1 this command goes into
	float *z,	the unit of data in the array is mm	effect, and the robotic arm
	float *r,	float *y	begins to move
5	float speed);	The first address of y coordinate array, and	=2 setting speed is less than
•	Represent four degrees of freedom	the unit of data in the array is mm	or equal to zero
	x(mm),y(mm),z(mm),r(deg) of all the point	float *z	=3 not initialized yet
	coordinates in a section of trajectory with	The first address of z coordinate array, and	=4 the first point in the
	four float arrays, and indicate the total	the unit of data in the array is mm	trajectory goes beyond
	number of points and running speed,	float *r	bounds
	introduce into the trail_move function;	The first address of r coordinate array, and	=6 the robotic arm servo not
	Note: the linear distance between two	the unit of data in the array is deg	opened

	adjacent points in the trajectory should be	float speed	=11 mobile terminal in
	equal to 1mm	Running speed	controlling
			=101 Incoming parameter
			NAN
			=102 In collision, could not
			move
			=103 joint was reset and
			need to be initialized again
			=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
		flast speed	=2 the incoming speed is
		The isist grand (dec()) when transforming	less than or equal to 0
	int change officials(the joint speed (deg/s) when transforming	=3 not initialized yet
ć		attitude, and the difference of the joint angles	=4 can't reach by the other
Ð	The speed	between the two attitudes will be judged. At	attitude
		the same time, divided by speed to get the	=6 servo not opened
		is the final manufacture time	=11 mobile terminal is
		is the final movement time	controlling
			=101 Incoming parameter
			NAN
			=102 In collision, could not
			move
			=103 joint was reset and
			need to be initialized again
7	void stop_move()	No Incoming Parameter	Cannot Return Value

	Stop the robot arm and stop all movement		
8	void servo_off() Turn off the servo	No Incoming Parameter	Cannot Return Value
9	bool servo_on() Trun on the servo	No Incoming Parameter	=0 not initialized or initialization not completed =1 settings successful
10	bool is_robot_goto_target(); Explain: Check the four incoming parameters whether the arm arrived the position or not		=true arrived =false not arrived
11	void set_allow_offset_at_target_position(float x_distance, float y_distance, float z_distance, float r_distance);	 1)x_distance X axes coordinate deviation 2)y_distance Y axes coordinate deviation 3)z_distance Y axes coordinate deviation 4)r_distance R axes coordinate deviation 	
12	void set_catch_or_release_accuracy(float accuracy); Explain: The allowed error to get the position when the Z-Arm moving with Y axes.	1)accuracy allowed error	
13	bool judge_in_range(float x, float y, float z, float ratation) Judge whether the output position point	float x X axis coordinate value mm float y Y axis coordinate value mm float z Z axis coordinate value mm	=false it cannot arrive =true it can arrive

	can arrive	float rotation	
		J4 joint angle deg	
	bool judge_position_gesture(
	float x, float y);		
	Explain: Callback bool	1)x	
	judge_position_gesture(float x, float y)	X axes coordinate of position	=ture success
14	before callback set_position_move(), if it	2)y	=false fail, Z-Arm is
14	could callback true that means available,	Y axes coordinate of position	running.
	check member variants		
	isReach_after_judge, if callback true,		
	position could be reached, if not, it could		
	not arrived.		
	inticipt home		=0 connected
	int joint_nome(=1 success
15	int joint_num);		=2 incoming parameter
	Explain: Reset the Z-Arm when it was	1)joint_num	error
	connected but haven't been initialized.	Joint number	
	After callback, Z-Arm will be back to not		=3 Z-Arm is initializing.
	initialized.		

IO libraries

No.	Function declaration	Incoming Parameter	Return Value
		intio_number	
	bool set_digital_out(Function declaration Incoming Parameter intio_number intio_number et_digital_out(Output io port number, value range number, is 0-2 containing 0 and 2, which can alue) be 0-2 currently; others are reserved output bool value Set the output value of io_number	-0 io number parameter error
1	intio_number,	is 0-2 containing 0 and 2, which can	
+	bool value)	be 0-2 currently; others are reserved	=1 settings successful
	Set io output	bool value	=3 not initialized yet
		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	
			=-1 io_out_num parameter
		 =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 1) int io_out_number the serial number of io interfaces. 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; 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 	error
	int and divided and in and annaly		=0 output state of io interface
-2	Int get_algital_out(int io_out_num);	1) int io_out_number	is off
	Obtain the state of 10 output interface	the serial number of to interfaces.	=1 output state of io interface
			is on
			=3 not initialized yet
		int io_in_number	=0 24 v signal input
	ist act divital is (Input the io port number 0-2,	=1 not connected, or no signal
2	intiget_orginal_in(=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 1) int io_out_number the serial number of io interfaces. 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; 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, 	input
÷	Cot the state value of the output IO	Specific pin connection mode is	=2 parameter io_in_number
	Get the state value of the output 10	shown in Appendix 1.3;	error
			=3 not initialized yet
		1) int type	
	int set_efg_state(int type, float distance)	Type of motor-driven	
	Aims to control efg-20 motor-driven	grippers: 20 for efg-20 and 8	
	gripper(effective stroke is 20mm, which is	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 1) int io_out_number the serial number of io interfaces. 1) 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; 1) int type Type of motor-driven grippers: 20 for efg-20 and 8 is for efg-8 ven is 2.) float distance If type=20, distance for gripping position, data range ed. (0,20), accurate to 0.1 If type=8, distance=0, stretch,	=1 Controls parameter
-2 int get_digital_out(int io_out_num); 1) -2 int get_digital_out(int io_out_num); 1) Obtain the state of io output interface th int get_digital_in(in intic_in_number) Get the state value of the output IO Si Get the state value of the output IO sh 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 2: unadjustable) 1: unadjustable) Notice: every time after the mechanical arm is powered up, the controlling type can't be changed. 2: unadjustable)		changed	
4	gripper(effective stroke is 8mm, which is	2) float distance	=0 Type parameter error
	unadjustable)	If type=20, distance for	=1 Set ok
	Notice: every time after the mechanical arm is	gripping position, data range	=3 not initialized yet
	powered up, the controlling type can't be changed.	(0,20), accurate to 0.1	
		If type=8, distance=0, stretch,	



II CPP version.

For now, only support the Windows x86 or x 64 application developed by CPP.

1, Preparation

1 New CPP program, copy hitbot_interface.h, ControlBeanEx.h, hitbot_interface.lib to the source program files, and INCLUDE the first 2 files with the program.

- 2 Copy the libraries of relative version to the debug files, included hitbot_interface.dll, share.dll, server.exe, small_scara_interface.dll.
- 3 Callback net_port_initial() to initialized the network.
- 4 Callback card_number_connect(int card_number) to check whether the equipment connected or not.
- 5 Callback ControlBeanEx* robot=get_robot(int card_number) to select the Z-Arm.
- 6 Callback robot->initial(int generation,float z_travel) to initialized the Z-Arm.

- 7 Callback robot->set_arm_length(float 11,float 12) to set joint 1 and joint 2 withZ-Arm, the default is 11=200,12=200.
- 8 After initialization, callback robot->unlock_position() to unlock the Z-Arm
- 9 Z-Arm could be controlled with other libraries after unlock.

2, Libraries instruction

1) Export libraries of hitbot_interface with dynamic-link library:

Function	libraries:

No.	Function declaration	Incoming Parameter	Return Value
	extern "C"declspec(dllexport)		
1	void net_port_initial();		
	Explain: initialize the server		
	extern "C"declspec(dllexport) int	Deard number (The number of	=0 disconnected
	card_number_connect(int num);	r)card_number (The number of	=1 connected
2	Explain: Check the Z-Arm connected or	Z-Arm, the fourth bit of the IP	=2 incoming parameter error
	not	adressy	=101 incoming parameter NAN
	extern "C"declspec(dllexport)	1)card_number (The number of	
3	ControlBeanEx * get_robot(int	Z-Arm, the fourth bit of the IP	Callback to the 7 -Arm number
5	card_number);	address)	
	Explain: get the Z-Arm pointer		
	extern "C"declspec(dllexport)		
4	<pre>void close_tcpserver();</pre>		
	Explain: close the tcpserver		

2) ControlBeanEx Type

Member function:

float x;//Set the coordinate with X axes of Z-Arm (mm)

float y; // Set the coordinate with Y axes of Z-Arm (mm)

float z; // Set the coordinate with Z axes of Z-Arm (mm)

float angle1; // Set the coordinate witeh first joint of Z-Arm(deg)

float angle2; // Set the coordinate with second joint of Z-Arm (deg)

float rotation;// // Set the coordinate with fourth joint (deg)

bool communicate_success;//state of host computer and Z-Arm, true (connected), false(disconnected)

bool initial_finish;//state of initialization, true(initialized), false(not initialized)

bool move_flag;//state of Z-Arm, true(running), false(stop)

bool servo_off_flag;//state of Z-Arm, true(servo on), false, (servo off)

-- The above variants should be update after calling get_scara_param()

bool isReach_after_judge;//

-- The above variants should call judge_position_gesture() and get callback with true before update.

float angle1_after_judge;// With judge_position_gesture() libraries, x and y are the relative coordinates of first joint.

float angle2_after_judge; // With judge_position_gesture() libraries, x and y are the relative coordinates of second joint.

-- The above variants should call judge_position_gesture() and angle2_after_judge get callback with true before update.

int efg_type;

float efg_distance;

-- The above variants should call get_efg_state() and get callback with 1 before update.

No.	Function declaration	Incoming Parameter	Return Value
	int initial(1) int generation.	=0 communication has not yet
	int generation,	=1 Z-Arm low	been established, this initialization
4	float z_travel)	configuration	is unsuccessful;
	Initial parameters corresponding to the model	=2 Z-Arm high	=1 initializing;
	Default J1 joint arm length is 200mm	configuration	=2 generation parameter error;

Functional member libraries:

	Default J2 joint arm length is 200mm.	2) float z_travel	=3 encoder value error;
		Set the up and down	=11 controlled by the mobile
		stroke to be 210 or	terminal, this initialization is not
		310 according to the	successful
		actual model, and you	=12 z_travel transmission error
		need to pass positive	
		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)	 float *x, coordinate value of x (mm) float *y, coordinate value of y (mm) float *z, coordinate value of z (mm) float *angle1, angle value of joint 1 (deg) float *angle2, angle value of joint 2 (deg) float * rotation, angle value of joint 4 (deg) float * rotation, angle value of joint 4 (deg) bool *communicate_succe ss, communication has not been connected communication has been established bool *initial_finish, o initialization successful initialization unsuccessful servo_off_flag, o servo not closed bool *move_flag, the robot arm is in standby state the robot arm is in motion state 	No Return Value
	void set_arm_length(float 11	
3	float 11,	J1 joint length,	No Return Value
	float 12)	reserved parameters,	

	Set J1 J2 joint arm length	you must introduce	
		200,	
		float 12	
		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	
4	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
	bool is_connected();		=true connected
5	Explain: check the Z-Arm connected or not		=false disconnected
	int get_card_num();		
6	Explain: Get the number of Z-Arm, and get invoked after		Callback the number of Z-Ar
	initialization		
			=0 reset the joint and need to be
			initialized again
			=1 joint could move regular
	int get joint state(int joint num).		=2 incoming parameter error
7	Evaluity Cot state of 7 Arm available after initialization	1)joint_num	=3 joint is not initialized
	Explain. Get state of Z-Arm, available arter initianzation	Joint number	=4 fail to get joint state (Only new
			version support)
			=5 in collision;
			=6 Drag-teaching mode;
0	<pre>bool set_drag_teach(bool state);</pre>	1)state	=true setting success
0	Explain: Only Z-Arm support, drag-teaching mode on, the	True on	=false setting fail

	other joint could be drag except joint 3, available after	False off	
	initialization		
	<pre>bool get_drag_teach();</pre>		-false drag teaching mode off
9	Explain: Only Z-Arm support, to check drag-teaching mode		
	is on or off, available after initialization		=true drag-teaching mode on
		1)state	
	<pre>bool set_cooperation_fun_state(bool state);</pre>	true collision	
10	Explain: Only Z-Arm support, to check collision protect	protect mode on	=true setting success
	mode is on or off, available after initialization	false collision	=false setting fail
		protect mode off	
11	bool get_cooperation_fun_state()	No input parameter	=false off
11	Query whether the coordination function is on	No input parameter	=true on
	bool is_collision()		=false no
12	Query whether the coordination function is triggered	No input parameter	=true yes
	(collided)		-

Movement libraries

No.	Function declaration	Incoming Parameter	Return Value
	int set position move(float goal_x	=0 the robotic arm is
	float goal x	X coordinate value of the target point, unit is	running other instructions,
	float goal x	mm	this command is invalid
	float goal z	float goal_y	=1 this command goes into
	flast station	Y coordinate value of the target point, unit is	effect, and the robot arm
		mm	begins to move
4	float speed,	float goal_z	=2 setting speed is less than
	noat acceleration,	Z coordinate value of the target point, unit is	or equal to zero
	int interpolation,	mm	=3 not initialized yet
	intmove_mode);	float goal rotation z	=4 in the MOVEL
		J4 angle value of the target point, unit is deg	movement, the intermediate
	Move to the target point from the current	float speed	process points go out of
	position attitude	running speed mm/s	bounds and it cannot arrive,

		float acceleration	and the robotic arm will
		acceleration value in T shape interpolation,	stop moving
		valid only when interpolation=2;	=6 robotic arm servo not
		int interpolation	opened
		1 is source interpolation and 2 is Tource	-7 in the MOVEL
		T is s curve interpolation, and 2 is 1 curve	
		interpolation	movement, any
		intmove_mode	intermediate process point
		=1 is MOVEJ	cannot arrive by the robotic
		The trajectory from the current position to the	arm's current attitude
		target position is a straight line (if it can	(attitude), and the robotic
		arrive)	arm will stop moving
		=2 is MOVEL	=8 setting acceleration is
		Each joint moves from the current position to	less than or equal to zero
		the target position, and the intermediate	=9 interpolation mode
		movement trajectory is generally not a	parameter error
		straight line	=10 move_mode move
			mode error
			=11 mobile terminal is
			controlling
		float angle1	
		The absolute angle of the target point joint 1, unit is deg	
		float angle2	running other instructions,
		The absolute angle of the target point joint 2, unit is deg	this command is invalid
	int set_angle_move(float z	=1 this command goes into
	float angle1,	The absolute coordinate of target point joint	begins to move
	a	float rotation	=2 setting speed is less than
2	noat anglez,	The absolute coordinate of target point joint	or equal to zero =3 not initialized yet
	float z,	4, unit is deg	=4 the position point goes
	float speed);	Running speed unit	beyond bounds
	• **	Judge the difference of the joint angles	=6 the robotic arm servo not
		between the current position and the target	=11 mobile terminal is
		point, divided by speed at the same time, to	controlling
		the longer time as the final movement time,	
		and then inversely calculate the actual	

		running speed of each joint	
3	int xyz_move(int direction, float distance, float speed) Motion of x, y, z single axis	running speed of each joint int direction =1 x axis direction motion =2 y 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
4	int single_joint_move(int axis, float distance, float speed);	 1)axis Input 1 to 4, matching with joint 1 to joint 4 2)distance Moving distance from the present position, When axis=3, unit of distance is mm, when axis=1 or 2 or 3, unit of distance is deg 3)speed 	 =0 Z-Arm is in state of other implementation, the present implementation is unavailable. =1 present implementation is available, Z-Arm begin moving

		Moving speed,	=2 set the speed less or
		When axis=3, unit of speed is mm/s	equal to 0
		When axis=1 or 2 or 4, unit of speed is deg/s	=3 Not initialized yet
			=4 Could not reach the
			position
			=5 parameter error of the
			number of output axis
			=6 Z-Arm servo off
			=11 Controlling by APP
			=101 Incoming parameter
			NAN
			=102 In collision, could not
			move
			=103 joint was reset and
			need to be initialized again
	int trail_move(intpoint_number	=0 the robotic arm is
	intpoint_number,	Number of points to be executed	running other instructions,
	float *x,	When axis=3, unit of speed is mm/sequWhen axis=1 or 2 or 4, unit of speed is deg/s=3Harmonic action of trajectory withIndexIndicate the totalIndexIndicate the totalIndexIndicate the totalIndexIndicate the totalIndexIndicate the totalIndexIndicate the totalIndexIndicate the totalIndicate array is mmIndicate array is mmIndicate the totalIndicate array is degIndicate array is degIndicate the totalIndicate array is degIndicate array is deg	this command is invalid
	float *y,		=1 this command goes into
	float *z,	the unit of data in the array is mm	effect, and the robotic arm
	float *r,	float *y	begins to move
5	float speed);	The first address of y coordinate array, and	=2 setting speed is less than
5	Represent four degrees of freedom	the unit of data in the array is mm	or equal to zero
	x(mm),y(mm),z(mm),r(deg) of all the point	float *z	=3 not initialized yet
	coordinates in a section of trajectory with	The first address of z coordinate array, and	=4 the first point in the
	four float arrays, and indicate the total	the unit of data in the array is mm	trajectory goes beyond
	number of points and running speed,	float *r	bounds
	introduce into the trail_move function;	The first address of r coordinate array, and	=6 the robotic arm servo not
	Note: the linear distance between two	the unit of data in the array is deg	opened

	adjacent points in the trajectory should be	float speed	=11 mobile terminal in
	equal to 1mm	Running speed	controlling
6	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
7	void stop_move() Stop the robot arm and stop all movement	No Incoming Parameter	controlling Cannot Return Value
8	void servo_off() Turn off the servo	No Incoming Parameter	Cannot Return Value
9	bool servo_on() Trun on the servo	No Incoming Parameter	=0 not initialized or initialization not completed =1 settings successful
10	bool is_robot_goto_target(); Explain: Check the four incoming parameters whether the arm arrived the position or not void	1)x_distance	=true arrived =false not arrived
11	set_allow_offset_at_target_position(float	X axes coordinate deviation	

	x_distance, float y_distance, float	2)y_distance	
	z_distance, float r_distance);	Y axes coordinate deviation	
		3)z_distance	
		Y axes coordinate deviation	
		4)r_distance	
		R axes coordinate deviation	
	void set_catch_or_release_accuracy(
	float accuracy);		
12	Explain: The allowed error to get the	1)accuracy allowed error	
	position when the Z-Arm moving with Y		
	axes.		
	had index in range	float x	
		X axis coordinate value mm	
	noat x,	float y	
	float y,	Y axis coordinate value mm	=0 it cannot arrive
13	float z,	float z	=1 it can arrive
	float ratation)	Z axis coordinate value mm	
	Judge whether the output position point	float rotation	
	can arrive	J4 joint angle deg	
	bool judge_position_gesture(
	float x, float y);		
14	Explain: Callback bool	1)x	
	judge_position_gesture(float x, float y)	X axes coordinate of position	=ture success
	before callback set_position_move(), if it	2)y	=false fail, Z-Arm is
	could callback true that means available,	Y axes coordinate of position	running.
	check member variants		
	isReach_after_judge, if callback true,		
	position could be reached, if not, it could		
	not arrived.		

			=0 connected
	int joint_home(=1 success
	int joint_num);		
			=2 incoming parameter
15	Explain: Reset the Z-Arm when it was	1)joint_num	
15	connected but haven't been initialized.	Joint number	error
			=3 Z-Arm is initializing.
	After callback, Z-Arm will be back to not		
	initialized.		

IO libraries

No.	Function declaration	Incoming Parameter	Return Value
		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 set_digital_out(bool value	
1	intio_number,	Set the output value of io_number	=0 io_number parameter error
+	bool value)	=0 corresponds to the disconnect	=1 settings successful
	Set io output	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	
			=-1 io_out_num parameter
			error
2	int get_digital_out(int io_out_num);	1) int io_out_number	=0 output state of io interface
	Obtain the state of io output interface	the serial number of io interfaces.	is off
			=1 output state of io interface
			is on

3	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
4	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.	 3) int type Type of motor-driven grippers: 20 for efg-20 and 8 for efg-8 4) 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
5	int get_efg_state (int* type, float* distance); Explain: Get the electric gripper model and position, after callback, value of type,distance will be assigned. Type=0 means gripper could not be identified, type=8 means EFG-8 electric gripper has been identified, type=20 means EFG-20 electric gripper has been identified.	1)type Int pointer 2)distance Float pointer	=1 Callback success =3 Not initialized

III IO interface instruction

Appendice

1. Interface instruction (old version):



2. IO output interface instruction (Old version):



1-2 are IOO output interfaces, 1 for high level, 2 for low level.
3-4 are IOO output interfaces, 3 for high level, 4 for low level.
5-6 are IOO output interfaces, 5 for high level, 6 for low level.
3. IO input interface instruction (Old version):



1-2 are IOO input interfaces, 1 and 2 connecting with signal cable at both ends.3-4 are IOO input interfaces, 3 and 4 connecting with signal cable at both ends.5-6 are IOO input interfaces, 5 and 6 connecting with signal cable at both ends.

4. Base interface instruction (New Version):



5. Joint 2 interface instruction (new version):

EFG电抓持	
5 IN1 4 OUT-COM 3 OUT2 2 OUT1 1 24V	 9 GND 8 GND 7 IN-COM 6 IN2