

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	1 / 37

Robotphoenix Robot-PCSDK (C++) Development and User Manual

ROBOTPHOENIX

济南翼菲自动化科技有限公司

2023-10-08

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	1 / 37

Document Description

The text and charts involved in this document are for internal use only by Robotphoenix and the party submitted by Robotphoenix. Please do not spread to any third party without the written permission of Robotphoenix.

Project name		Robot-PCSDK(C++)			
Document name		Robotphoenix Robot-PCSDK (C++) Development and User Manual			
Drafter		Zhang Qiang	Drafting date	2023.10.08	
Reviewer					
Approver					
Document Change Record					
Version	Chapter	Modification Type	Date	Reviser	Remarks
V1.0		First edition	2023.10.08	Zhang Qiang	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	2 / 37

Contents

1. Development environment and robot controller configuration错误！未定义书签。

2. API function description 9

3. Data definition usage32

3.1 Data classes definition32

3.2 Function return value faults and solutions 35

3.3 Data units36

1. Development environment and robot controller configuration

Microsoft Visual Studio 2015, C++ language development, supports adding library file references to call functions for use. The development project adds references to three library files libprotobuf.lib, libprotoc.lib and RobotPcSdk.lib. These library files are static libraries and do not require placing dll files under the running program. Add the following configuration to the project file under QT:

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	3 / 37

```
CONFIG(debug, debug|release) {
```

```
LIBS += -L$$PWD/PcSdk/Protobuf-3.6.1/Debug/ -llibprotobufd
```

```
LIBS += -L$$PWD/PcSdk/Protobuf-3.6.1/Debug/ -llibprotocd
```

```
LIBS += -L$$PWD/PcSdk/ -lRobotPcSdkd
```

```
}
```

```
CONFIG(release, debug|release) {
```

```
LIBS += -L$$PWD/PcSdk/Protobuf-3.6.1/Release/ -llibprotobuf
```

```
LIBS += -L$$PWD/PcSdk/Protobuf-3.6.1/Release/ -llibprotoc
```

```
LIBS += -L$$PWD/PcSdk/ -lRobotPcSdk
```

```
}
```

Add the SDK header file in the call control code file #include "PcSdk/robotpcsdk.h", all API functions are static functions and can be called directly using classes, such as: YIROBOTAPI::GetVel(nVel, strErrInfo), for detailed usage methods, please refer to the demo routine.

Supported models: Delta models, SCARA models, Diamond models.

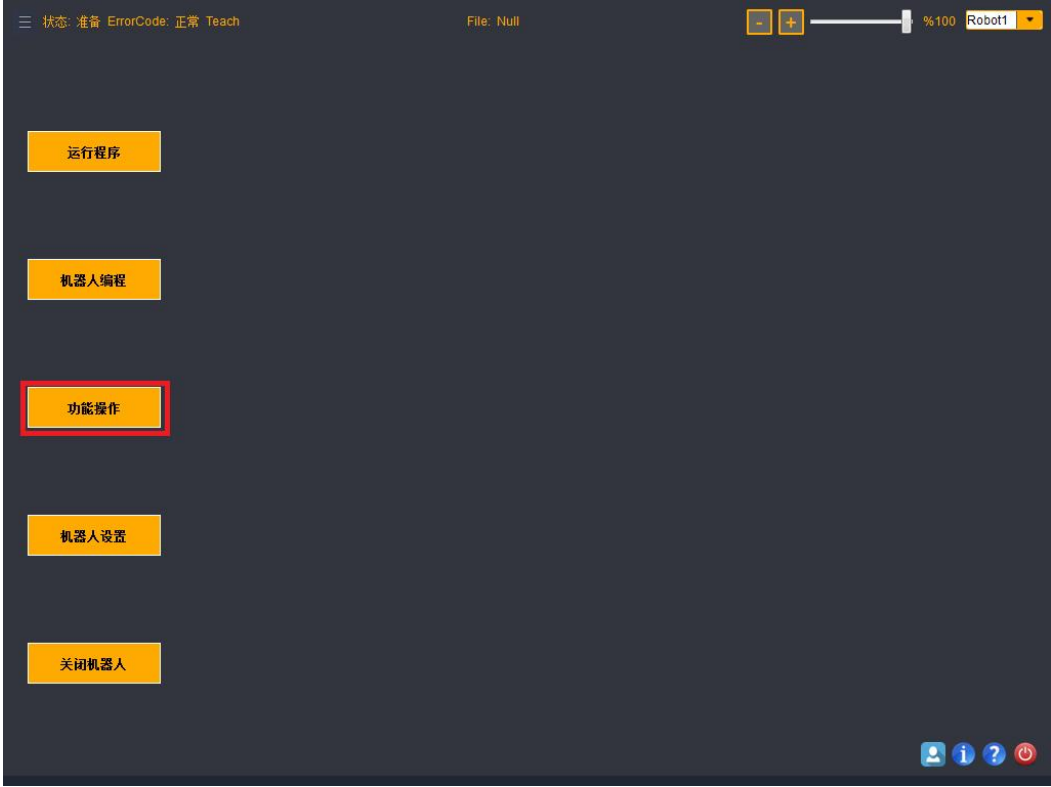
Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	4 / 37

Support the control of 1 robot, but do not support switching and selecting multiple robots for control.

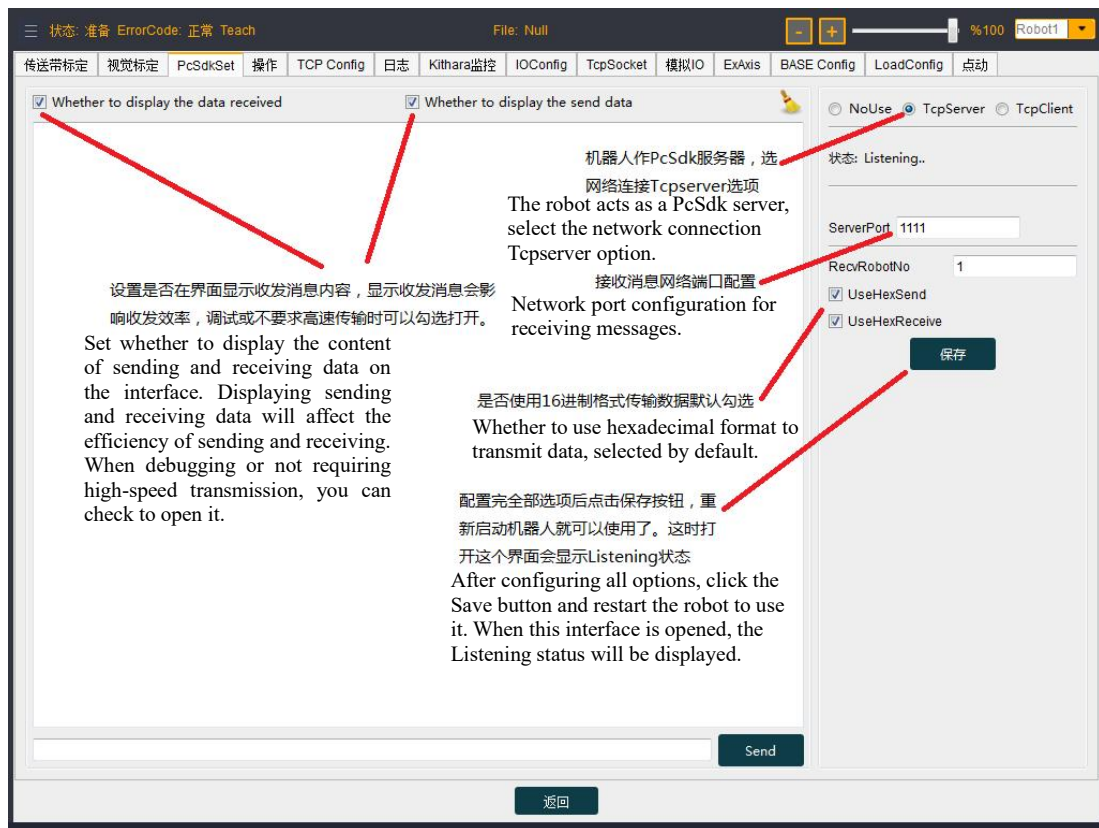
PcSdk server end configuration for robot controller end:

Open the robot control end and select the function operation to enter the settings interface. Select the PcSdkSet page for configuration.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	5 / 37

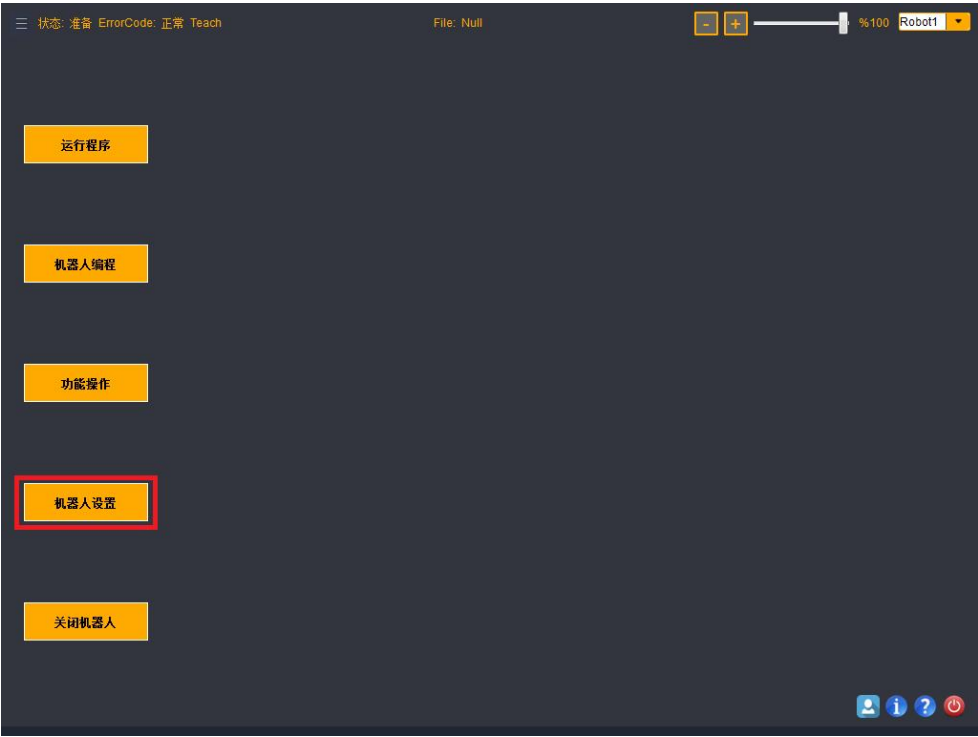


Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	6 / 37

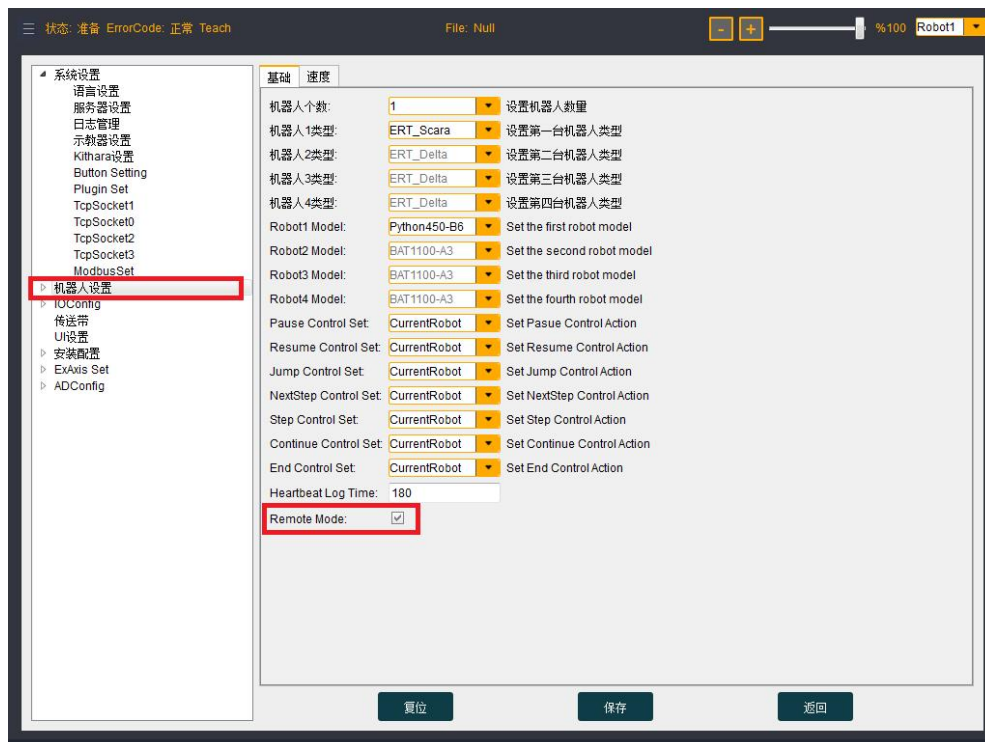


Robot configuration uses remote control mode, select robot settings option:

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	7 / 37



Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	8 / 37



Select the remote control mode in the settings interface, close the control software after setting it up, and then open it again. The robot control mode is initially in local mode. At this time, you cannot control the robot movement or set to obtain parameters. You can only switch modes through the SDK, and can choose to switch to remote teaching, remote debugging, or remote running mode on the SDK end.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	9 / 37

2. API function description

S. N.	Function name	Description	Parameters	Return value	Remarks
1	<code>int Connect(string ipAddr, int Port, int timeout=2)</code>	Establish robot network connection	ipAddr : robot IP address Port : the port number timeout : connection timeout time, in seconds, the default value is 2 seconds.	Returning 0 means successful connection, less than 0 means failed connection.	
2	<code>int Disconnect()</code>	Close robot network connection		Returning 0 means success, and less than 0 means failure.	
3	<code>bool IsConnect()</code>	Get the current connection status with the robot		Returning true means that it is connected, while false means that it is not connected.	
4	<code>int GetDO(int num, int & status, string & cmdErrInfo)</code>	Get the output DO	num: output DO number (0-39) status: output DO value (0-OFF, 1-ON) cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	0-31 are for general IO, 32-39 are used for robot configuration.
5	<code>int SetDO(int num, int status, string & cmdErrInfo)</code>	Set the output status of DO	num: output DO number (0-39) Status: output DO value (0-OFF, 1-ON) cmdErrInfo: the error	Returning 0 means successful setting, less than	0-31 are for general IO, 32-39 are used for robot configuration.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	10 / 37

			information of command execution. It is valid when the return value is less than zero.	0 means failed setting.	
6	<pre>int SetOutputPulse(int num, int status, double startTime, double durationTime, string & cmdErrInfo)</pre>	Set the pulse output	num: output DO number (0-39) status : output DO value (0-OFF, 1-ON) startTime : start time, in seconds. durationTime:duration time, in seconds. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful setting, less than 0 means failed setting.	0-31 are for general IO, 32-39 are used for robot configuration.
7	<pre>int SetMovIOPulse(int num, int status, int kind, double value, double durationTime, string & cmdErrInfo)</pre>	Set the pulse output during movement	num: output DO number (0-39) status : output DO value (0-OFF, 1-ON) kind: set the type of IO during movement (0-time type, 1-path percentage, 2-distance), currently only time type is supported. value: when the type is time, it means that the signal is output value seconds before reaching the movement point. When the type is path percentage, it means that the signal is output when value*100% of the entire	Returning 0 means successful setting, less than 0 means failed setting.	0-31 are for general IO, 32-39 are used for robot configuration. Kind currently only supports 0 and 1, and this function has not yet been implemented.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	11 / 37

			path is completed from the start of movement. When the type is distance, greater than 0 means that the signal is output after the movement starts from the starting point to Value millimeters, and less than or equal to 0 means that the signal is output before the movement reaches Value millimeters from the end point. durationTime: duration time, in seconds. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
8	int GetDI(int num, int & status, string & cmdErrInfo)	Query the input status of DI by bit.	num: input DI number (0-39) Status: input DI value (0-OFF, 1-ON) cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	0-31 are for general IO, 32-39 are used for robot configuration.
9	int GetDOGroup(int num, bool vecstatus[], string & cmdErrInfo)	Get output DO by group	num: output DO group number (0-4) vecstatus: DO status in each group, the length of the bool array is 8. Among them,	Returning 0 means successful query, and less than 0 means failed query.	0-3 are for general IO, 4 is used for robot configuration.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	12 / 37

			vecstatus[0]-vecstatus[7] respectively correspond to the D0 status from the smallest to the largest serial number in each group. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
10	int SetDOGroup(int num, bool vecstatus[], string & cmdErrInfo)	Set the output status of DO by group	num: input DO group number (0-4) vecstatus: D0 status in each group, the length of the bool array is 8. Among them, vecstatus[0]-vecstatus[7] respectively correspond to the D0 status from the smallest to the largest serial number in each group. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful setting, less than 0 means failed setting.	0-3 are for general IO, 4 is used for robot configuration.
11	int GetDIGroup(int num, bool vecstatus[], string & cmdErrInfo)	Query the input status of DI by group	num: input DI group number (0-4) vecstatus: DI status in each group, the length of the bool array is 8. Among them, vecstatus[0]-vecstatus[7] respectively correspond to	Returning 0 means successful query, and less than 0 means failed query.	0-3 are for general IO, 4 is used for robot configuration.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	13 / 37

			the DI status from the smallest to the largest serial number in each group. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
12	int GetPosHere(YIROBOT_POS & pos, string & cmdErrInfo)	Get the current TCP based on the robot position data of the current Base coordinate system	pos: the returned position data, see YIROBOT_POS position parameter struct definition for details. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
13	int GetPosHereC(YIROBOT_POS & pos, string & cmdErrInfo)	Get the flange coordinate system (TCP0) based on the robot position data of the base coordinate system (Base0)	pos: the returned position data, see YIROBOT_POS position parameter struct definition for details. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
14	int MovL(YIROBOT_POS pos, long & nMovID, string & cmdErrInfo, int vel = 100, int zone = 0)	Linear move	pos : target position of linear move nMovID: returned move command number. Then the command execution status can be queried based on this	Returning 0 means the command is sent successfully, and less than 0 means the command fails	Running in 3-remote debugging or 4-remote running mode.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	14 / 37

			command number. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero. vel: velocity, the range is 1-100, the default value is 100. zone : blend radius, the default value is 0. (the length of the turning area is zone multiplied by the transitional unit length, with the transitional unit length being 0.1 millimeters, the same below).	to be sent.	
15	<pre>int MovJ(YIROBOT_POS pos, long & nMovID, string & cmdErrInfo, int vel = 100, int zone = 0)</pre>	Joint move	pos : position parameter struct, see definition for details. nMovID: returned move command number. Then the command execution status can be queried based on this command number. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero. vel: movement velocity, the	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 3-remote debugging or 4-remote running mode, used for SCARA models.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	15 / 37

			range is 1-100, the default value is 100. zone: blend radius		
16	<pre>int MovDoor(YIROBOT_POS pos, long & nMovID, YIROBOT_MOVEDOORPARAM para, string & cmdErrInfo, int vel = 100, int zone = 0)</pre>	Door-type move	pos : position parameter struct, see definition for details. nMovID: returned move command number. Then the command execution status can be queried based on this command number. mdpara: MovDoor command parameter struct, which is used to pass multiple parameter information of the MovDoor command. See the definition for details. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero. vel: movement velocity, the range is 1-100, the default value is 100. zone: blend radius	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 3-remote debugging or 4-remote running mode.
17	<pre>int MovC(YIROBOT_POS posMid, YIROBOT_POS posDst, long & nMovID, string & cmdErrInfo, int vel = 100, int zone = 0)</pre>	Circular move	posMid: the position of a point in the middle of the arc. posDst: the position of the end point of the arc. nMovID: returned move	Returning 0 means the command is sent successfully, and less than 0 means the command fails	Running in 3-remote debugging or 4-remote running mode.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	16 / 37

			command number. Then the command execution status can be queried based on this command number. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero. vel: movement velocity, the range is 1-100, the default value is 100. zone: blend radius	to be sent.	
18	<pre>int MovCIO(YIROBOT_POS posMid, YIROBOT_POS posDst, long & nMovID, int vel, int zone, YIMOV_IO movIo[], int ioNum, string & cmdErrInfo)</pre>	Circular move with IO move command, 1-4 inputs and outputs	posMid: the position of a point in the middle of the arc. posDst: the position of the end point of the arc. nMovID: returned move command number. Then the command execution status can be queried based on this command number. vel: movement velocity, the range is 1-100. zone: blend radius movIo: IO control struct, see definition for details. ioNum: the number of groups of IO control struct, the range is 1-4. cmdErrInfo: the error	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 3-remote debugging or 4-remote running mode

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	17 / 37

			information of command execution. It is valid when the return value is less than zero.		
19	int MovLIO(YIROBOT_POS pos, long & nMovID, int vel, int zone, YIMOV_IO movIo[], int ioNum, string & cmdErrInfo)	Linear move with IO move command, 1-4 inputs and outputs	pos: target position nMovID: returned move command number. Then the command execution status can be queried based on this command number. vel: movement velocity, the range is 1-100. zone: blend radius movIo: IO control struct, see definition for details. ioNum: the number of groups of IO control struct, the range is 1-4. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 3-remote debugging or 4-remote running mode.
20	int MovLMultiStepIOPulse(YIROBOT_POS pos, long & nMovID, int vel, int zone, int num, int status, int kind, double durationTime, double stepValue[], int pulseNum, string &	Output multiple (1-25) pulse signals during linear move	pos: target position nMovID: returned move command number. Then the command execution status can be queried based on this command number. vel: movement velocity, the range is 1-100. zone: blend radius	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 3-remote debugging or 4-remote running mode.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	18 / 37

cmdErrInfo)		num: output D0 number (0-39) status : output D0 value (0-OFF, 1-ON) kind: set the IO type during movement (0-time type, 1-path percentage, 2-distance). Currently, only time type and path percentage are supported. Among them, the path percentage is the proportion of the distance from the end point that occupies the entire path. The proportional value range is 0-1. Time is the advance time before reaching the end point, in seconds. durationTime : pulse duration time StepValue : 1-25 time or proportional values determined by kind, the time type is seconds, and the percentage is the decimal between 0-1. pulseNum : the number of pulses, with a range of 1-25. cmdErrInfo: the error information of command		
-------------	--	---	--	--

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	19 / 37

			execution. It is valid when the return value is less than zero.		
21	int MovJIO(YIROBOT_POS pos, long & nMovID, int vel, int zone, YIMOV_IO movIo[], int ioNum, string & cmdErrInfo)	Joint move with IO move command, 1-4 inputs and outputs	pos: target position nMovID: returned move command number. Then the command execution status can be queried based on this command number. vel: movement velocity, the range is 1-100. zone: blend radius movIo: IO control struct, see definition for details. ioNum: the number of groups of IO control struct, the range is 1-4. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 3-remote debugging or 4-remote running mode, used for SCARA models.
22	int MovDoorIO(YIROBOT_POS pos, long & nMovID, YIROBOT_MOVEDOORPARAM para, int vel, int zone, YIMOV_IO movIo[], int ioNum, string & cmdErrInfo)	Door-type move with IO move command, 1-4 inputs and outputs	pos: target position nMovID: returned move command number. Then the command execution status can be queried based on this command number. mdpara: MovDoor command parameter struct, and it is used to pass multiple parameter information of	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 3-remote debugging or 4-remote running mode.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	20 / 37

			MovDoor command. See the definition for details. vel: movement velocity, the range is 1-100. zone: blend radius movIo: IO control struct, see definition for details. ioNum: the number of groups of IO control struct, the range is 1-4. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
23	<pre>int GetRecentlyCompletedCmdNum(long & nmovid, string & cmdErrInfo)</pre>	Query the number of the recently completed move command (MovJ, MovL, MovC, MovDoor)	nmovid: the command number, which represents the result of the query. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
24	<pre>int GetCurCmdID(long & nmovid, string & cmdErrInfo)</pre>	Query the number of the currently executing move command (MovJ, MovL, MovC, MovDoor)	nmovid: the command number, which represents the result of the query. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
25	<pre>int GetCurCmdSts(ref int</pre>	Query the execution	sts: completion status,	Returning 0 means	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	21 / 37

	sts, ref string cmdErrInfo)	status of the currently executing move command	which represents the result of the query. -1-execution completed, abnormal status. 0-execution completed, normal status. 1-executing. 2-waiting for execution. 3-uninitialized. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	successful query, and less than 0 means failed query.	
26	int GetCmdSts(int & sts, string & cmdErrInfo)	Query the execution status of the move command with the specified number	nmoveid: command number sts: completion status, which represents the result of the query. -1-execution completed, abnormal status. 0-execution completed, normal status. 1-executing. 2-waiting for execution. 3-uninitialized. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
27	int GetEmergStopSts(int & sts, string & cmdErrInfo)	Query emergency stop status	sts: the status of the emergency stop switch, which represents the result of the query. 0-emergency stop is released. 1-emergency stop is pressed.	Returning 0 means successful query, and less than 0 means failed query.	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	22 / 37

			cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
28	int MotorEnable(int sts, long & nMovID, string & cmdErrInfo)	Motor enable control	sts: 0-disable, 1-enable. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the motor enable command is completed, and less than 0 means it fails.	The waiting time for command execution is 0-2 seconds.
29	int GetMotorEnableSts(int & sts, string & cmdErrInfo)	Query the current motor enable status	sts: motor enable status, which represents the result of the query. 0-disable, 1-enable. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
30	int GetSysErr(int & sts, string & strRobotErr, string & cmdErrInfo)	Get system fault error information	sts: system error code, which represents the result of the query. 0 means no fault. A non-0 value means there is a fault. For specific fault information, see the parameter robotErrInfo. robotErrInfo: system fault information, normal means no fault or specific fault information.	Returning 0 means successful query, and less than 0 means failed query.	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	23 / 37

			cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
31	int ResetErr(string & cmdErrInfo)	Fault reset	cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means that the fault reset command is completed, and less than 0 means that the fault reset command fails.	
32	int SetWorkMode(int mode, string & cmdErrInfo)	Set work mode	mode: control through sdk belongs to remote mode, and local mode does not support remote sdk control. 0-local mode manual control: remote mode is invalid. Slow speed. The enable button needs to be pressed. The single command is completed in continuous mode. Single-step mode. Execute according to decomposition action. 1-local mode automatic control: remote mode is invalid. Fast speed. No need to press the enable button. In continuous mode, commands in the command	Returning 0 means that the mode setting is successful, and less than 0 means that the mode setting fails.	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	24 / 37

			queue are executed sequentially. Single-step mode. Completely execute the current command. 2-remote teaching: local mode is invalid. Slow speed. No need to press the enable button. Remote enable. Support teaching commands, such as jog commands. 3-remote debugging: local mode is invalid. Slow speed. No need to press the enable button. Remote enable. Teaching commands are not supported. While the current command is being executed or the robot is in running status, the issuance of move commands, teaching commands, and setting commands are invalid. The single command is completed in continuous mode. Single-step mode. Execute according to the decomposition actions. 4-remote running: local mode is not valid. Fast speed. No need to press the enable button. Remote enable. Teaching commands		
--	--	--	--	--	--

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	25 / 37

			are not supported. Support pre-issuance of move commands. In continuous mode, commands in the command queue are executed sequentially. Single-step mode. Completely execute the current command. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
33	<pre>int GetWorkMode(int & mode, string & cmdErrInfo)</pre>	Get work mode	mode: control through sdk belongs to remote mode, and local mode does not support remote sdk control. When setting the work mode, only modes 2, 3, and 4 of the remote mode class can be set. 2-remote teaching: local mode is invalid. Slow speed. No need to press the enable button. Remote enable. Support teaching commands, such as jog commands. 3-remote debugging: local mode is invalid. Slow speed. No need to press the enable button. Remote enable. Teaching commands are not	Returning 0 means successful query, and less than 0 means failed query.	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	26 / 37

			supported. While the current command is being executed or the robot is in running status, the issuance of move commands, teaching commands, and setting commands are invalid. The single command is completed in continuous mode. Single-step mode. Execute according to the decomposition actions. 4-remote running: local mode is not valid. Fast speed. No need to press the enable button. Remote enable. Teaching commands are not supported. Support pre-issuance of move commands. In continuous mode, commands in the command queue are executed sequentially. Single-step mode. Completely execute the current command. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
34	int GetPosHereJ(YIROBOT_JOI	Get the position in the current robot	pos: the returned position data, and see YIROBOT_POS	Returning 0 means successful	Used for SCARA models

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	27 / 37

	NT & jpos, string & cmdErrInfo)	joint coordinate system	position parameter struct definition for details. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	query, and less than 0 means failed query.	
35	int GetRobotType(unsigned int & uiRobotType, string & cmdErrInfo)	Get robot series type	uiRobotType: robot series type 0x0001-Delta model 0x0008-SCARA model 0x0010-Diamond model 0x0040-Wafer handling model(sdk is currently not supported.) 0x0200-Mantis model(sdk is currently not supported.) 0xffff-Basic model cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
36	int GetRobotSubType(unsigned int & uiRobotSubType, string & cmdErrInfo)	Get robot series subtype	uiRobotSubType: robot series subtypes, and only Wafer handling models have subtypes. 0-undefined subtype 1-101- one-arm wafer handling type 102-two-arm wafer handling type	Returning 0 means successful query, and less than 0 means failed query.	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	28 / 37

			cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
37	int GetRobotModel(string & strRobotModel, string & cmdErrInfo)	Get robot model	strRobotModel: robot model cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
38	int SetVel(int vel, string & cmdErrInfo)	Set the current robot velocity	vel: velocity percentage, the range is 1-100. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means that the setting is successful, and less than 0 means that the setting fails.	
39	int GetVel(int & vel, string & cmdErrInfo)	Get the current robot velocity	vel: velocity percentage, the range is 1-100. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
40	int SetAccRamp(int startAcc, int endAcc, string & cmdErrInfo)	Set the acceleration of the movement segment	startAcc: the acceleration percentage of the starting segment, the range is 10-100. endAcc: the acceleration percentage of the end segment, the range is	Returning 0 means that the setting is successful, and less than 0 means that the setting fails.	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	29 / 37

			10-100. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
41	int GetAccRamp(int & startAcc, int & endAcc, string & cmdErrInfo)	Get acceleration	startAcc: the acceleration percentage of the starting segment, the range is 10-100. endAcc: the acceleration percentage of the end segment, the range is 10-100. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means successful query, and less than 0 means failed query.	
42	int Jog(int coordinate, int axis, int cmd, int mode, double stepValue, string & cmdErrInfo)	Teaching move commands	coordinate: teaching method 0-World coordinate system 1-Base coordinate system 2-Tcp coordinate system 3-Joint coordinate system 4-external axis axis: axis number, the range is 1-6, which corresponds to the J1-J6 axes when teaching joints, and corresponds to X/Y/Z/RZ/RX when teaching Cartesian coordinates.	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Running in 2-remote teaching mode

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	30 / 37

			cmd: teaching command, 0-stop, 1-forward teaching, 2-inverse teaching. mode: continuous or inching mode 0-continuous mode 1-inching mode stepValue : inching step length. When axis is XYZ, it is the length, in millimeters. When axis is RXRYRZ or J1-J6, it is the angle, in degrees. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.		
43	int Pause(string & cmdErrInfo)	Pause command	cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	
44	int Resume(string & cmdErrInfo)	Resume command	cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the command is sent successfully, and less than 0 means the command fails	

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	31 / 37

				to be sent.	
45	<code>int End(string & cmdErrInfo)</code>	End command	cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	
46	<code>int SetWaitTime(int milisectime, string & cmdErrInfo)</code>	Set the timeout waiting time for sending messages and waiting for receiving replies on the SDK side. The sdk side waits for timeout and directly returns a network connection error.	milisectime: timeout time in milliseconds. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	
47	<code>int SetRobotWaitTime(int milisectime, string & cmdErrInfo)</code>	Set the waiting time for the robot side to detect the SDK side disconnection timeout. After the robot detects that the SDK side has disconnected and timed out, it calls enable	milisectime: timeout time in milliseconds. cmdErrInfo: the error information of command execution. It is valid when the return value is less than zero.	Returning 0 means the command is sent successfully, and less than 0 means the command fails to be sent.	Based on actual usage feedback, the timeout disable function is currently not used.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	32 / 37

		to stop movement.			
--	--	-------------------	--	--	--

3. Data definition usage

3.1 Data classes definition

```
//Robot position data
struct YIROBOT_POS
{
    //Robot coordinate position and arm parameter data
public:
    double X=0;//X
    double Y=0;//Y
    double Z=0;//Z
    double RX=0;//RX
    double RY=0;//RY
    double RZ=0;//RZ
    int armType1=0;//Arm parameter 1, used for SCARA model, 0-left arm,
    1-right arm.
    int armType2=0;
    int armType3=0;
    int armType4=0;//Arm parameter 4, used for SCARA or Delta models,
    -180-180 degrees represents the 0th circle. When multiple circles are
    involved, the arm parameter increases by 1 for every 360 degree increase
    and decreases by 1 for every 360 degree decrease.
```

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	33 / 37

```

    int toolNo=0;//Tool coordinate number

    int userNo=0;//User coordinate number

};

//Robot position and external axis data in joint coordinate system.
struct YIROBOT_JOINT
{
    //Robot joint coordinate system position and external axis data.
public:
    double axis1=0;//The angle of the first axis of the joint coordinate
    system, in degrees.

    double axis2=0;//The angle of the second axis of the joint coordinate
    system, in degrees.

    double axis3=0;//The angle of the third axis of the joint coordinate
    system, in degrees.

    double axis4=0;//The angle of the fourth axis of the joint coordinate
    system, in degrees.

    double axis5=0;//The angle of the fifth axis of the joint coordinate
    system, in degrees.

    double axis6=0;//The angle of the sixth axis of the joint coordinate
    system, in degrees.

    double exAxis1=0;//The angle of the first axis of the external axis,
    in degrees.

    double exAxis2=0;//The angle of the second axis of the external axis,
    in degrees.

    double exAxis3=0;//The angle of the third axis of the external axis,
    in degrees.

    double exAxis4=0;//The angle of the fourth axis of the external axis,
    in degrees.

```

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	34 / 37

```

    double exAxis5=0;//The angle of the fifth axis of the external axis,
in degrees.

    double exAxis6=0;//The angle of the sixth axis of the external axis,
in degrees.
};

//Move command related IO data.
struct YIMOV_IO
{
//Move command related IO output.
    int ioNo=0;//IO number
    int ioVa=0;//IO output value (0-OFF, 1-ON)
    int kind=0;//Set the type of IO during movement (0-time type, 1-path
percentage, 2-distance), currently only time type is supported.

    double value=0;

    //When the type is time, it means that the signal is output Value seconds
before reaching the movement point.

    //When the type is path percentage, it means that the signal is output
when Value% of the entire path is completed from the start of movement.

    //When the type is distance, greater than 0 means that the signal is
output after the movement starts from the starting point to Value
millimeters, and less than or equal to 0 means that the signal is output
before the movement reaches Value millimeters from the end point.
};

//Robot movdoor command parameter data
struct YIROBOT_MOVEDOORPARA
{
//Robot movdoor command parameter data
public:

```

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	35 / 37

```

double UpDistance=0;//Up distance,it should not exceed the robot's
workspace.

double DownDistance=0;//Down distance,it should not exceed the
robot's workspace.

double UpBlendR=0;//Up blend ratio,the ratio of turning length to up
length, between 0-1.

double DownBlendR=0;//Down blend ratio,the ratio of turning length
to down length, between 0-1.

double RotPercent=0;//Rotation percentage, the percentage of time
taken by the rotation process to the total time of the command movement,
between 0-1. In order to avoid collision with surrounding objects when
rotating during the pick-and-place process, it should be less than 1, and
the default is 0.5.

int MiddleMode=0; //Middle segment mode, suitable for SCARA models,
0 middle segment is linear move MovL, 1 is the middle segment joint move
MovJ, currently only the middle segment linear move 0 is supported.
};

```

3.2 Function return value faults and solutions

0: (normal command)

-1:The robot command failed to be sent or an error occurred during the execution of the command by the robot. For specific error information, see the return parameter `cmdErrInfo`.

-255: Network communication is abnormal, check the network connection and re-call after confirming it is correct.

Document Name	Robotphoenix Robot-PCSDK (C++) Development and User Manual	Document Number	RP-KZ-01-014-C-V1.0	Version	V1.0
Centralized Management	Technical Pre Research Department	Revision Date	2023.10.08	Page	36 / 37

-256: Network communication times out while waiting for a reply. Check the running status of the network and robot programs. Re-call if correct. If necessary, set a longer waiting time.

3.3 Data Units

Support metric unit system definition, suitable for unspecified unit data.

Angle: Degree

Time: Second

Length: mm