



Modbus Communication Manual

Copyright and Disclaimer

The information contained in this manual is subject to change without prior notice and should not be regarded as a commitment by Hangzhou Robotphoenix Industrial Robotics Co., Ltd. (hereinafter referred to as Robotphoenix). Except as expressly stated anywhere in this manual, nothing in this manual should be regarded as any type of guarantee or warranty made by Robotphoenix for the loss of personnel or property of the user, personal injury or property damage, fitness for a particular purpose or similar matters.

Without the written permission of Robotphoenix, it is strictly prohibited to reprint or copy this manual and any part of it, disclose any content to a third party, or use it for any unauthorized purpose. Violators of this clause will be prosecuted.

Contents

1. Modbus Application Overview	1
1.1 Modbus communication protocol	1
1.2 Modbus application scenario	1
2. Application of Modbus TCP	2
2.1 Communication connection	2
2.2 Modbus TCP slave configuration	2
2.3 Modbus TCP master configuration	6
2.4 Modbus TCP variable settings	11
2.5 Modbus TCP variable usage	15
3. Use of ModbusRTU	18
3.1 Communication Connection	18
3.2 Modbus RTU slave configuration	23
3.3 Modbus RTU master configuration	26
3.4 Modbus RTU variable settings	29
3.5 Modbus RTU variable usage	332
4. Modbus control application	35
4.1 Introduction to reading working status function	35
4.2 Error reset function introduction	42
4.3 Modbus emergency stop reset function introduction	46
5. Modbus remote jog	52
5.1 Introduction to the function of reading robot speed	52

5.2 Start remote teaching mode	60
5.3 Set the robot speed in remote teaching mode	64
5.4 Jog operation in remote teaching mode	67
5.5 PTP operation in remote teaching mode	72
6. Robot Modbus slave address allocation table	78
6.1 Bits(Coil)	78
6.2 InputBits	78
6.3 HoldRegisters	78
6.4 InputRegisters	84

1. Modbus Application Overview

1.1 Modbus communication protocol

Modbus communication protocol is a standard industrial fieldbus protocol. Robotphoenix King Kong system supports Modbus RTU protocol based on RS485 bus connection, and also supports Modbus TCP protocol based on Ethernet connection. Modbus TCP features: Modbus TCP uses standard TCP/IP to package and compress Modbus information packets. In this way, Modbus TCP devices can be connected and communicated through Ethernet. Compared with the RS-485 interface, Modbus TCP also allows the use of more addresses, can adopt a multi-master station architecture, and the transmission rate can reach GB/S. The number of slave stations in the Modbus TCP network is only limited by the capabilities of the network physical layer.

1.2 Modbus application scenarios

In the on-site Modbus application, Robotphoenix robot can be used as a control object, that is, as a Modbus master station to control others, and can also be used as a controlled object, that is, as a Modbus slave station. This document mainly focuses on the connection of Modbus as a master and slave station, as well as the application instructions for reading the robot's manual automatic status, error reset, etc. Under the premise that the robot is configured as a Modbus slave, the master device can operate the robot's manual and automatic states according to the Modbus address

table corresponding to the robot. When the robot is not configured as a Modbus slave, the corresponding Modbus function is invalid.

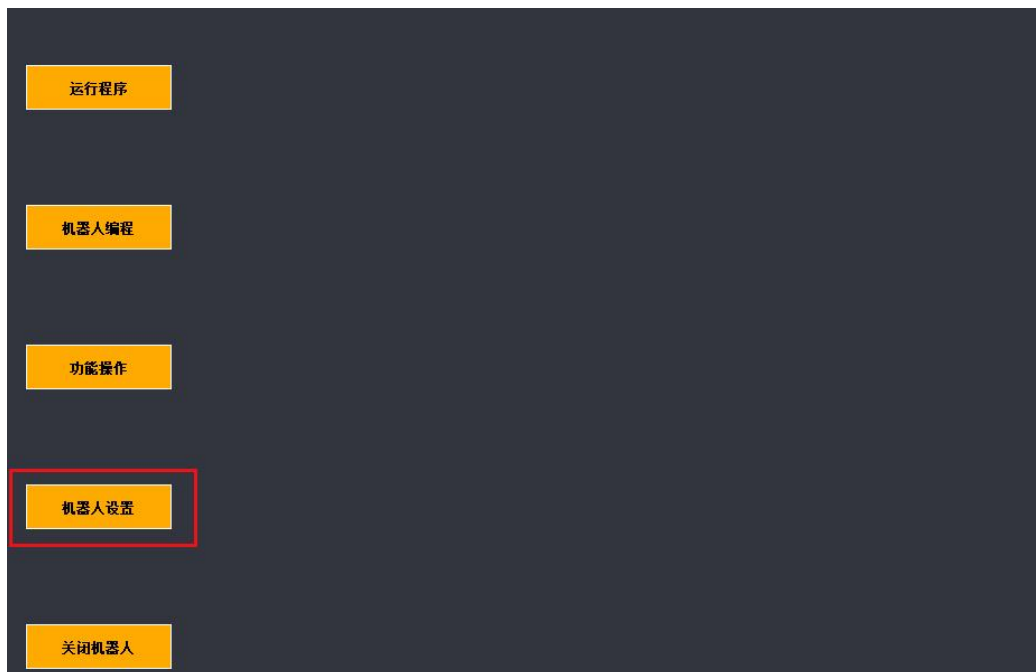
2. Applications of Modbus TCP

2.1 Communication connection

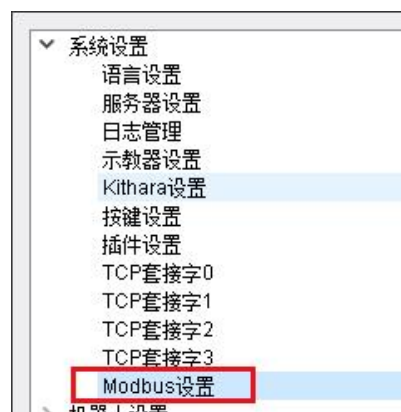
Other devices are connected to the network port of the robot control cabinet via a network cable.

2.2 Modbus TCP Slave Configuration

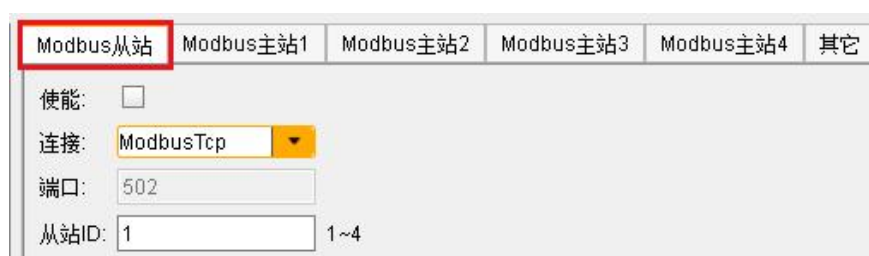
(1) Open YiKingSoft software and click the "Robot Settings" button to enter the robot settings interface.



(2) Enter the robot settings interface and select "Modbus settings".



(3) Select the "Modbus Slave" tab



(4) Enable: Select "Enable" for Modbus slave.

RP-KZ-01-004-C

Modbus从站	Modbus主站1	Modbus主站2	Modbus主站3	Modbus主站4	其它
使能: ☒ 连接: ModbusTcp 端口: 502 从站ID: 1 1~4					

(5) Connection: Select "ModbusTcp" mode

Modbus从站	Modbus主站1	Modbus主站2	Modbus主站3	Modbus主站4	其它
使能: ☒ 连接: ModbusTcp 端口: 502 从站ID: 1 1~4					

(6) Port: Default is 502 and cannot be changed.

Modbus从站	Modbus主站1	Modbus主站2
使能: ☒ 连接: ModbusTcp 端口: 502 从站ID: 1 1~4		

(7) Slave ID: Enter the slave address (range: 1-4)

Modbus从站	Modbus主站1	Modbus主站2
使能: ☒ 连接: ModbusTcp 端口: 502 从站ID: 1 1~4		

(8) Click the "Save" button to save the Modbus slave settings.

(9) Click the "Other" tab to set the high and low order

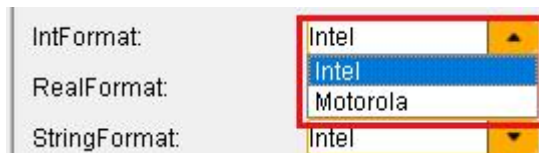
For data in Int, Real, and String formats, you can set each type to store data in the order of the high 8 bits in front or the low 8 bits ahead. The default is to store the low 8 bits ahead.

Note: There are two options:

INTEL——lower 8 bits ahead MOTOROLA——high 8 bits in front

Modbus从站	Modbus主站1	Modbus主站2	Modbus主站3	Modbus主站4	其它																		
<table> <tr> <td>IntFormat:</td> <td>Intel</td> <td>▼</td> <td colspan="3">Intel low bytes ahead,Motorola high byte in front</td> </tr> <tr> <td>RealFormat:</td> <td>Intel</td> <td>▼</td> <td colspan="3">Intel low bytes ahead,Motorola high byte in front</td> </tr> <tr> <td>StringFormat:</td> <td>Intel</td> <td>▼</td> <td colspan="3">Intel low bytes ahead,Motorola high byte in front</td> </tr> </table>						IntFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front			RealFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front			StringFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front		
IntFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front																				
RealFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front																				
StringFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front																				

Example: A certain brand of PLC stores data with the high 8 bits in front. The PLC stores an integer 1 (Hex0001) in a register. The PLC writes this number into a register of the robot. This register is associated with an input int variable, and the value of this variable is 256 (Hex0100). Then we can change IntFormat to Motorola mode.



(10) Set up read and write blocks

There is no need to set up read/write blocks when the robot acts as a slave.

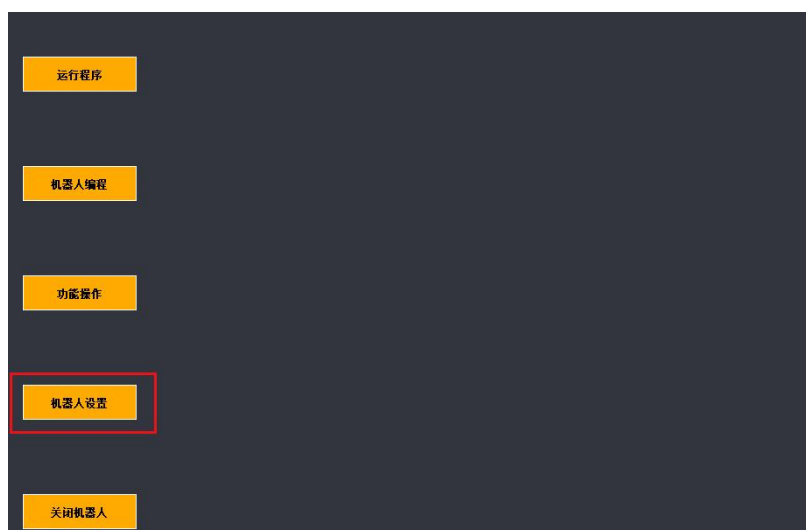
(11) Slave address

When the robot acts as a slave, there are 400 holding registers for the master to read and write, of which 0-99 are system registers, used for external input start, stop, pause, resume and other functions; 100-399 are allocated to users to customize communication through instructions. When the robot acts as a slave, there are 200 input registers for the master to read only.

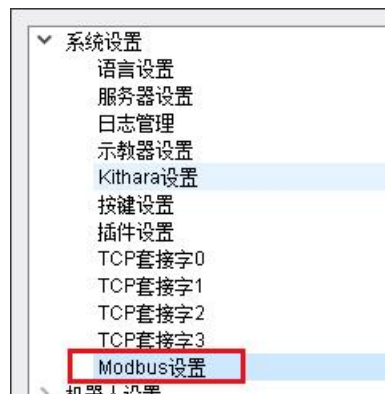
For details, please see the attached: Robot Modbus slave address allocation table

2.3 Modbus TCP Master Configuration

(1) Open YiKingSoft software and click the "Robot Settings" button to enter the robot settings interface.



(2) Enter the robot settings interface and select "Modbus settings".

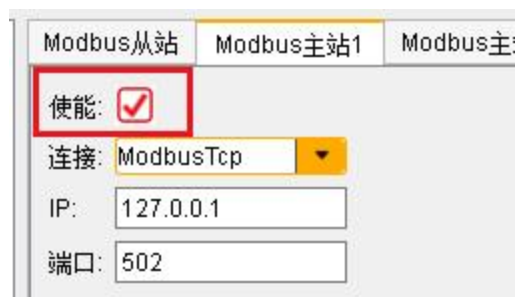


(3) You can choose Modbus Master 1/Modbus Master 2/Modbus Master 3/Modbus Master

4. There are four master stations in total. Each master station can be configured separately, or only one or several of them can be enabled.



(4) Enable: Select “Enable” for Modbus master.



(5) Connection: Select ModbusTCP mode

Modbus从站 Modbus主站1 Modbus从站2

使能: ☒

连接: ModbusTcp

IP: 127.0.0.1

端口: 502

(6) IP: Enter the IP address of the communication partner. The IP can be freely defined.

Modbus从站 Modbus主站1 Modbus从站2

使能: ☒

连接: ModbusTcp

IP: 127.0.0.1

端口: 502

(7) Port: Default is 502 and cannot be changed.

Modbus从站 Modbus主站1 Modbus从站2

使能: ☒

连接: ModbusTcp

IP: 127.0.0.1

端口: 502

(8) Click the "Save" button to save the settings.

Modbus从站	Modbus主站1	Modbus主站2	Modbus主站3	Modbus主站4	其它
使能: ☒ 连接: ModbusTcp IP: 127.0.0.1 端口: 502					
复位 保存 返回					

(9) Set the high-low order

Select the "Other" tab. For data in Int, Real, and String formats, you can set each type to store data in the order of the high 8 bits in front or the low 8 bits ahead.

The default is to store the low 8 bits ahead.

Two options: INTEL—lower 8 bits ahead MOTOROLA—higher 8 bits in front

Modbus从站	Modbus主站1	Modbus主站2	Modbus主站3	Modbus主站4	其它
IntFormat: Intel Intel low bytes ahead, Motorola high byte in front RealFormat: Intel Intel low bytes ahead, Motorola high byte in front StringFormat: Intel Intel low bytes ahead, Motorola high byte in front					

Example: A certain brand of PLC stores data with the high 8 bits in front. If the

PLC stores a string AB in a register, the robot reads the register and saves the value in the Modbus String variable as BA. Then we can change the StringFormat to Motorola mode.

(10) Set read and write blocks

A maximum of two blocks can be set, and both can be enabled by "enable" or only one can be enabled.

Note: 1. Register addresses cannot overlap, input and output addresses within a block cannot overlap, and addresses between two blocks cannot overlap.

2. The address range must match the address space allowed by the slave and cannot exceed the limit. For example, our read block is 0~11, a total of 12 addresses, and the slave only has 10 addresses from 1 to 9, so the register value cannot be read.

Modbus从站	Modbus主站1	Modbus主站2	Modbus主站3	Modbus主站4	其它
IntFormat:	Intel				Intel low bytes ahead, Motorola high byte in front
RealFormat:	Intel				Intel low bytes ahead, Motorola high byte in front
StringFormat:	Intel				Intel low bytes ahead, Motorola high byte in front
RegisterBlock1Enable:	☒				
RegisterInBlock1Start:	0				
RegisterInBlock1End:	99				
RegisterOutBlock1Start:	100				
RegisterOutBlock1End:	199				
RegisterBlock2Enable:	☒				
RegisterInBlock2Start:	200				
RegisterInBlock2End:	299				
RegisterOutBlock2Start:	300				
RegisterOutBlock2End:	399				

(11) If you set the read and write blocks in "10", after setting, click the "Save" button to save the settings.

(12) Restart YiKingSoft to make all the above changes take effect.

2.4 Modbus TCP variable settings

1、Click the "Robot Programming" button to enter the robot programming interface.



2、Click on the "Variables" tab on the right → find the Modbus tab.



3、Press the "Add" button to enter the add variable interface.

RP-KZ-01-004-C

索引	名称	类型	值	S/C	S/CID	输入/输出	Reg	Bit	Number	Min	Max	公共
----	----	----	---	-----	-------	-------	-----	-----	--------	-----	-----	----

添加

删除

4、Select the Base tab

Base

Modbus

名称:

类型:

Bool

变量数量:

1

值:

false

使用范围限制:

false

最小值:

最大值:

公共属性:

false

注释

添加

取消

name	Enter the variable name. (Format: 1. Start with a letter + a number 2. Pure letters. Cannot be Chinese characters or pure numbers. Cannot be a keyword, and cannot be the same as an existing variable)			
type	variable	storage	scope	Accuracy
	Bool	1bit	False/True	
	int type	1word	-32768—32767	
		2word	-2147483648—2147483647	
	Double	2word	-/+3.4e38	Display 4 decimal places, up to 7 significant digits
		4word	-/+1.7e308	Display 4 decimal places, up to 16 decimal places
	String	1byte		
Number of	The specific number of variables			

variables	
value	The initial value of the variable, fill in the appropriate value according to the variable type
Scope of use	Flase: Do not set the value range of the variable, and the maximum and minimum values below cannot be entered; TRUE: Set the value range of the variable, and enter the maximum and minimum values.
Minimum	Maximum value of variable
Maximum	Variable minimum value
Public	False: This variable is not visible in the user interface
Properties	True: This variable is visible in the UI

5、Select the Modbus tab

BaseModbus

BOOL 型变量时

Station Type:Client

Station ID:

Reg Type:Input

Reg Address:

Bit Address:

Reg Length:1

添加

取消

BaseModbus

其他型变量时

Station Type:Client

Station ID:

Reg Type:Input

Reg Address:

Reg Length:1

添加

取消

name	effect
------	--------

Station Type	Modbus master/slave selection
Station ID	Visible when the master station is selected. Used to fill in the address of the slave station.
Reg Type	Input/Output Selection
Reg Address	Fill in the register address
BitAddress	(Visible when the variable type is bool) Fill in the bit address (range: 0-15). Example: When a bool variable test_bool is set, Reg Address = 2, Bit Address = 10, the state of the 10th bit of register 2 is the state of test_bool.
Reg Length	(Can be set when the variable is a type other than bool) Select/fill in the number of registers. The int type variable currently supports 16-bit int and 32-bit int, and you can choose 1 or 2 registers. Double type variables currently support 32-bit int and 64-bit double, and can choose 2 or 4 registers. The length of a string variable needs to be defined according to the length of the string to be sent or received. One letter occupies one byte.

2.5 Modbus TCP variable usage

Modbus variables are used in the same way as general local variables and global

variables. After Modbus communication is established, Modbus variables are constantly refreshed. You can see the current value being refreshed in the variable interface or debug interface, regardless of whether there is a program running. You can see the Modbus variable selection list in the expression editor.

InPut:	Input0
OutPut:	Output0
Var:	
GlobalVar:	
ModbusVar:	text_bool
Point:	
AD:	AD0
ADReal:	ADReal0

Modbus variable use cases

The robot is the master station, and the PLC is the slave station. Create a Modbus input Double variable `x_offset`, which is the offset of the object along the x direction calculated by the PLC, in mm. Create a Modbus output int variable `count` to tell the PLC how many objects the robot has picked up.

Use the `MoveDoor` instruction as the grab position. Before grabbing, use the `ToolOffset` instruction to offset the position of `MoveDoor`. The offset can be associated with the Modbus input variable `x_offset`. After grabbing, the Modbus

variable count is incremented by 1.

The current status of the two variables in the Modbus variable table are as follows, and are constantly refreshed.

x_offset	double	10.3400	Client	1	In	101	0	2
count	int	3	Client	1	Out	112	0	1

The variables are used in the program as follows. In the current state, when the robot executes MoveDoor, it is offset 10.34mm in the X direction from the original position and tells the PLC that it has grabbed 3 objects.



As long as the Modbus communication is established, the robot, as the master, will automatically read the 101st register of the PLC and put the read value into the variable x_offset; it will automatically write the value of the variable count into the 112th register of the PLC. The robot will automatically complete these operations.

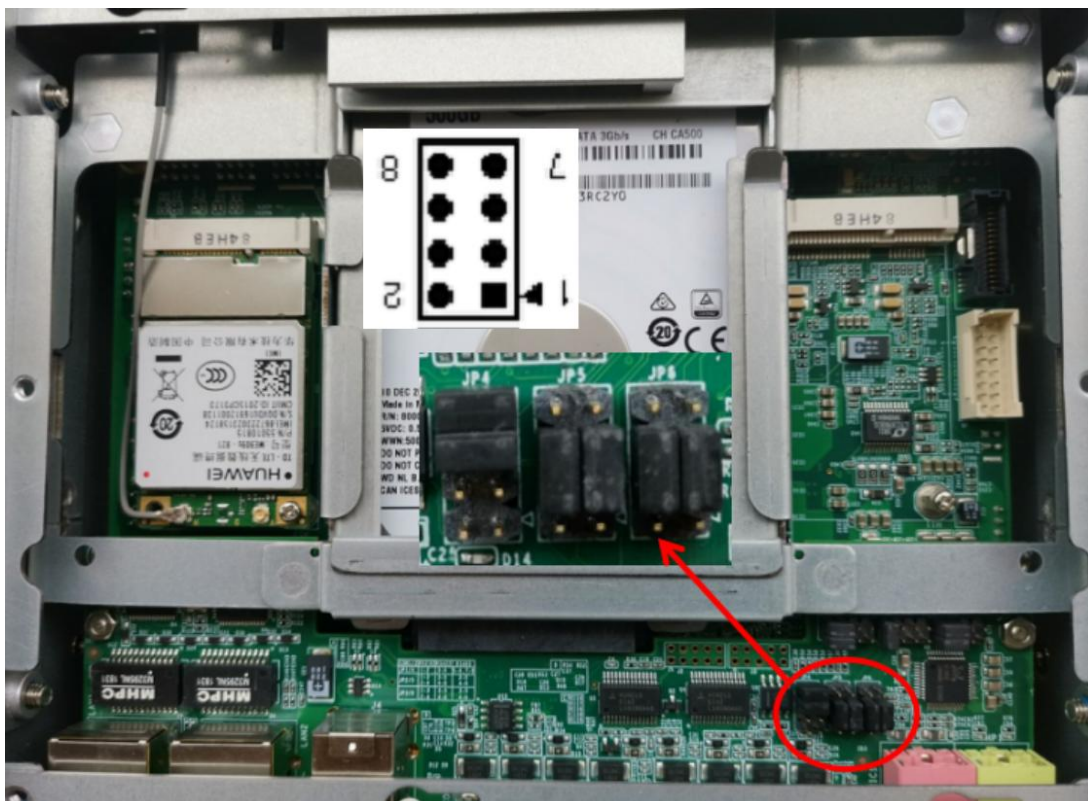
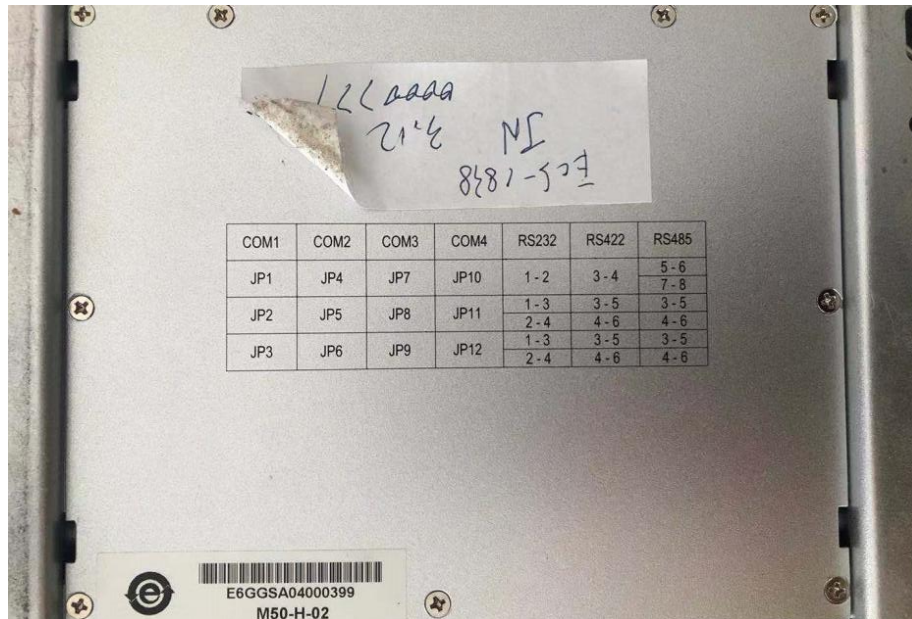
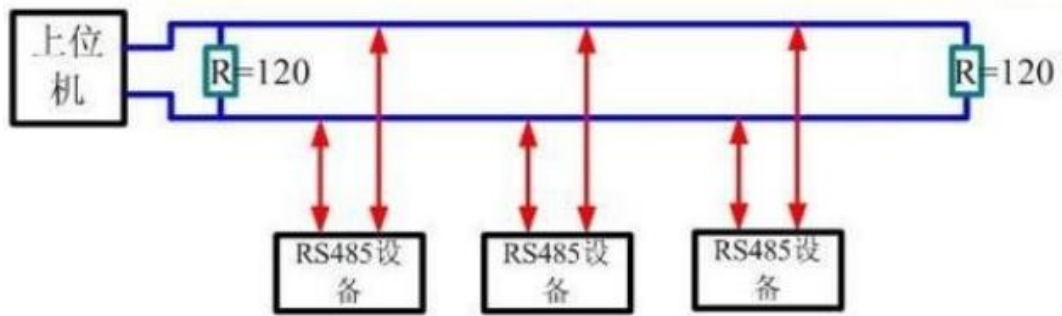
The user only needs to correctly set the Modbus function and use the Modbus variables.

3. Use of ModbusRTU

3.1 Communication connection

(1) First, change the COM port of the industrial computer to RS485 mode (the default is RS232). Taking the EV0C industrial computer as an example, please open the back cover of the industrial computer and configure the jumper on the circuit board according to the information marked on the back cover to convert the COM2 port to RS485 mode. Short-circuit JP4's 5, 6, 7, 8 pins; short-circuit JP5's 3, 5, 4, 6 pins; short-circuit JP6's 3, 5, 4, 6 pins. As shown in the figure, the pin marked with a small triangle is pin 1.

(2) If the communication distance exceeds 300 meters, terminal resistance needs to be added. Only one terminal resistance is added at the beginning and end, as shown in the figure below. If the industrial computer COM2 port needs to enable terminal resistance, it is also necessary to short-circuit JP5's 7 and 8 pins, and JP6's 7 and 8 pins.

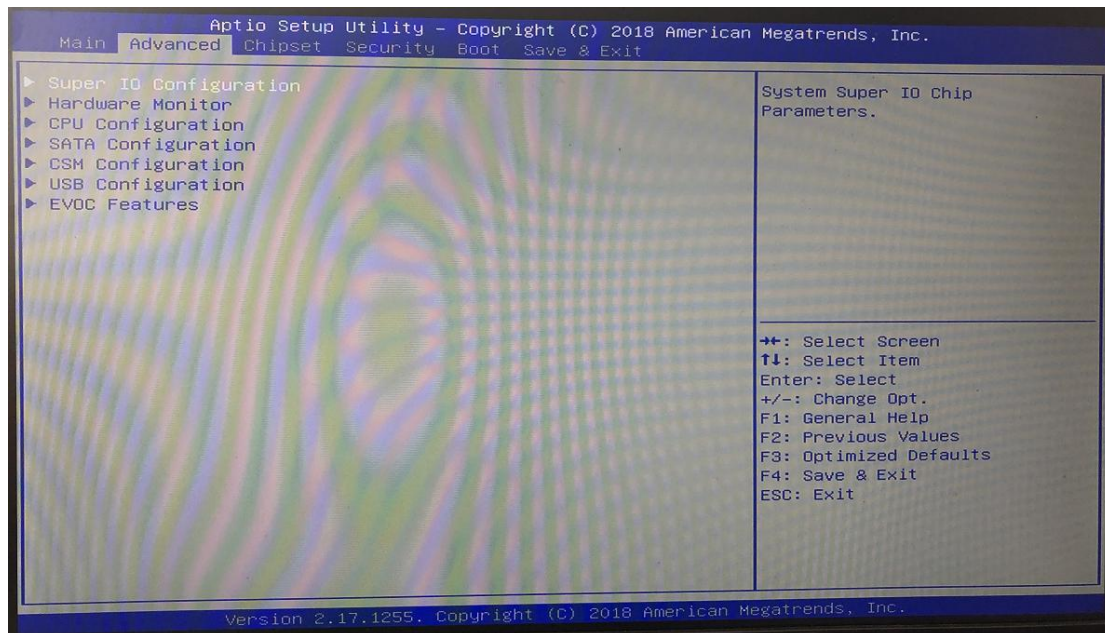


RP-KZ-01-004-C

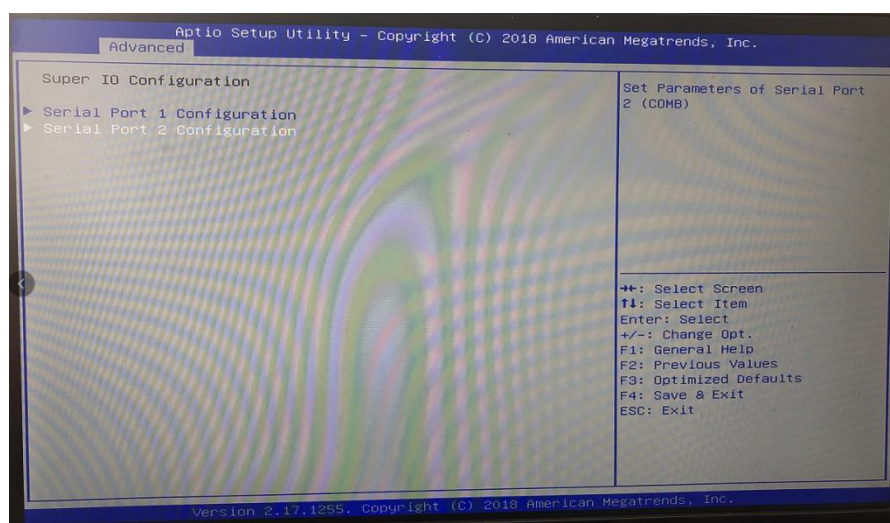
(3) Change the serial port flow control configuration.

a. Plug in the keyboard and restart the industrial computer. Press and hold the Del key while booting to enter the BIOS setup.

b. Select the Advanced tab, select Super IO Configuration, and press Enter

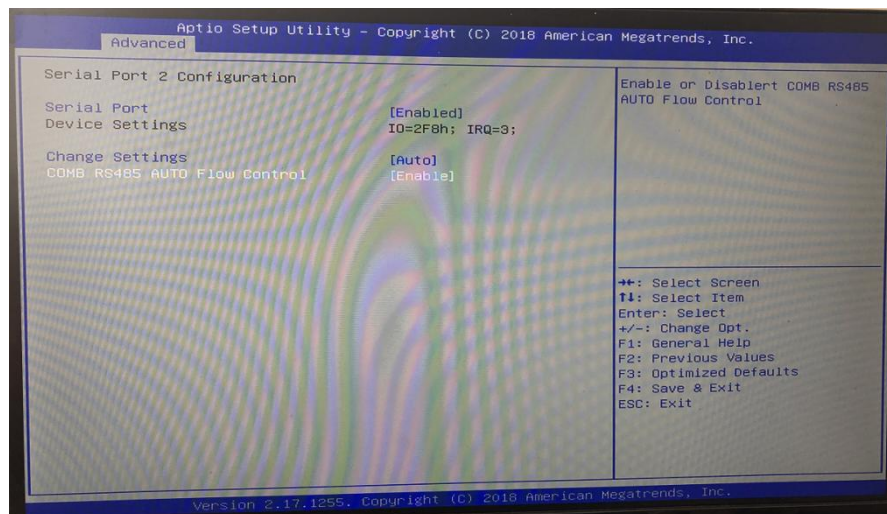


c. Select Serial Port 2 Configuration and press Enter

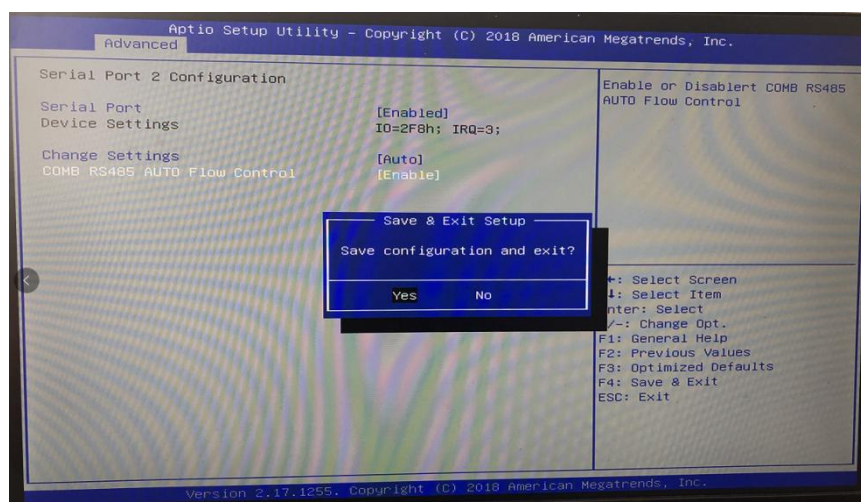


d. Select COMB RS485 AUTO Flow Control, and press the + or - key to switch the status

to Enable.

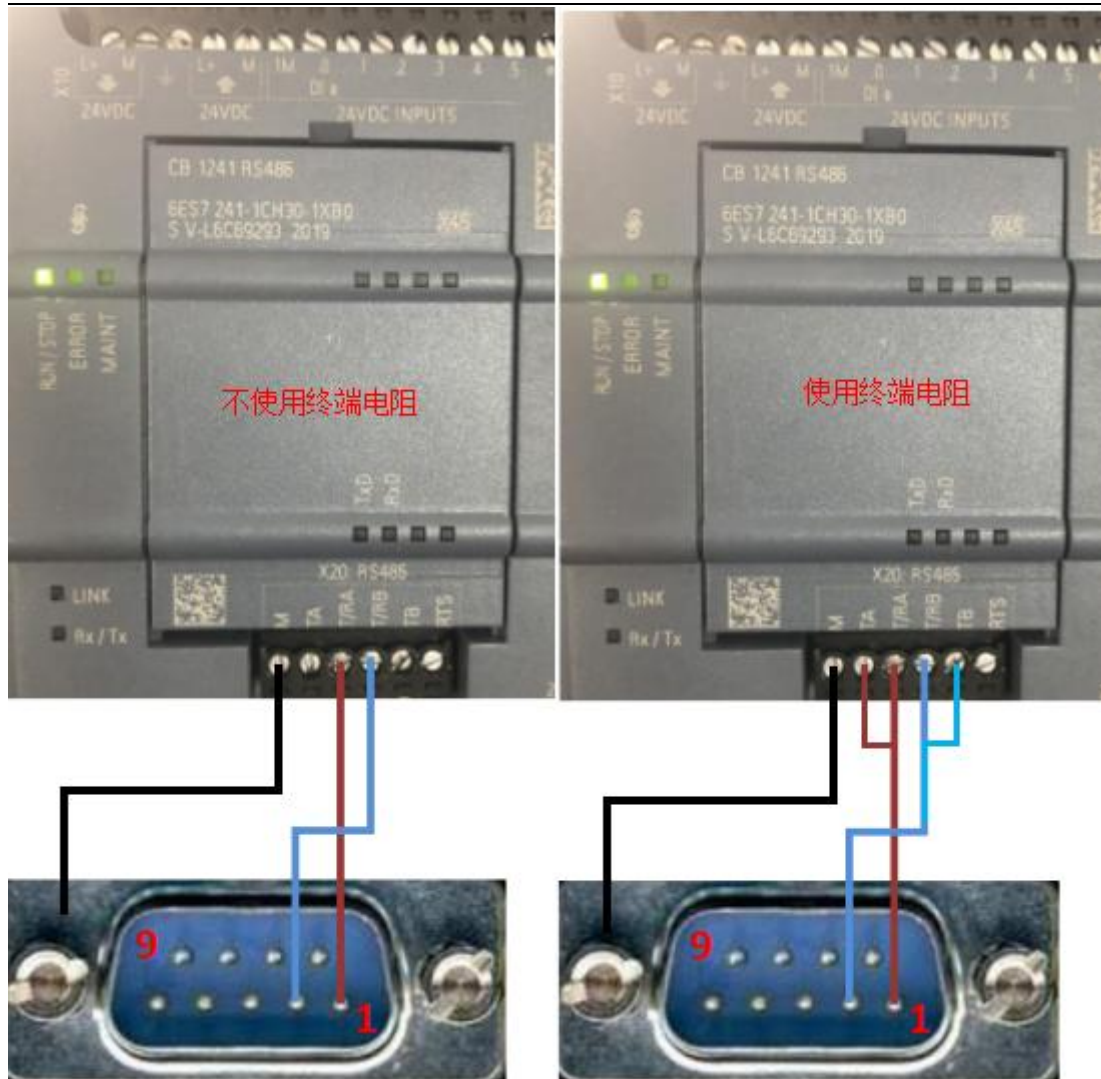


e. Press F4 to save and exit



(4) Connect the Siemens RS485 communication board and the COM2 port of the industrial computer as shown in the figure.

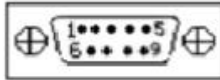
RP-KZ-01-004-C



If you need to use the terminal resistor on the Siemens RS485 communication board, please short-circuit TA and T/RA, and then short-circuit TB and T/RB.

Advantech industrial computer serial port pins are as follows :

管脚	信号名称		
	RS-232 (Default)	RS-422	RS-485
1	DCD#	TXN	DATA-
2	RXD	TXP	DATA+
3	TXD	RXP	NC
4	DTR#	RXN	NC
5	GND	GND	GND
6	DSR#	NC	NC
7	RTS#	NC	NC
8	CTS#	NC	NC
9	RI#	NC	NC



COM1 ~ COM4

3.2 Modbus RTU slave configuration

(1) Open yikingsoft and click  to enter the robot settings interface

(2) Enter the robot settings interface and select ModbusSet settings.



(3) Select ModbusMaster1/ModbusMaster2/ModbusMaster3/ModbusMaster4

There are four master stations in total. Each master station can be configured separately, or only one or several of them can be enabled.

(4) Select the ModbusSlave tab

(5) Check the slave station (Enable) to enable

(6) Select SerialPort mode in Connection

(7) Enter the serial port name

(8) Select the appropriate baud rate

(9) SlaveID input slave address (range: 1-4)

ModbusSlave

ModbusMaster1

ModbusMaster2

从站选项卡

Enabled:

☒

勾选使能

Connection:

SerialPort

▼

选择串口模式

串口名称:

COM2

串口名称

波特率:

115200

▼

选择波特率

数据位:

8

校验位:

NoParity

▼

停止位:

1

SlaveID:

1

从站地址

1~4

(10) Save Settings

(11) Set the high-low order

Select the Other tab. For int, real, and string data, you can set each type to store data in the order of the high 8 bits in front or the low 8 bits ahead. The default is to store data in the order of the low 8 bits ahead.

Two options: INTEL—low 8 bits ahead

MOTOROLA—high 8 bits in front

ModbusSlave	ModbusMaster1	ModbusMaster2	ModbusMaster3	ModbusMaster4	Other
IntFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front		
RealFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front		
StringFormat:	Intel	▼	Intel low bytes ahead,Motorola high byte in front		

For example, a certain brand of PLC stores data with the high 8 bits in front. The PLC stores an integer 1 (Hex0001) in a register, and writes this number into a

register of the robot. This register is associated with an input int variable, and the value of this variable is 256 (Hex0100). Then we can change IntFormat to Motorola mode.

(12) Set read and write blocks

There is no need to set up read/write blocks when the robot acts as a slave.

(13) Slave Address

When the robot is a slave, there are 400 holding registers for the master to read and write, including

0-99: System register, used for external input start, stop, pause, resume, etc.

100-399: Assigned to users to customize communications through instructions

When the robot acts as a slave, there are 200 input registers for the master to read only.

For details, please see the attached: Robot Modbus slave address allocation table

3.3 Modbus RTU Master Configuration

(13) Open yikingsoft and click  the button to enter the robot setting interface.

(14) Enter the robot settings interface and select modbusSet settings.



(15) Check the Master Station (Enable) to enable

(16) Select SerialPort mode in Connection

(17) Enter the serial port name

(18) Select the appropriate baud rate

(19) Select the appropriate communication format: data bits, parity bits, stop

bits



(20) Save Settings

(21) Set the high-low order

Select the Other tab. For int, real, and string data, you can set each type to store

data in the order of the high 8 bits in front or the low 8 bits ahead. The default is to store data in the order of the low 8 bits ahead.

Two options: INTEL—low 8 bits ahead

MOTOROLA—high 8 bits in front

ModbusSlave	ModbusMaster1	ModbusMaster2	ModbusMaster3	ModbusMaster4	Other
IntFormat:	Intel	Intel low bytes ahead,Motorola high byte in front			
RealFormat:	Intel	Intel low bytes ahead,Motorola high byte in front			
StringFormat:	Intel	Intel low bytes ahead,Motorola high byte in front			

For example, a certain brand of PLC stores data with the high 8 bits in front. If the PLC stores a string AB in a register, the robot reads the register and saves the value in the modbus string variable as BA. Then we can change the StringFormat to Motorola mode.

(22) Set read and write blocks

A maximum of two blocks can be set, and both can be enabled or only one can be enabled.

Note: 1. Register addresses cannot overlap, input and output addresses within a block cannot overlap, and addresses between two blocks cannot overlap.

2. The address range must match the address space allowed by the slave and cannot exceed the limit. For example, our read block is 0~11, 12 addresses, and the slave only has 1~9 10 addresses, so the register value cannot be read.

RegisterBlock1Enable:	☒
RegisterInBlock1Start:	0
RegisterInBlock1End:	24
RegisterOutBlock1Start:	25
RegisterOutBlock1End:	49
RegisterBlock2Enable:	☒
RegisterInBlock2Start:	50
RegisterInBlock2End:	54
RegisterOutBlock2Start:	55
RegisterOutBlock2End:	59

(23) Save Settings

(24) Restart yikingsoft to make all the above changes take effect

3.4 Modbus RTU variable settings

1. Click  to enter the robot programming interface
2. Select the modbus tab under the variables tab



3. Press the Add  button
4. Select the Base tab

The image shows a configuration window for a Modbus variable. It has two tabs: 'Base' and 'Modbus', with 'Modbus' currently selected. The window contains several input fields and dropdown menus. The 'Name' field is empty. The 'Type' dropdown is set to 'Bool'. The 'Value' dropdown is set to 'false'. The 'Use Range' dropdown is set to 'false'. Below these are empty input fields for 'Min Value' and 'Max Value'. The 'Add to Parameter List' dropdown is set to 'false'. At the bottom, there are two buttons: '添加' (Add) and '取消' (Cancel).

Name: Enter the variable name. Variable name format: 1) Start with a letter + number
2) Pure letters. Chinese characters or pure numbers are not allowed. It cannot be a keyword and cannot be the same as an existing variable.

Type: bool, int, Double, string

type	storage	scope	Accuracy
Bool	1bit	False/True	
Int	1word	-32768-32767	
	2word	-2147483648-2147483647	
Double	2word	-/+3.4e38	Display 4 decimal places, up to 7 significant digits
	4word	-/+1.7e308	Display 4 decimal places, up to 16 decimal places

String	1 byte		
	1 letter		

For example, a 2-word double variable is displayed as 12345.6789, and the last 89 is inaccurate.

Value: The initial value of the variable. Fill in the appropriate value according to the type of the variable.

Use Range: False—Do not set the value range of the variable, and the maximum and minimum values below cannot be entered; TRUE: Set the value range of the variable, and the maximum and minimum values can be entered.

Add to parameter list: false – this variable is not visible in the user interface

True—This variable is visible in the user interface

5. Select the Modbus tab

Base Modbus BOOL 型变量时

Station Type: Client

Station ID:

Reg Type: Input

Reg Address:

Bit Address:

Reg Length: 1

添加
取消

Base Modbus 其他型变量时

Station Type: Client

Station ID:

Reg Type: Input

Reg Address:

Reg Length: 1

添加
取消

Station Type: Master/Slave selection

Station ID: Visible when selecting the master station. Fill in the slave station

address

Reg Type: Input/Output Selection

Reg Address: Fill in the register address

Bit Address: Visible when the variable is of bool type, fill in the bit address (range: 0-15). For example: when setting a bool type variable test_bool, Reg Address = 2,

Bit Address = 10, the state of the 10th bit of register 2 is the state of test_bool.

Reg Length: This setting is available when the variable is of a type other than bool.

Select/fill in the number of registers.

The int type variable currently supports 16-bit int and 32-bit int, and you can choose 1 or 2 registers.

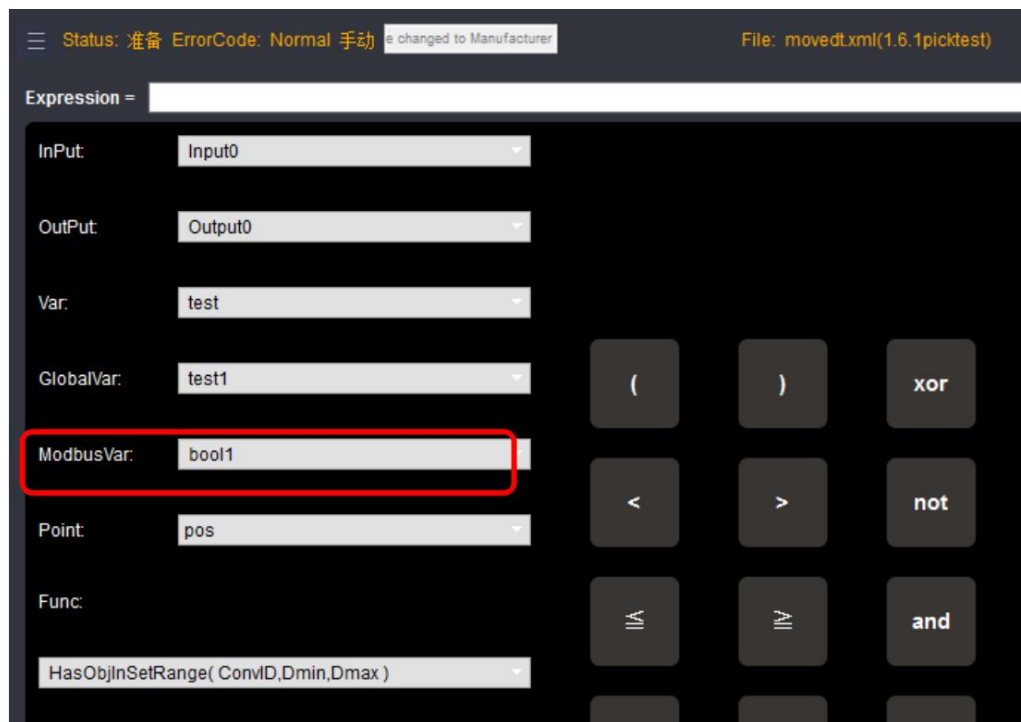
Double type variables currently support 32-bit double and 64-bit double, and can choose 2 or 4 registers.

The length of a string variable needs to be defined according to the length of the string to be sent or received, with one letter occupying one byte.

3.5 Modbus RTU variable usage

Modbus variables are used in the same way as general local variables and global variables. After Modbus communication is established, Modbus variables are constantly refreshed. You can see the current value being refreshed in the variable interface or debug interface, regardless of whether there is a program running.

In the expression editor you can see the Modbus variable selection list



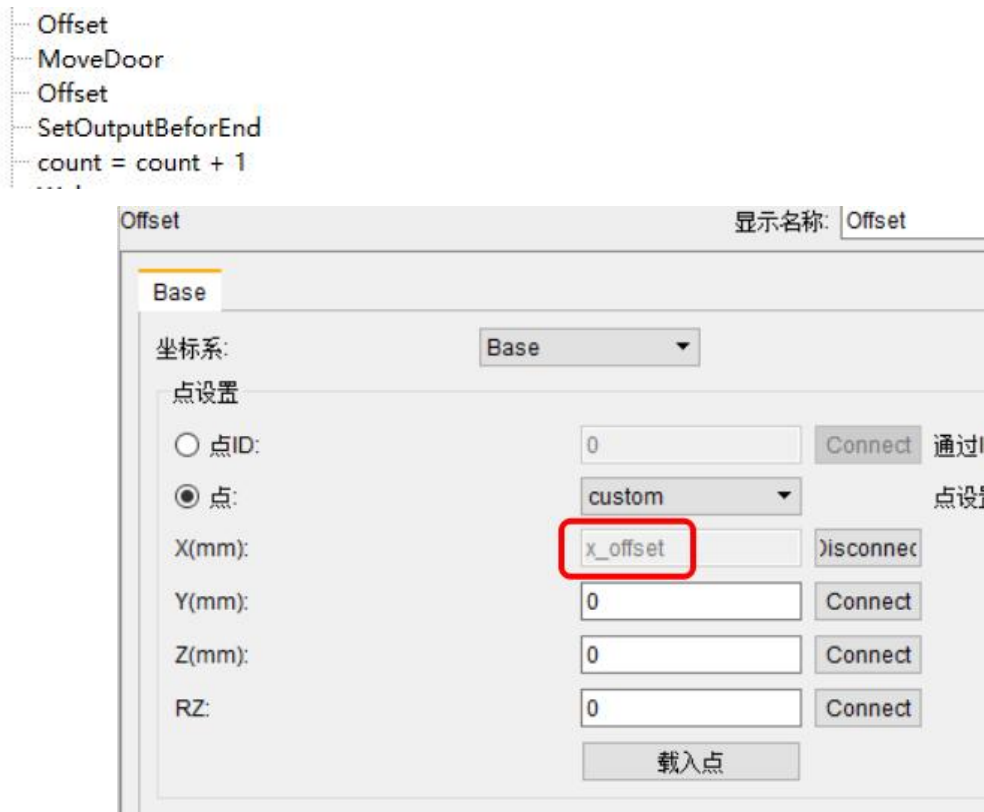
For example: the robot is the master station, the PLC is the slave station, create a modbus input double variable X-offset, this variable is the offset of the object along the x direction calculated by the PLC, the unit is mm. Create a modbus output int variable count to tell the PLC how many objects the robot has picked up now. MoveDoor command is used to grab the position. Before that, the position of movedoor is offset by offset command. The offset can be associated with modbus input variable x_offset. After grabbing, modbus variable count is incremented by 1.

The current status of the two variables in the Modbus variable table are as follows, and are constantly refreshed.

x_offset	double	10.3400	Client	1	In	101	0	2
count	int	3	Client	1	Out	112	0	1

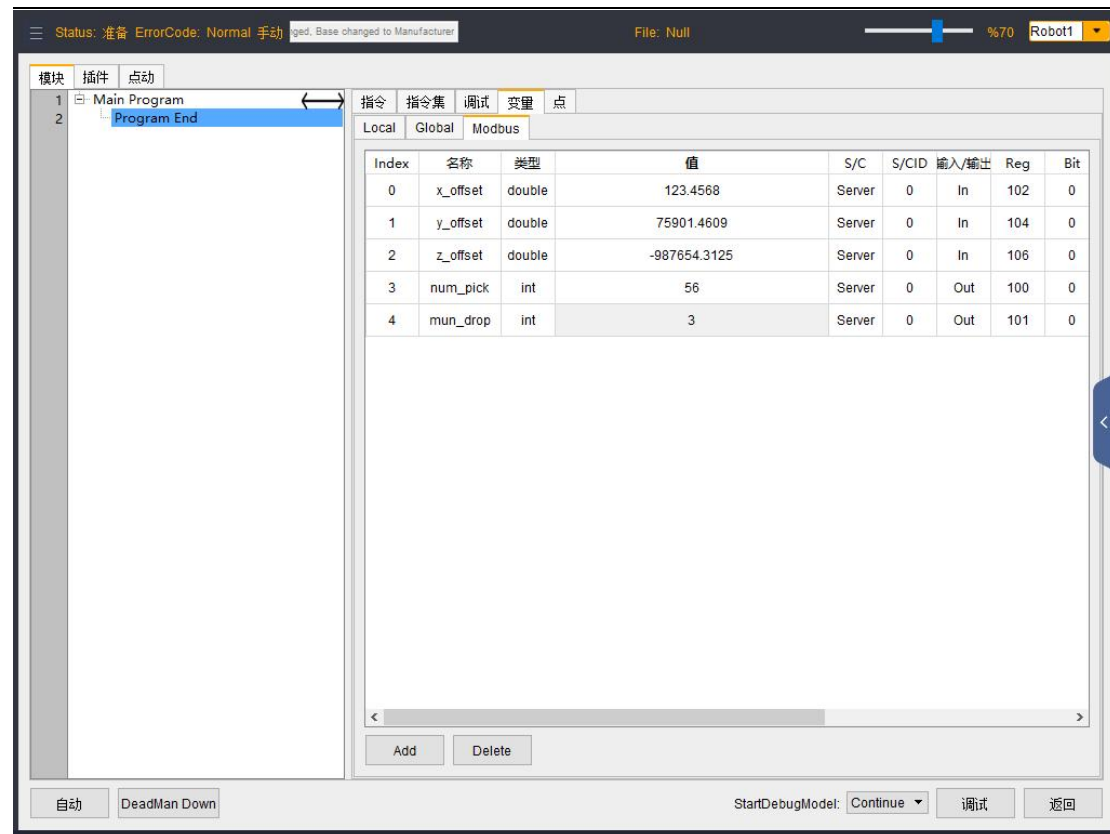
The variables are used in the program as follows. In the current state, when the robot executes movedoor, it is offset 10.34mm in the X direction from the original

position and tells the PLC that it has grabbed 3 objects.



As long as the modbus communication is established, the robot, as the master, automatically reads the plc's 101 register and puts the read value into the variable x_offset; it automatically writes the value of the variable count into the plc's 112 register. These operations are automatically completed by the robot, and the user does not have to worry about them. The user only needs to correctly set the modbus function and use the modbus variables.

RP-KZ-01-004-C



4. Modbus Control Application

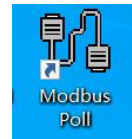
4.1 Introduction to the read working status function

Implement Modbus query of manual and automatic status and other functions. In this case, the status is queried through Modbus software. The specific operation is based on the actual situation.

4.1.1 Modbus Poll software settings

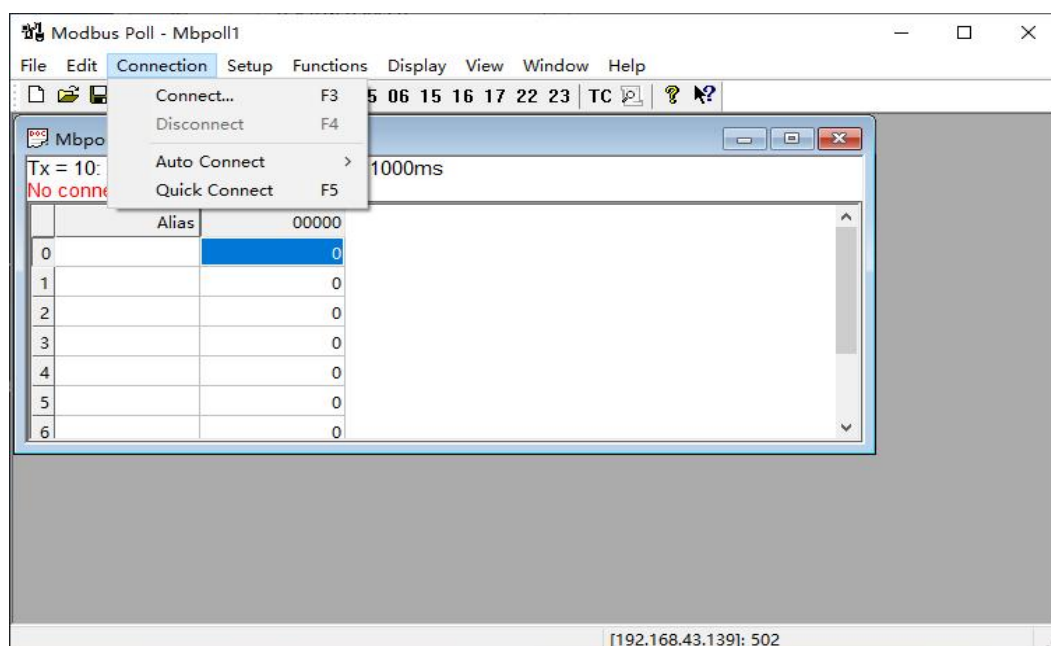
(1) Open YiKingSoft software to set up Modbus.

(2) Open Modbus and connect (Modbus is used as the master station for illustration).

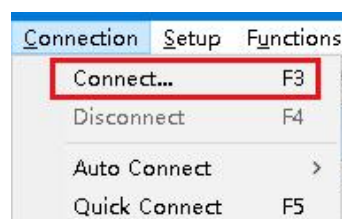


(3) Open the Modbus Poll software and click to enter the Modbus setting interface.

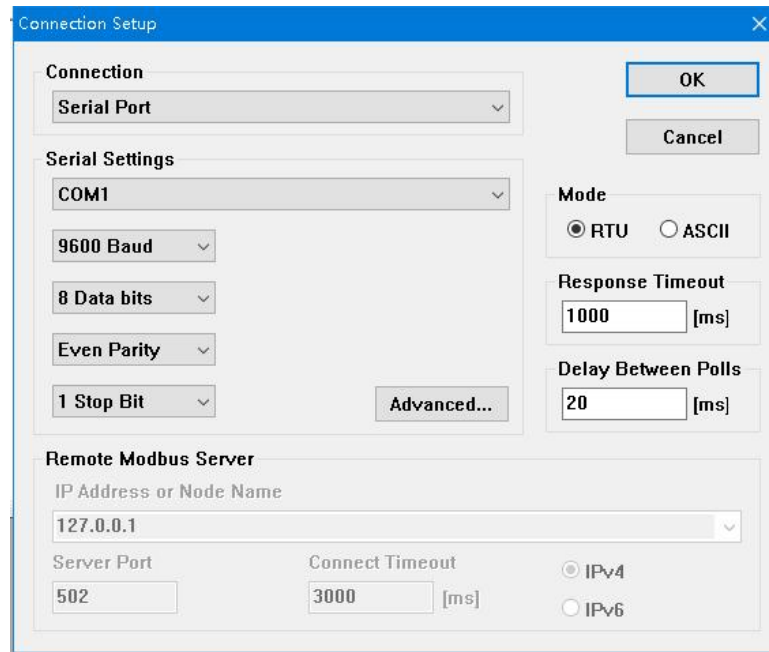
(4) Enter the Modbus Poll software setting interface and click the "Connection" tab to set up.



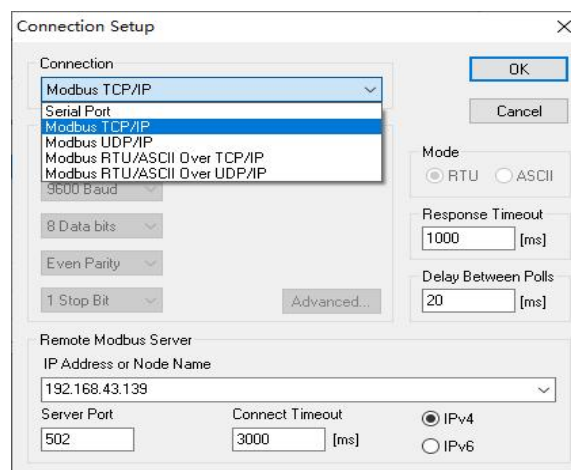
(5) Click the "Connection" drop-down list and select Connect... (F3) to enter the Connection settings interface.



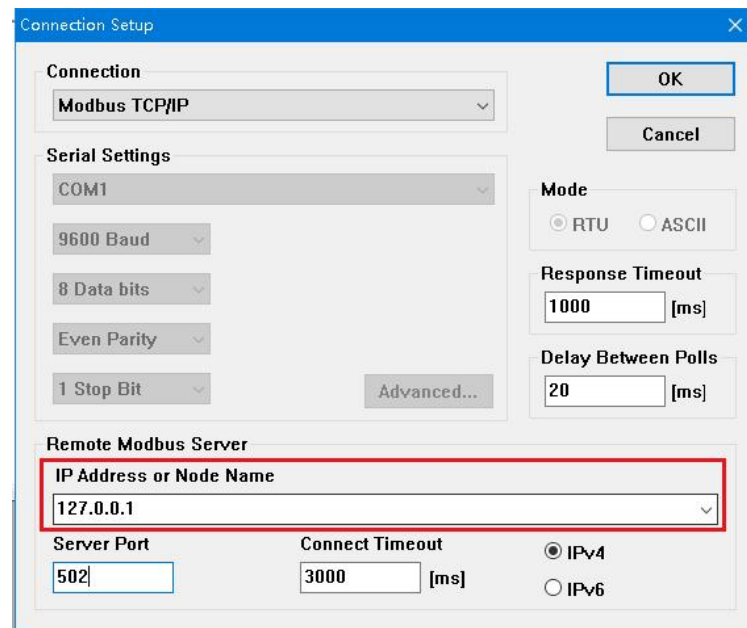
(6) Enter the "Connection Setup" interface.



(7) Select Connection and select Modbus TCP/IP in the drop-down window.

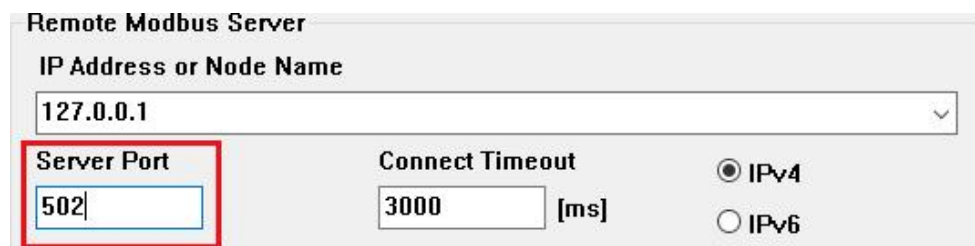


(8) Select “IP Address or Node Name” and enter the IP address of the connected slave.



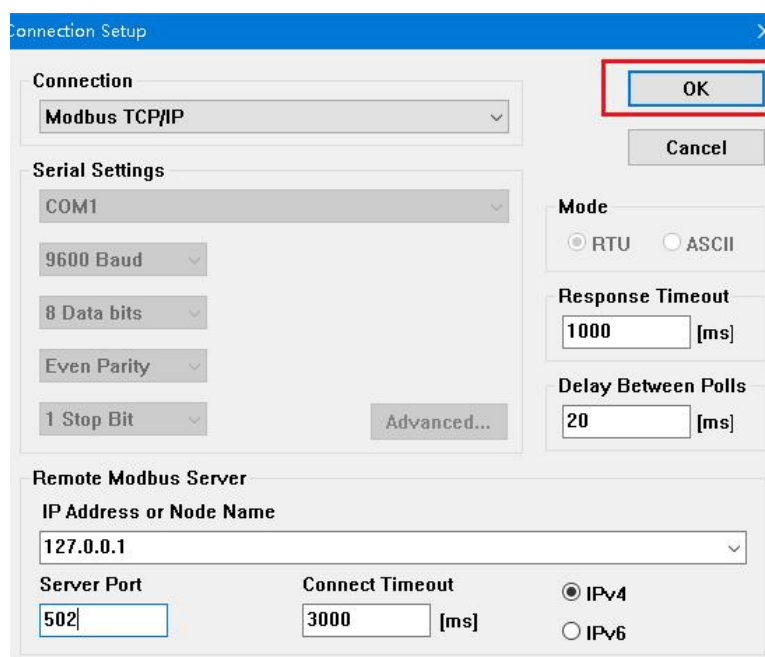
The image shows a 'Connection Setup' dialog box. It has several sections: 'Connection' with a dropdown set to 'Modbus TCP/IP'; 'Serial Settings' with dropdowns for 'COM1', '9600 Baud', '8 Data bits', 'Even Parity', and '1 Stop Bit', plus an 'Advanced...' button; 'Mode' with radio buttons for 'RTU' (selected) and 'ASCII'; 'Response Timeout' set to '1000 [ms]'; 'Delay Between Polls' set to '20 [ms]'; and 'Remote Modbus Server' with a dropdown for 'IP Address or Node Name' set to '127.0.0.1', a 'Server Port' field set to '502', a 'Connect Timeout' field set to '3000 [ms]', and radio buttons for 'IPv4' (selected) and 'IPv6'. 'OK' and 'Cancel' buttons are at the top right.

(9) Select “Server Port” and enter the port number 502.



This is a close-up of the 'Remote Modbus Server' section of the dialog. It shows the 'IP Address or Node Name' dropdown set to '127.0.0.1'. Below it, the 'Server Port' text box is highlighted with a red rectangle and contains the value '502'. To the right, the 'Connect Timeout' is '3000 [ms]' and the 'IPv4' radio button is selected.

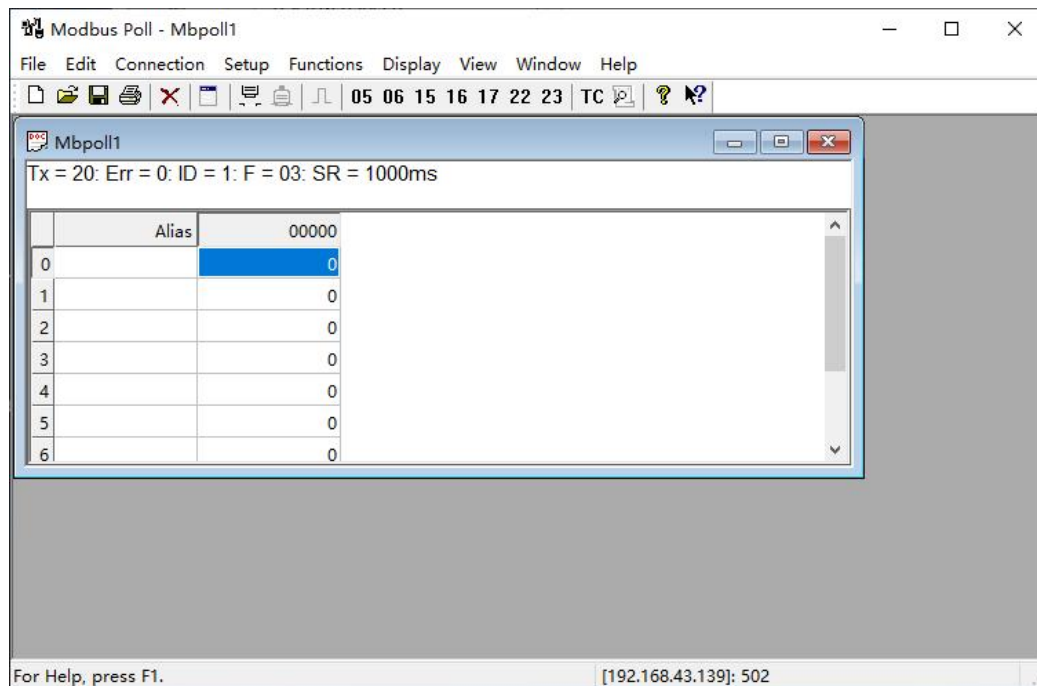
(10) Click the "OK" button to connect.



This image shows the full 'Connection Setup' dialog box again, identical to the first one. However, the 'OK' button in the top right corner is highlighted with a red rectangle, indicating the next step in the process.

(11) After clicking the "OK" button, it will automatically return to the previous

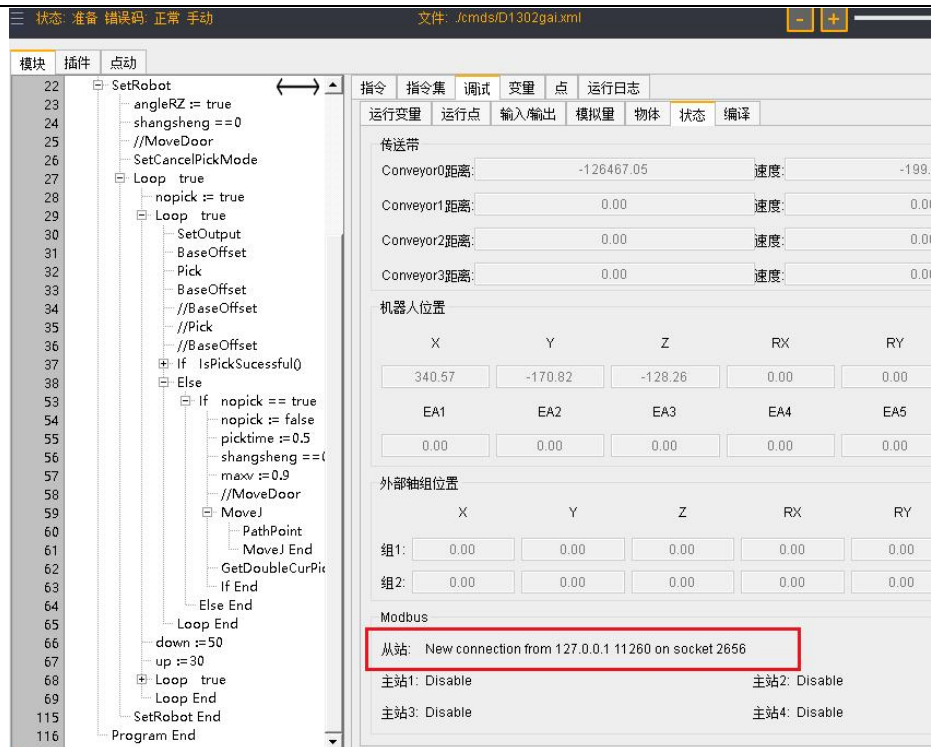
connection interface.



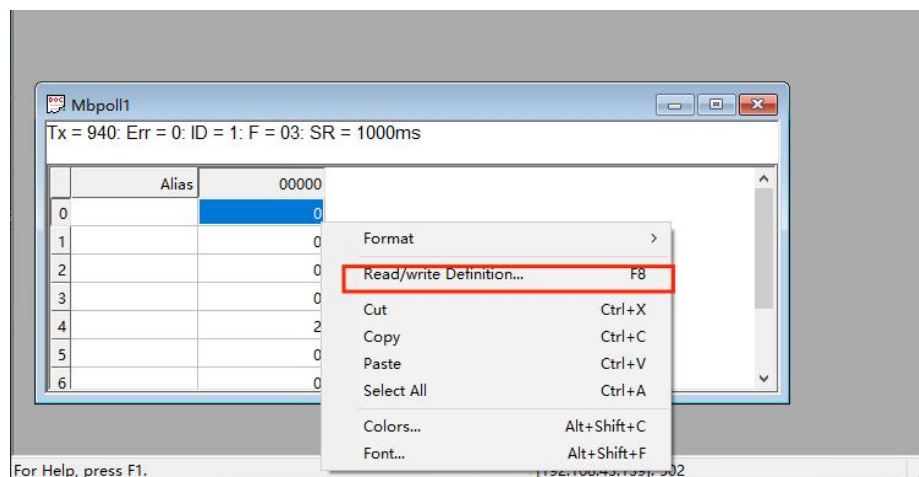
4.1.2 Modbus communication with YiKingSoft software

(1) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station according to the above "2" and "3". This time, Modbus Poll is the master station and the robot software is the slave station.

RP-KZ-01-004-C



(2) To modify the Modbus read/write function, right-click "Read/write Definition... (F8)" to enter the read/write definition modification interface.



(3) Enter the read/write definition modification interface, select the "Function" drop-down option and select "04 Read Input Registers (3x)", read the input register of the lower computer, and click the "OK" button to complete the setting.

Read/Write Definition

Slave ID: 1

Function: 03 Read Holding Registers (4x) (selected)

Address: 01 Read Coils (0x)
02 Read Discrete Inputs (1x)
03 Read Holding Registers (4x)
04 Read Input Registers (3x) (selected)

Quantity: 40011 -> 10

Scan Rate: 05 Write Single Coil
06 Write Single Register
15 Write Multiple Coils
16 Write Multiple Registers

Disable: ☐ Read/Write Disabled
☐ Disable on error

Buttons: OK, Cancel, Apply, Read/Write Once

View: Rows (10 selected, 20, 50, 100, Fit to Quantity)
☐ Hide Alias Columns
☐ PLC Addresses (Base 1)
☐ Address in Cell
☐ Enron/Daniel Mode

(4) Return to the interface for reading the input register address bit, the manual and automatic address bit is 9.

(0-manual, 1-automatic , 2-remote teaching, 3-remote debugging, 4-remote operation)

Modbus Poll - Mbpoll1

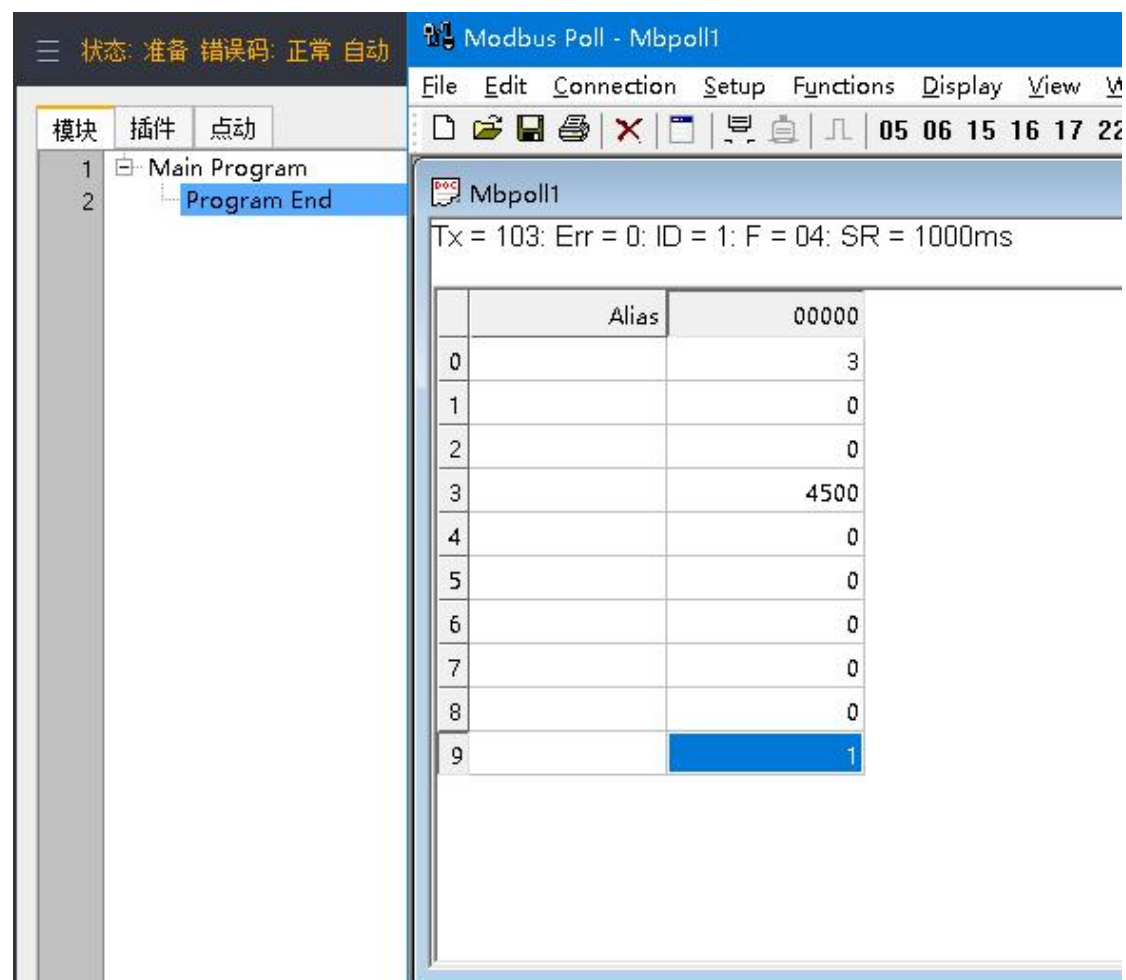
File Edit Connection Setup Functions Display View Window Help

05 06 15 16 17 22 23 TC ? ?

Mbpoll1

Tx = 42: Err = 0: ID = 1: F = 04: SR = 1000ms

	Alias	00000
0		3
1		0
2		0
3		4500
4		0
5		0
6		0
7		0
8		0
9		0



4.2 Error reset function introduction

Implement the Modbus error reset state function. This case describes the error reset function through Modbus software. Specific operations are based on actual conditions.

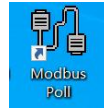
Modbus Error Reset: 0—Default

1—Error reset, the command enters the "reset stop" process

2—Error reset, the command enters the "command recovery" process

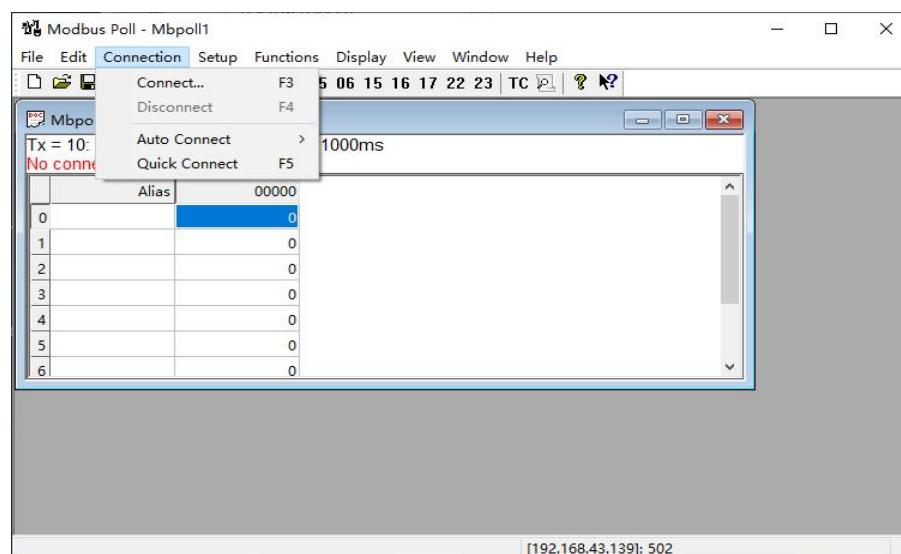
4.2.1 Modbus Poll software communication connection

(1) Perform Modbus Poll software connection communication. The software connection communication for querying the manual and automatic status description is the same.



(2) Open Modbus Poll and click to enter the Modbus setting interface.

(3) Enter the Modbus settings interface and select Connection settings.



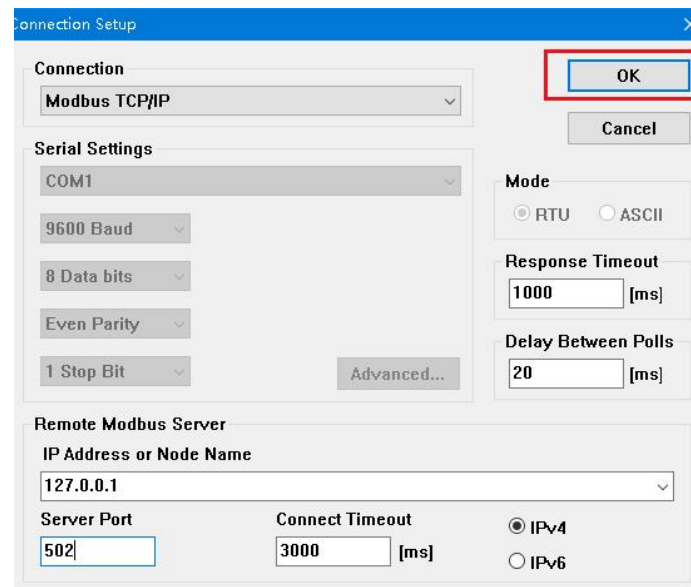
(4) Select Connect... (F3) to enter the Connection interface.

(5) Select Modbus TCP/IP in the Connection drop-down window.

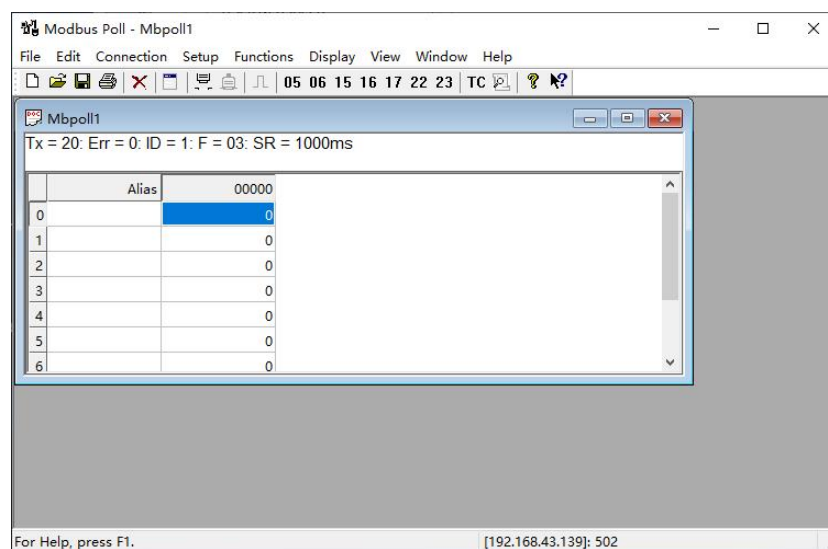
(6) IP Address or Node Name: Enter the IP address of the connected slave node.

(7) Enter 502 as the Server Port number.

(8) Click OK to connect. (Settings are shown in the figure)



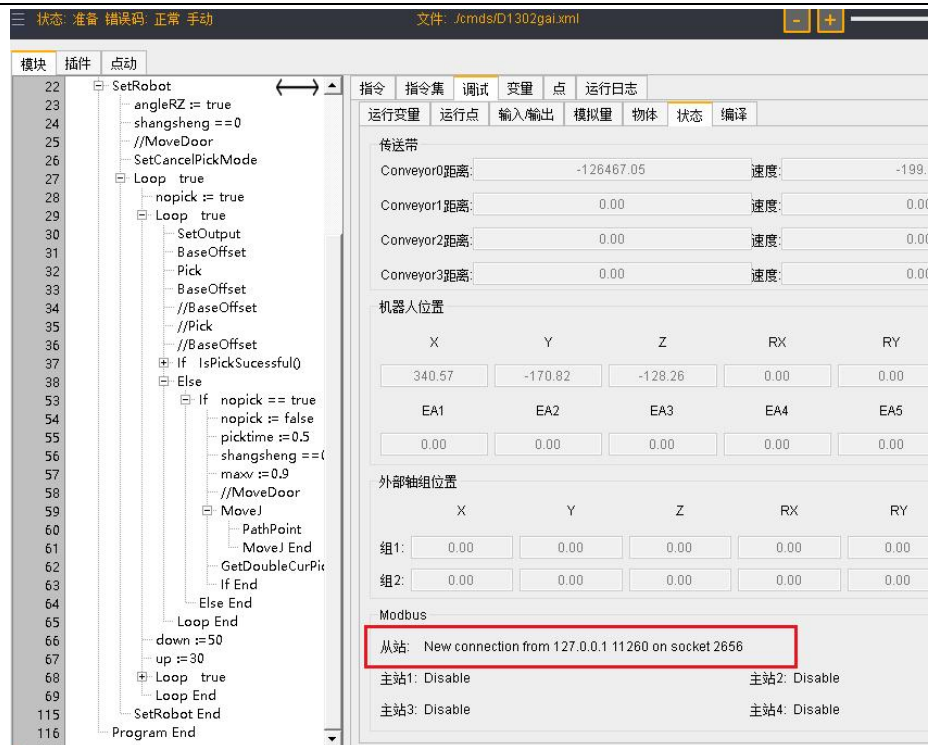
(9) After clicking the "OK" button, it will automatically return to the previous connection interface.



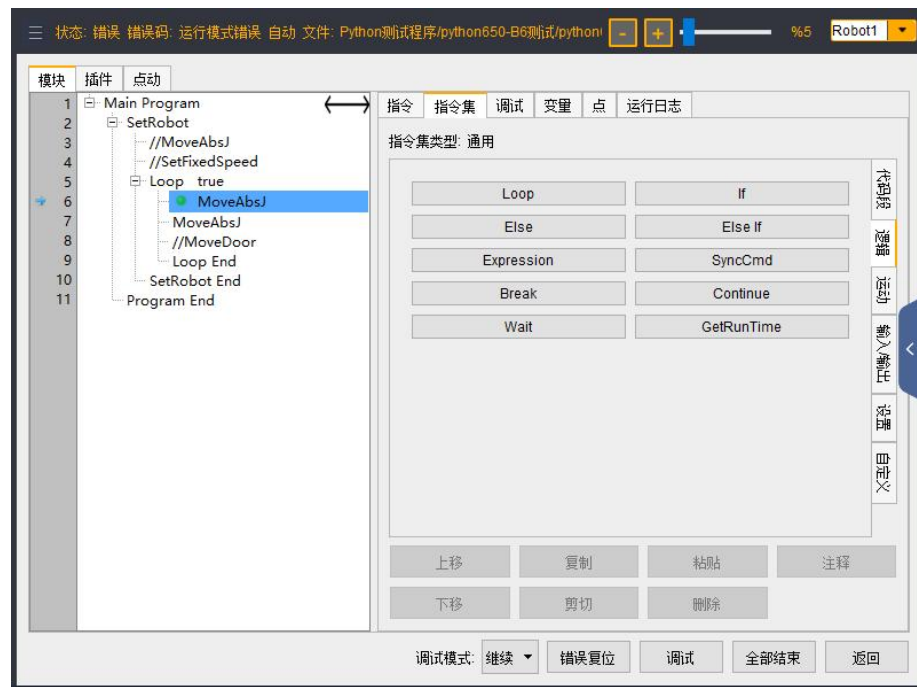
4.2.2 Modbus communication with YiKingSoft software

(1) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station according to the above "2" and "3". This time, Modbus Poll is the master station and the robot software is the slave station.

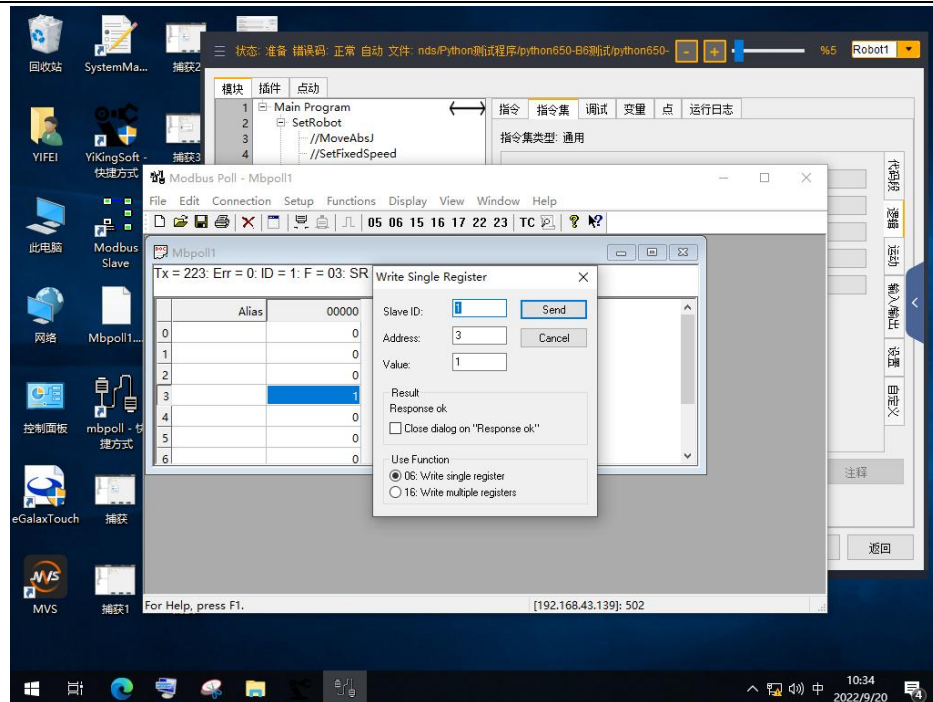
RP-KZ-01-004-C



(2) When an error instruction appears in YiKingSoft software (as shown in the figure).



(3) The Value of Modbus address bit 3 can be changed to 1 or 2 to reset the error.



(4) Software error reset successful.

4.3 Modbus emergency stop reset function introduction

Implement the Modbus emergency stop reset state function. This case describes the emergency stop reset function through Modbus software. Specific operations are based on actual conditions.

Modbus Error Reset: 0—Default

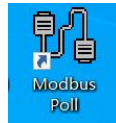
1—Emergency stop reset, the command enters the "reset stop" process

2—Emergency stop reset, the command enters the "command recovery" process

4.3.1 Modbus Poll software communication connection

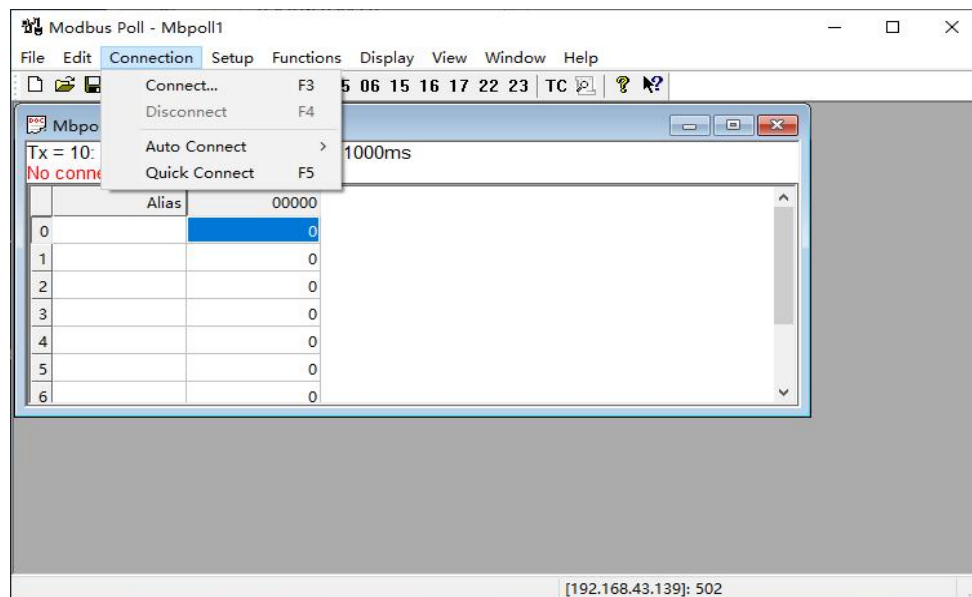
(1) Perform Modbus Poll software connection communication. The software connection

communication for querying the manual and automatic status description is the same.



(2) Open Modbus Poll and click to enter the Modbus setting interface.

(3) Enter the Modbus settings interface and select Connection settings.



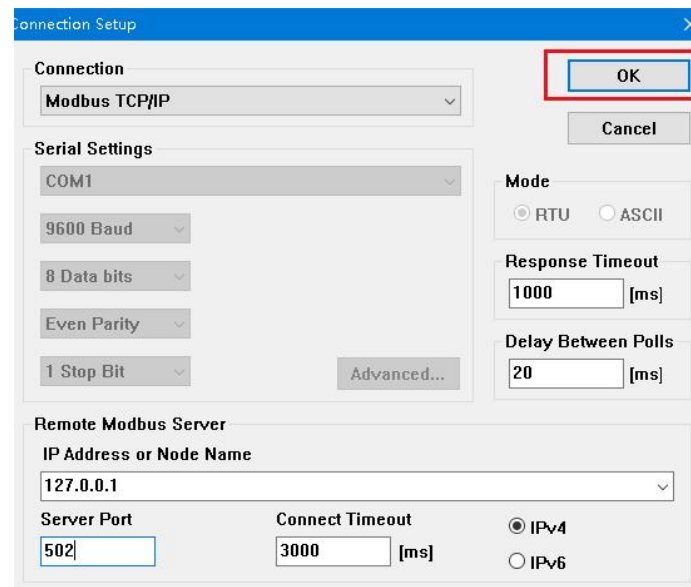
(4) Select Connect... (F3) to enter the Connection interface.

(5) Select Modbus TCP/IP in the Connection drop-down window.

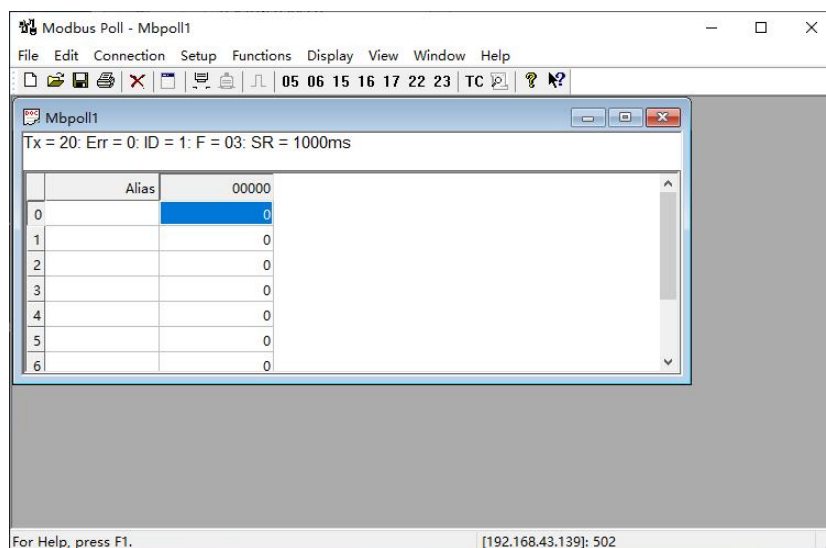
(6) IP Address or Node Name: Enter the IP address of the connected slave node.

(7) Enter 502 as the Server Port number.

(8) Click OK to connect. (Settings are shown in the figure)



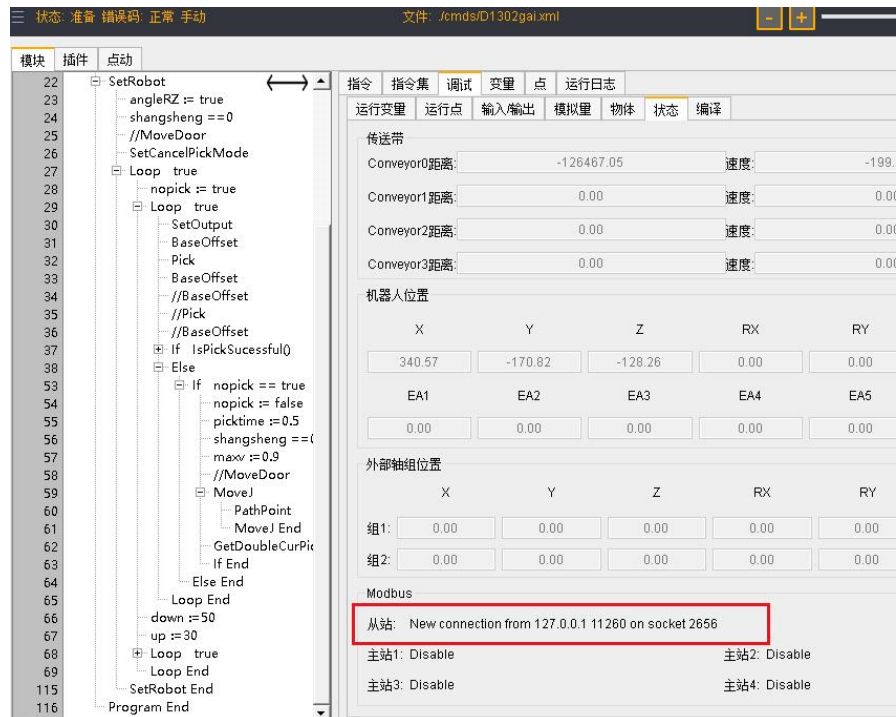
(9) After clicking the "OK" button, it will automatically return to the previous connection interface.



4.3.2 Modbus and YiKingSoft software connection

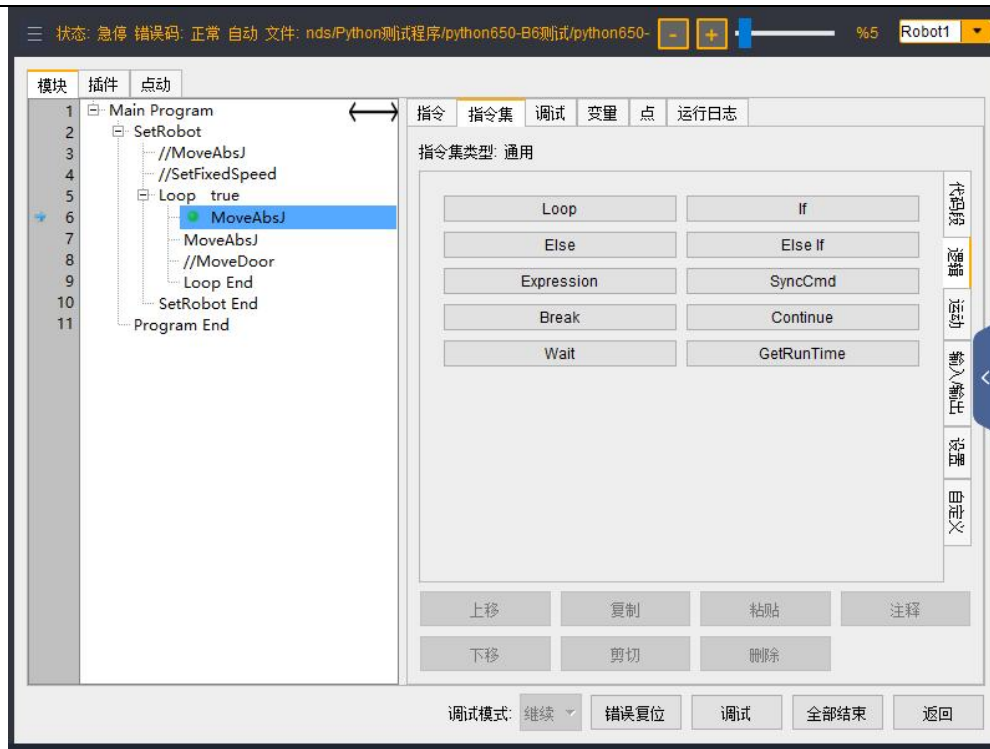
(1) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station

according to the above "2" and "3". This time, Modbus Poll is the master station and the robot software is the slave station.



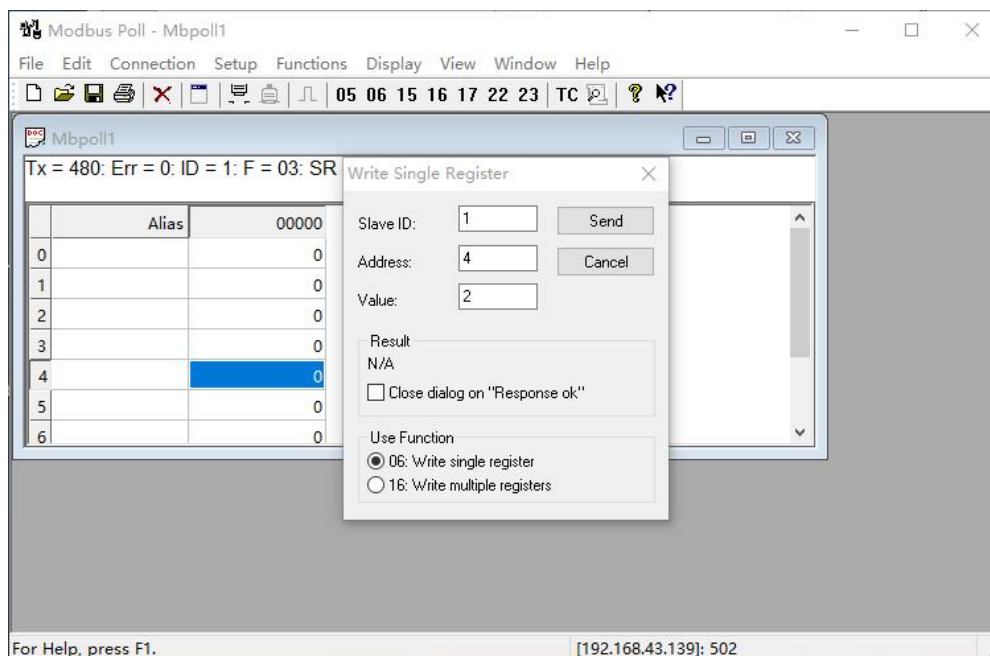
(2) When the emergency stop button is pressed, the software will enter the emergency stop status.

RP-KZ-01-004-C

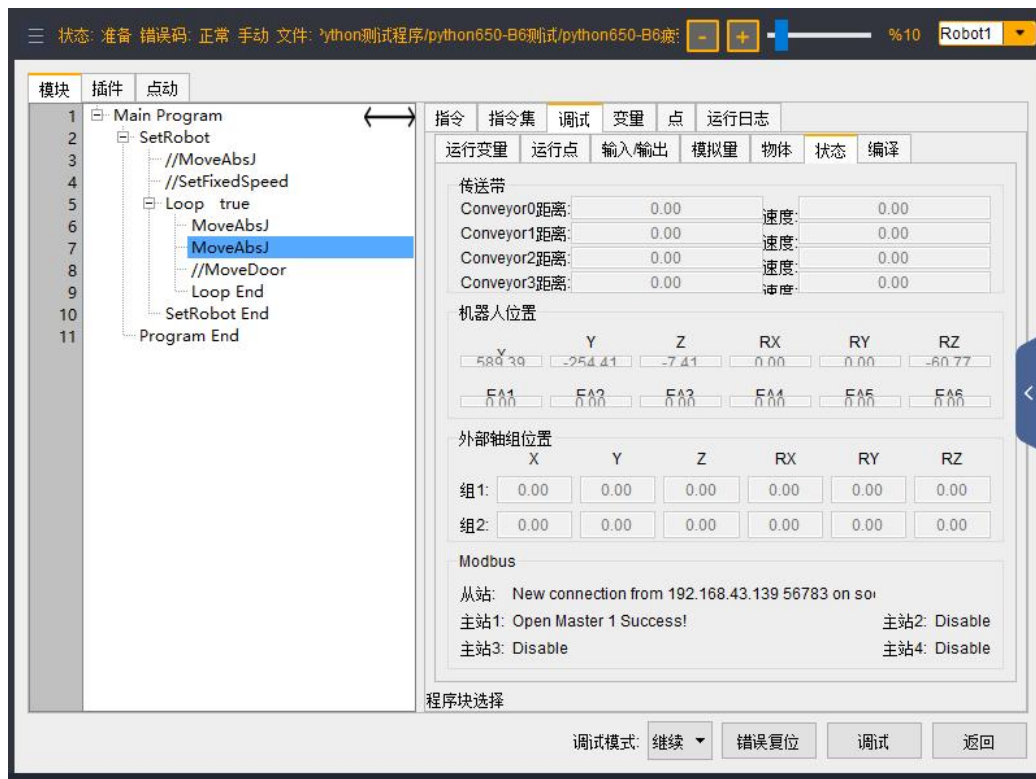


(3) Rotate and pull out the emergency stop button on the electric cabinet.

(4) Change the value of Modbus address bit 4 to 1 or 2 to reset the emergency stop alarm.



(5) The software status is restored to the ready state (reset stop) or running state (instruction recovery).



5. Modbus remote jog

Since the software monitors the edge changes in the register, if you want to execute the same function continuously, you need to assign a code value with no function (the default value 0 is recommended) to the corresponding address bit of the register after executing it once, and then execute the function again.

In Modbus remote teaching mode, the jog operation is realized by the master station reading and writing the corresponding address bits of HoldRegisters . Among them:

Address position 90 writes the remote jog function code.

Address bit 91 is the moving distance of the PTP operation.

Address 92 is the coordinate system for the jog operation.

Please refer to the appendix for details.

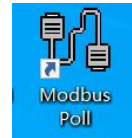
5.1 Introduction to the function of reading robot speed

Implement the Modbus query function for the robot's current speed . In this case, the status is queried through the Modbus software. The specific operation is based on the actual situation.

5.1.1 Modbus Poll software settings

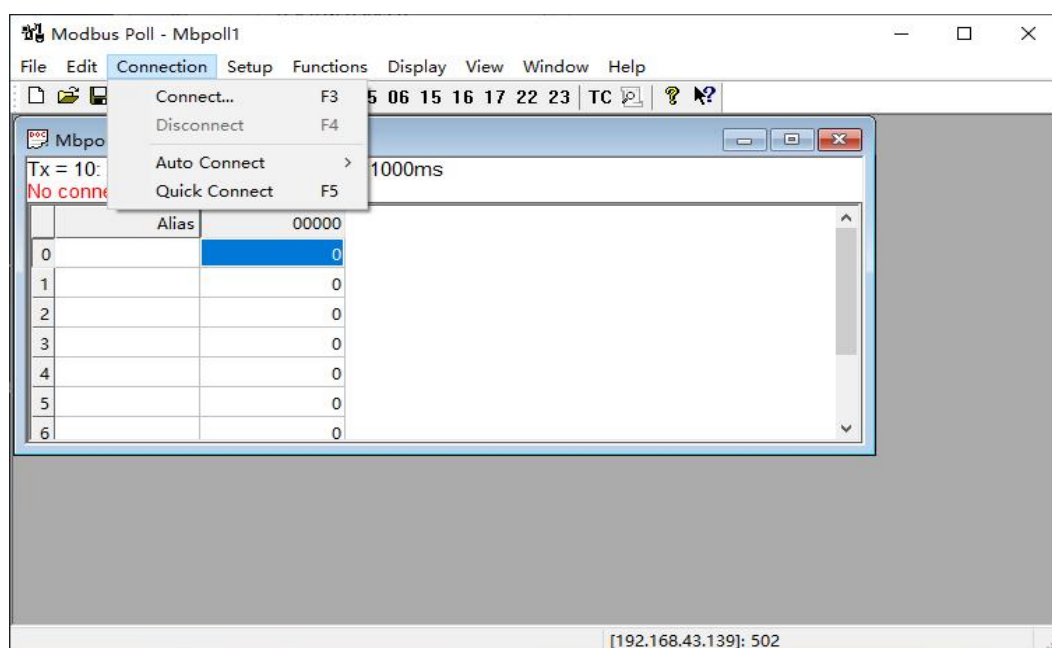
(1) Open YiKingSoft software to set up Modbus.

(2) Open Modbus and connect (Modbus is used as the master station for illustration).

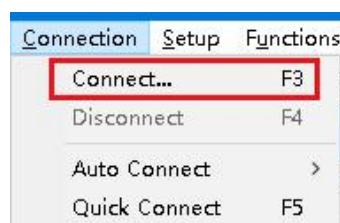


(3) Open the Modbus Poll software and click to enter the Modbus setting interface.

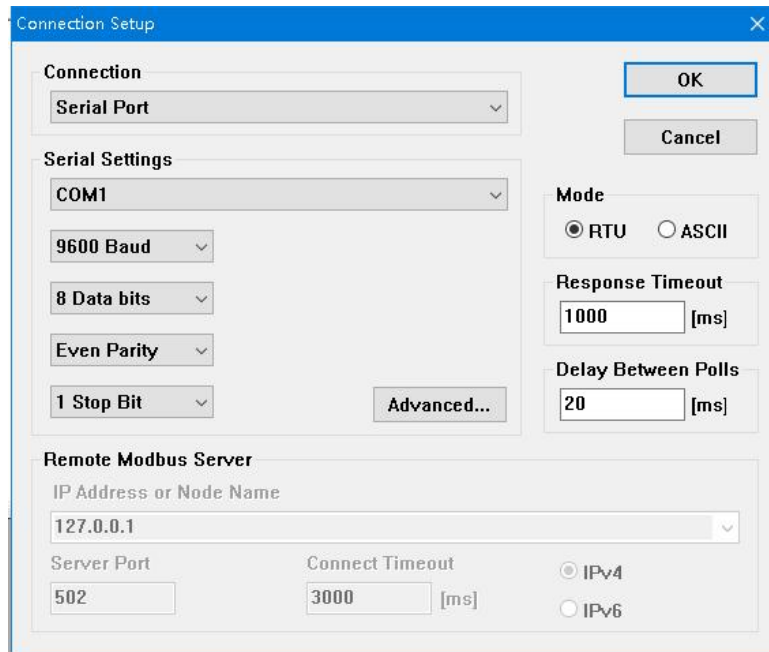
(4) Enter the Modbus Poll software setting interface and click the "Connection" tab to set up.



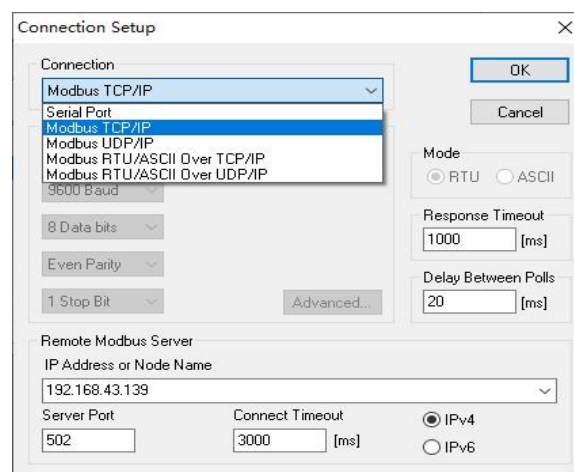
(5) Click the "Connection" drop-down list and select Connect... (F3) to enter the Connection settings interface.



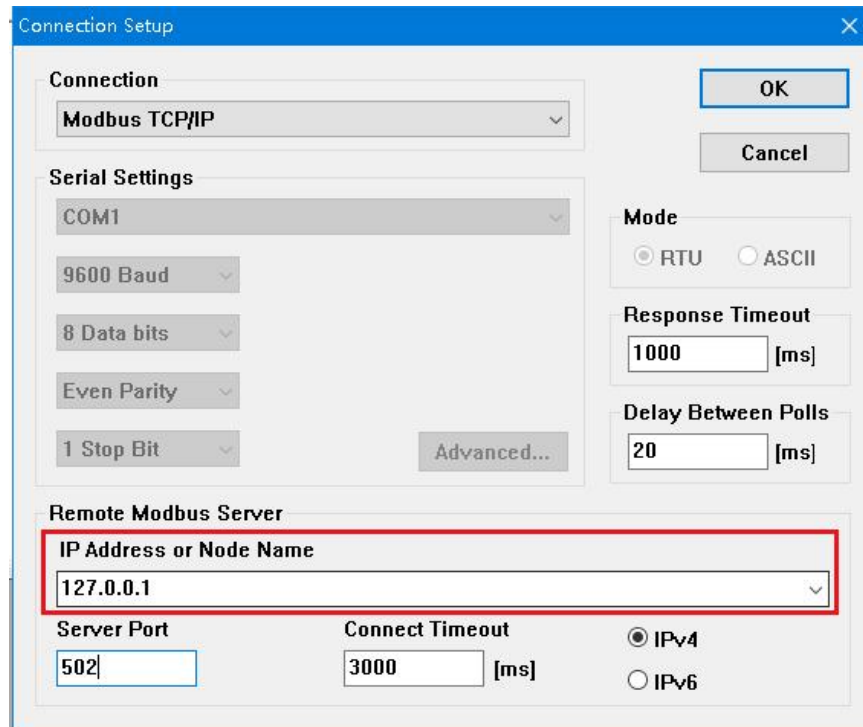
(6) Enter the "Connection Setup" interface.



(7) Select Connection and select Modbus TCP/IP in the drop-down window.

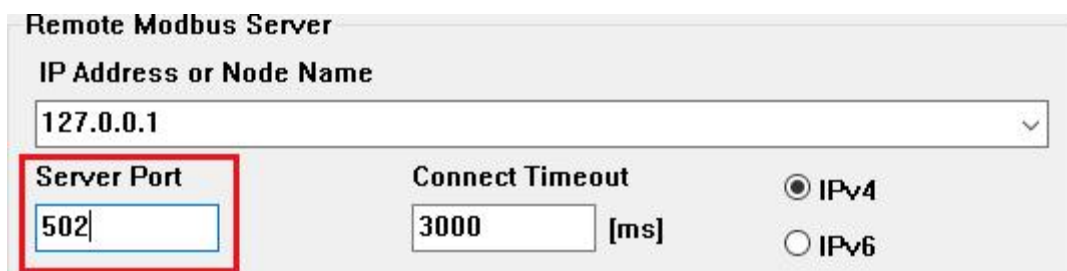


(8) Select “IP Address or Node Name” and enter the IP address of the connected slave.



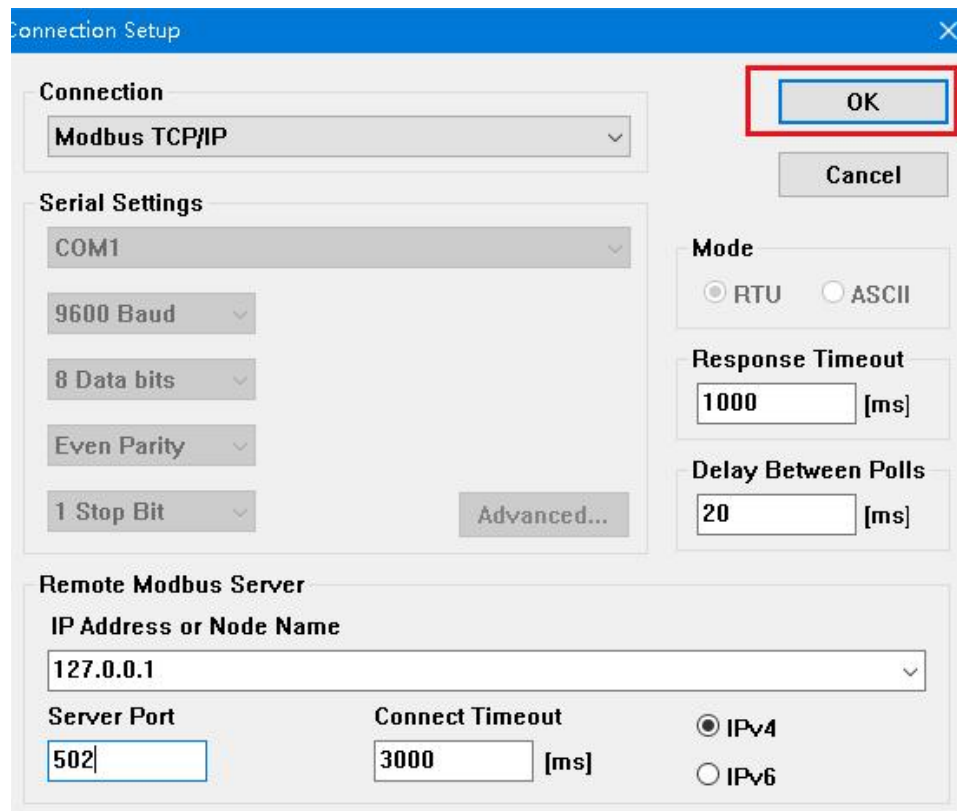
The image shows a 'Connection Setup' dialog box with a blue title bar and a close button. It contains several sections: 'Connection' with a dropdown menu set to 'Modbus TCP/IP'; 'Serial Settings' with dropdowns for 'COM1', '9600 Baud', '8 Data bits', 'Even Parity', and '1 Stop Bit', plus an 'Advanced...' button; 'Mode' with radio buttons for 'RTU' (selected) and 'ASCII'; 'Response Timeout' with a text box '1000' and '[ms]'; 'Delay Between Polls' with a text box '20' and '[ms]'; and 'Remote Modbus Server' with a dropdown for 'IP Address or Node Name' set to '127.0.0.1', a 'Server Port' text box with '502', a 'Connect Timeout' text box with '3000' and '[ms]', and radio buttons for 'IPv4' (selected) and 'IPv6'. 'OK' and 'Cancel' buttons are in the top right.

(9) Select “Server Port” and enter the port number 502.



This image is a close-up of the 'Remote Modbus Server' section from the previous dialog. It shows the 'IP Address or Node Name' dropdown set to '127.0.0.1'. Below it, the 'Server Port' text box contains '502' and is highlighted with a red rectangle. To the right, the 'Connect Timeout' text box contains '3000' and '[ms]', and the 'IPv4' radio button is selected.

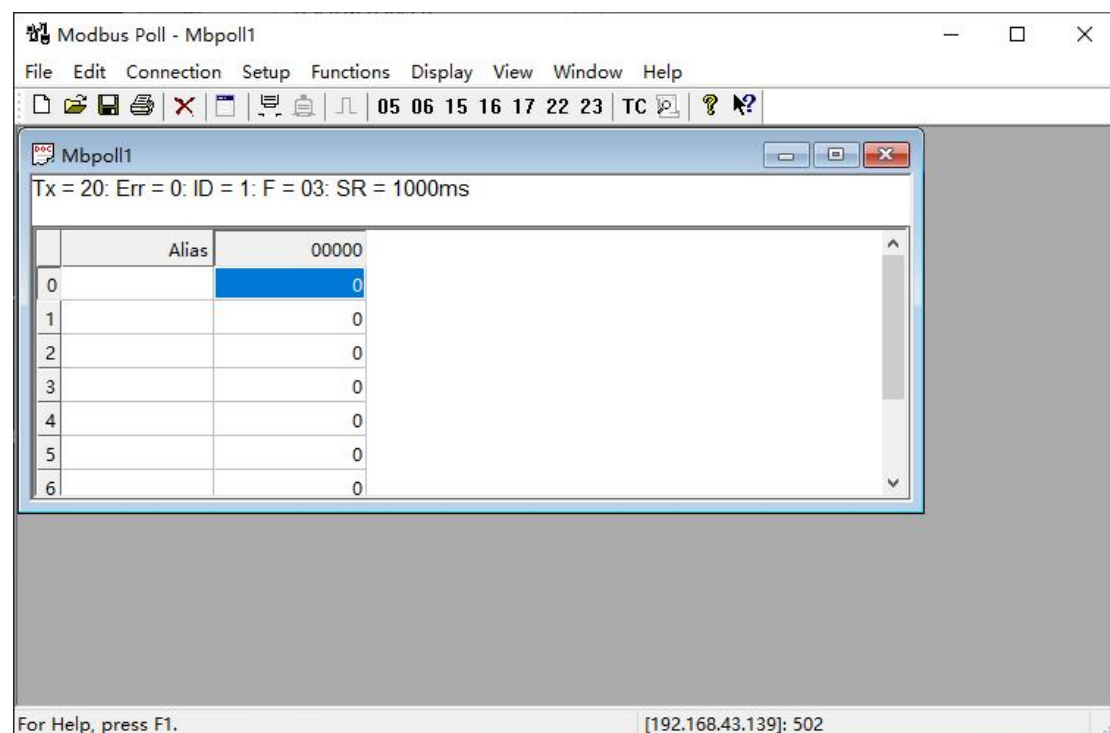
(10) Click the "OK" button to connect.



The image shows a 'Connection Setup' dialog box with the following fields and controls:

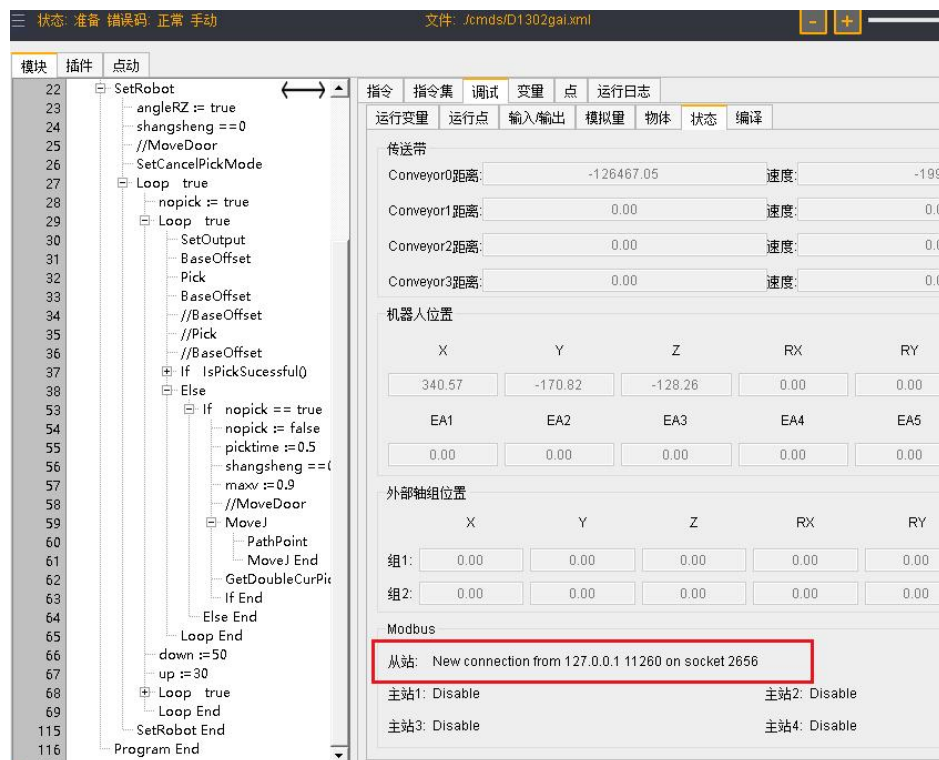
- Connection:** A dropdown menu set to 'Modbus TCP/IP'.
- Serial Settings:** A group of dropdown menus for 'COM1', '9600 Baud', '8 Data bits', 'Even Parity', and '1 Stop Bit'. An 'Advanced...' button is located to the right of these settings.
- Mode:** Radio buttons for 'RTU' (selected) and 'ASCII'.
- Response Timeout:** A text box containing '1000' followed by '[ms]'.
- Delay Between Polls:** A text box containing '20' followed by '[ms]'.
- Remote Modbus Server:** A section containing:
 - IP Address or Node Name:** A dropdown menu set to '127.0.0.1'.
 - Server Port:** A text box containing '502'.
 - Connect Timeout:** A text box containing '3000' followed by '[ms]'.
 - Protocol:** Radio buttons for 'IPv4' (selected) and 'IPv6'.
- Buttons:** 'OK' and 'Cancel' buttons are located in the top right corner. The 'OK' button is highlighted with a red rectangle.

(11) After clicking the "OK" button, it will automatically return to the previous connection interface.

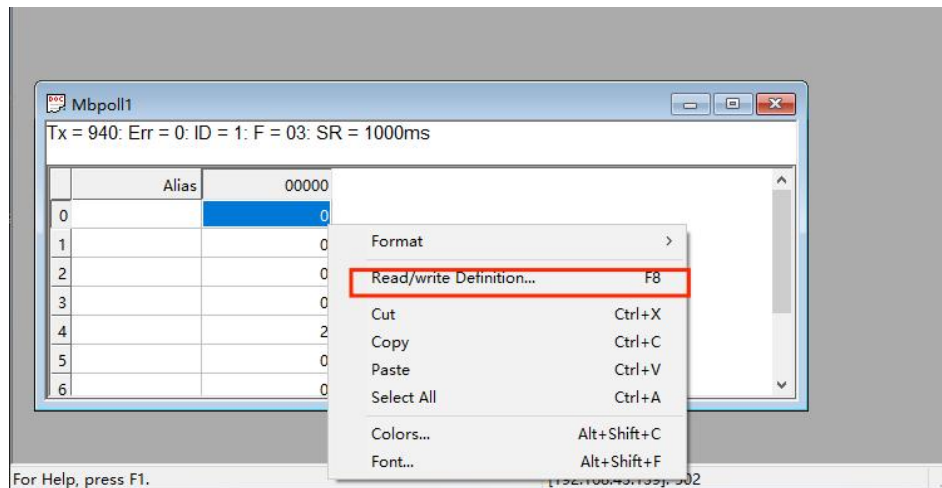


5.1.2 Modbus communication with YiKingSoft software

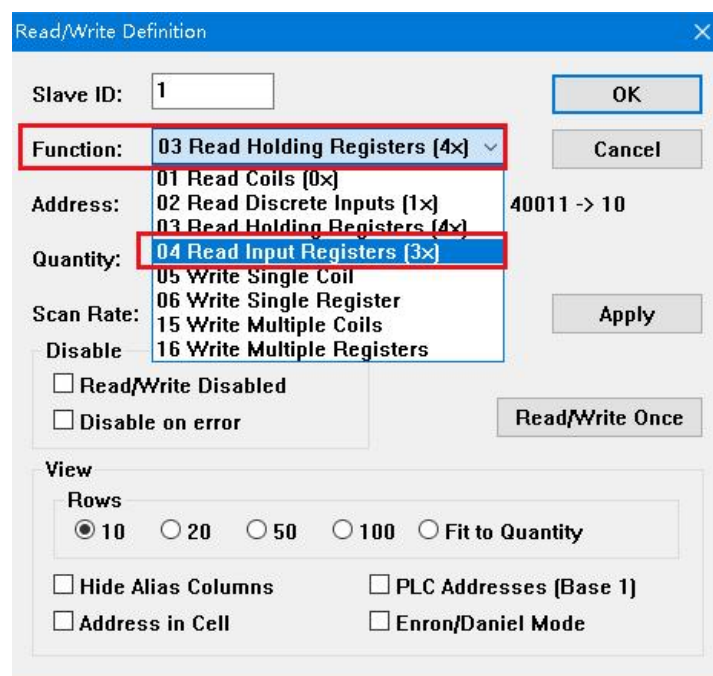
(1) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station according to the above "2" and "3". This time, Modbus Poll is the master station and the robot software is the slave station.



(2) To modify the Modbus read/write function, right-click "Read/write Definition... (F8)" to enter the read/write definition modification interface.



(3) Enter the read/write definition modification interface, select the "Function" drop-down option and select "04 Read Input Registers (3x)", read the input register of the lower computer, and set the Quantity to 20, and click the "OK" button to complete the setting.



Slave ID: OK

Function: Cancel

Address: Protocol address. E.g. 30011 -> 10

Quantity:

Scan Rate: [ms] Apply

Disable
☐ Read/Write Disabled
☐ Disable on error

Read/Write Once

View
 Rows
☒ 10 ☐ 20 ☐ 50 ☐ 100 ☐ Fit to Quantity
☐ Hide Alias Columns ☐ PLC Addresses (Base 1)
☐ Address in Cell ☐ Enron/Daniel Mode

(4) Return to the read input register address bit interface, the robot speed address bit is 10 .

Modbus Poll - [Mbpoll1]

File Edit Connection Setup Functions Display View Window Help

05 06 15 16 17 22 23 TC

Tx = 161; Err = 6; ID = 1; F = 04; SR = 1000ms

	Alias	00000	Alias	00010
0		3		43
1		0		0
2		0		0
3		0		0
4		0		0
5		-7100		0
6		0		0
7		0		0
8		0		0
9		0		0

For Help, press F1. [127.0.0.1]: 502

量 数组 点 示波器 I/O状态 运行日志

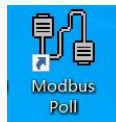
显示名称: Main Program

Robot1

5.2 Start remote teaching mode

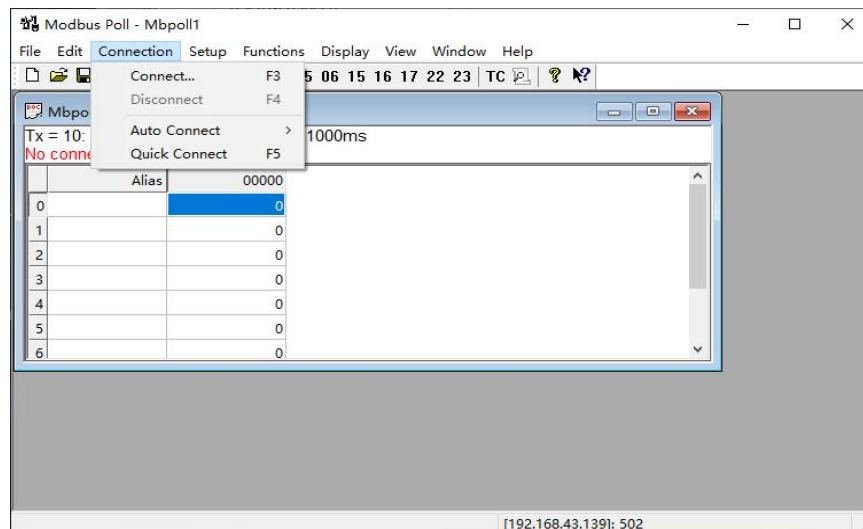
5.2.1. Modbus Poll software settings

(1) Perform Modbus Poll software connection communication. The software connection communication for querying the manual and automatic status description is the same.



(2) Open Modbus Poll and click to enter the Modbus setting interface.

(3) Enter the Modbus settings interface and select Connection settings.



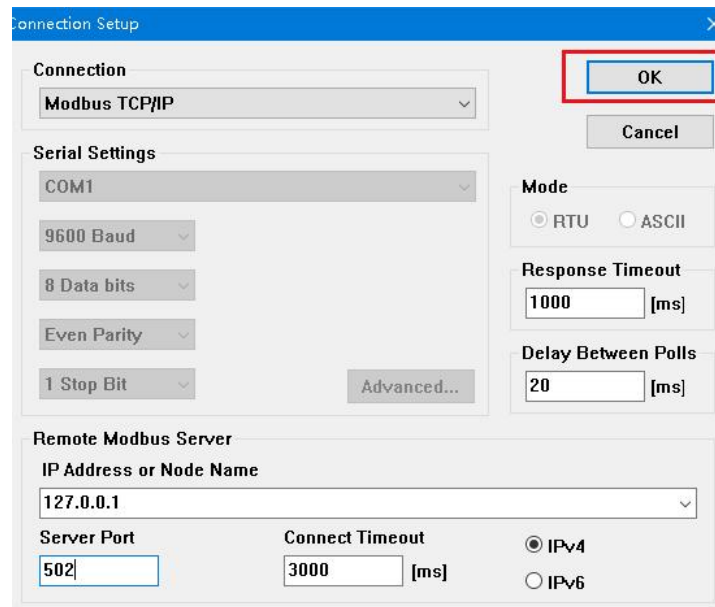
(4) Select Connect... (F3) to enter the Connection interface.

(5) Select Modbus TCP/IP in the Connection drop-down window.

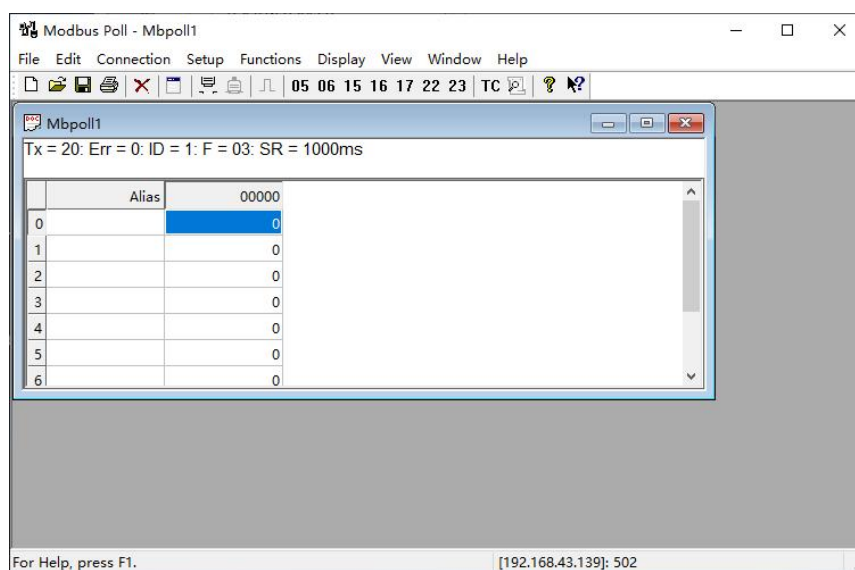
(6) IP Address or Node Name: Enter the IP address of the connected slave node.

(7) Enter 502 as the Server Port number.

(8) Click OK to connect. (Settings are shown in the figure)



(9) After clicking the "OK" button, it will automatically return to the previous connection interface.

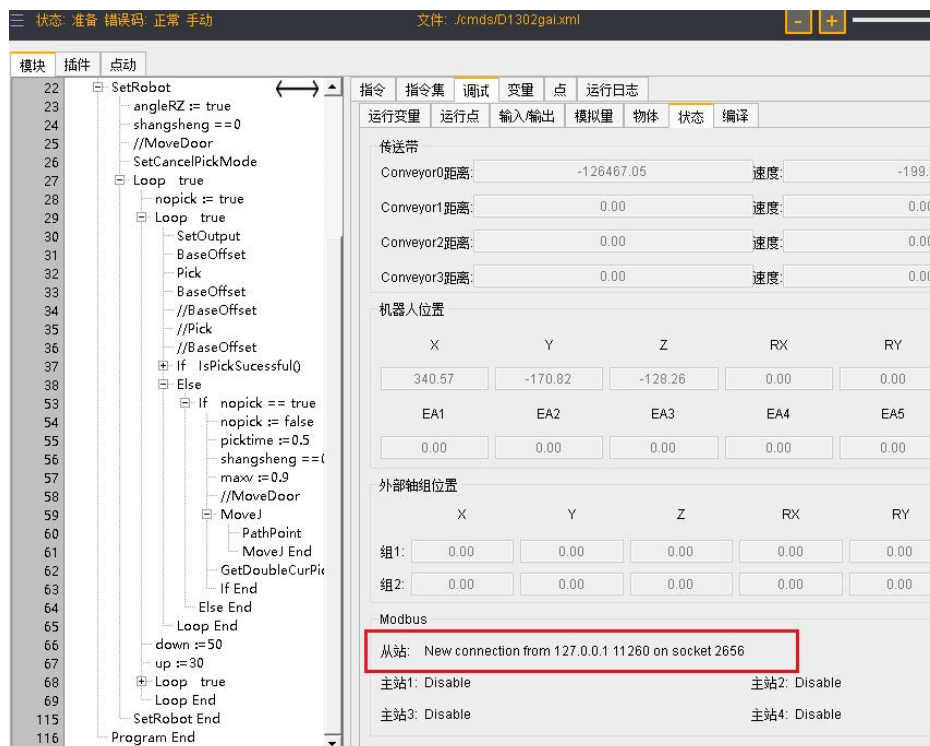


5.2.2 Modbus and YiKingSoft software connection

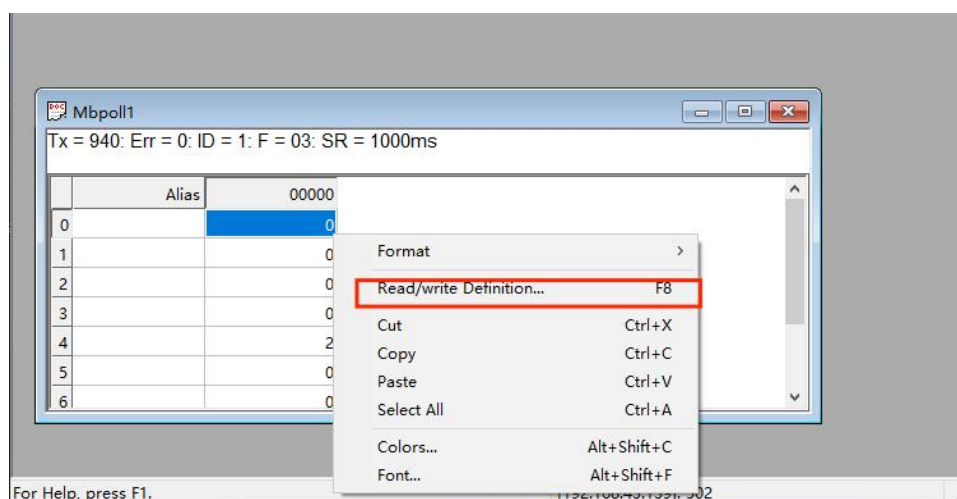
(1) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station according to the above "2" and "3". This time, Modbus Poll is the master station

RP-KZ-01-004-C

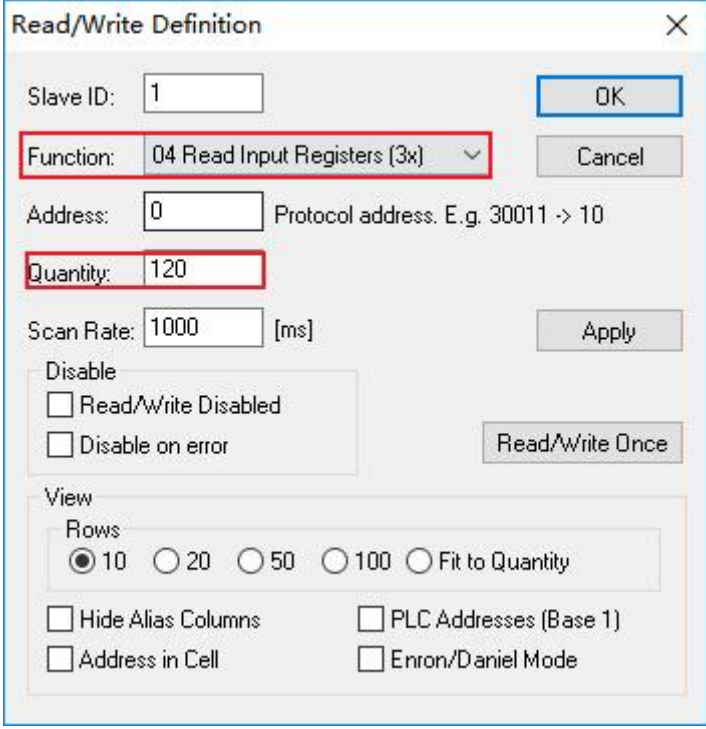
and the robot software is the slave station.



(2) To modify the Modbus read/write function, right-click "Read/write Definition... (F8)" to enter the read/write definition modification interface.

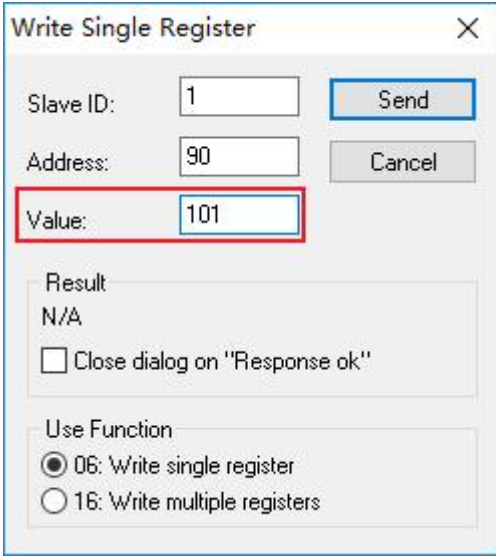


(3) Enter the read/write definition modification interface, select the "Function" drop-down option and select "0 3 Read Holding Registers (4 x)", read the input register of the lower computer, and set the Quantity to 120, and click the "OK" button to complete the setting.



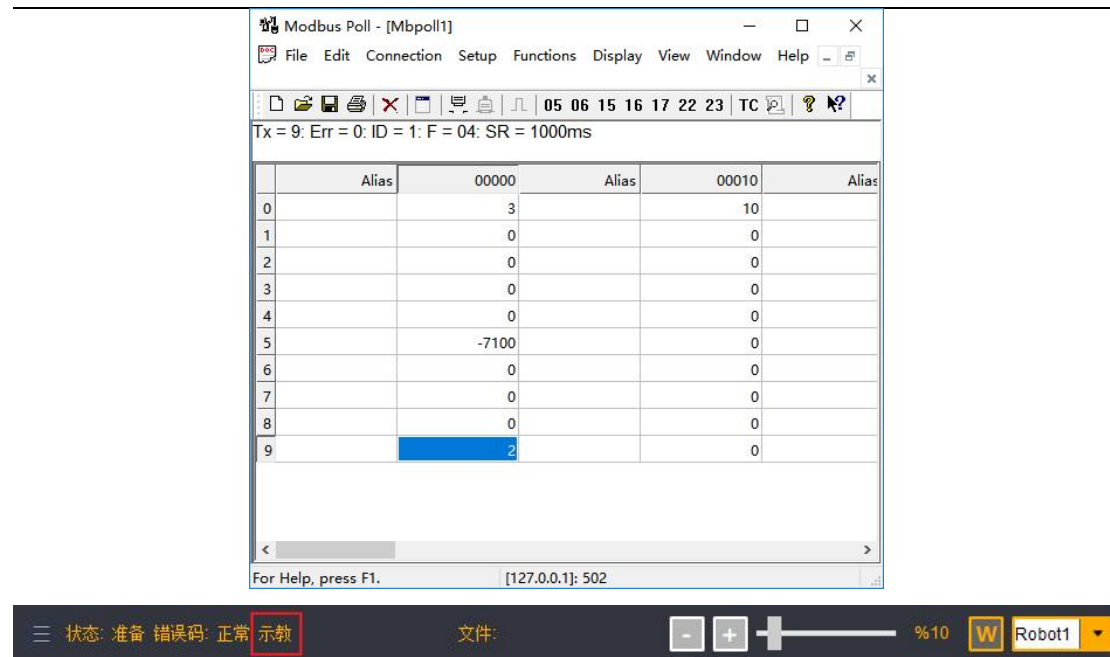
The 'Read/Write Definition' dialog box is shown with the following settings: Slave ID: 1, Function: 04 Read Input Registers (3x), Address: 0, Quantity: 120, Scan Rate: 1000 [ms]. The 'Disable' section has 'Read/Write Disabled' and 'Disable on error' unchecked. The 'View' section has 'Rows' set to 10, 'Hide Alias Columns' unchecked, 'PLC Addresses (Base 1)' unchecked, 'Address in Cell' unchecked, and 'Enron/Daniel Mode' unchecked. Buttons for OK, Cancel, Apply, and Read/Write Once are present.

(4) Change the value of the Modbus HoldRegisters address bit 90. When it is 1 01, it switches to remote teaching mode .



The 'Write Single Register' dialog box is shown with the following settings: Slave ID: 1, Address: 90, Value: 101. The 'Result' section shows 'N/A' and 'Close dialog on "Response ok"' is unchecked. The 'Use Function' section has '06: Write single register' selected and '16: Write multiple registers' unselected. Buttons for Send and Cancel are present.

(5) Check the current working status. The specific operation process is shown in Section 1 of Chapter 4. If the address bit 9 of InputRegisters is 2, it means that it is currently in remote teaching mode. Or check the status bar of YiKingSoft software.



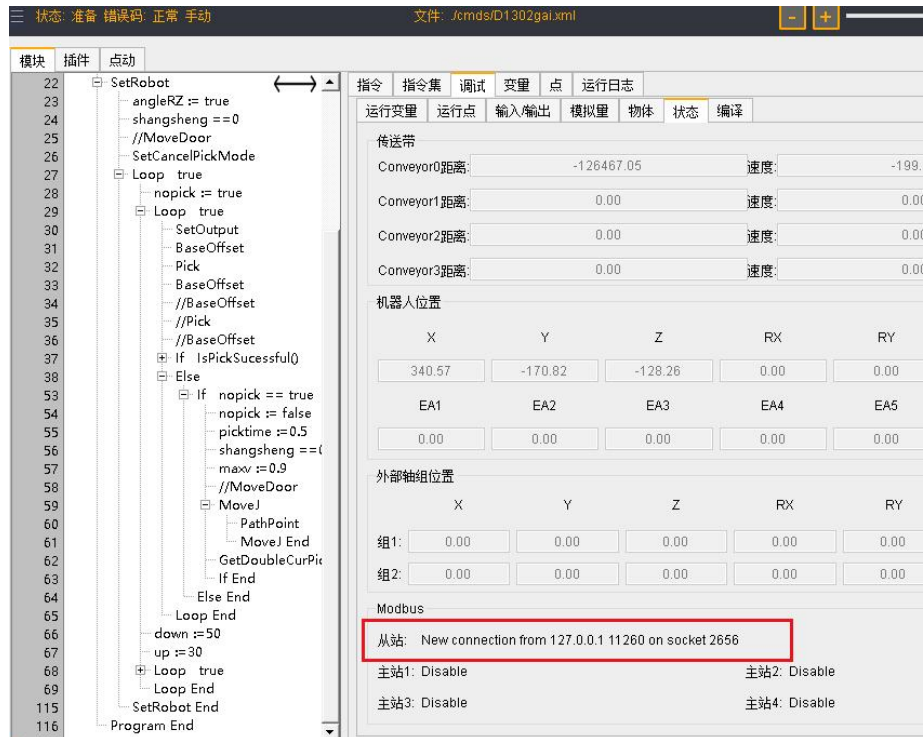
5.3 Setting the robot speed in remote teaching mode

When YiKingSoft is in remote teaching mode, the robot will have a speed independent of the local manual and automatic mode, that is, modifying the robot speed in remote teaching mode will not affect the robot speed in local and manual automatic mode.

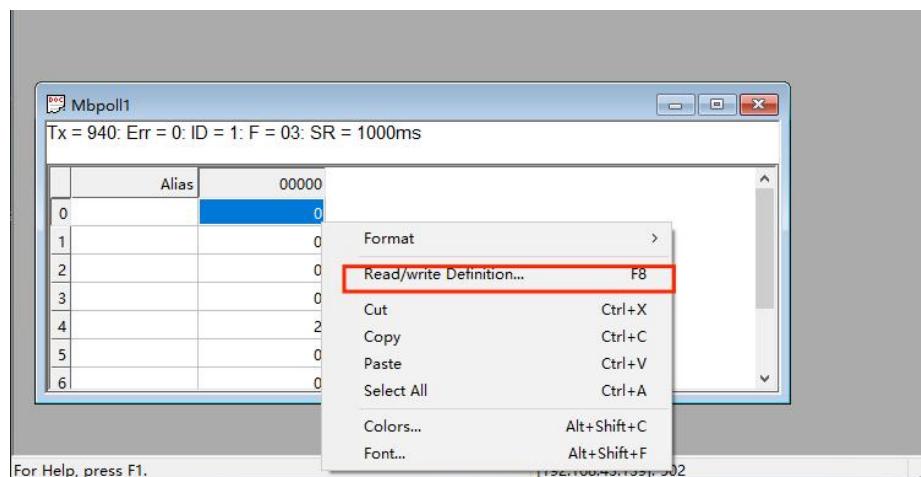
Each time you switch to remote teaching mode, the robot speed in remote teaching mode is set to 10.

The operation of setting the robot speed in remote teaching mode is as follows:

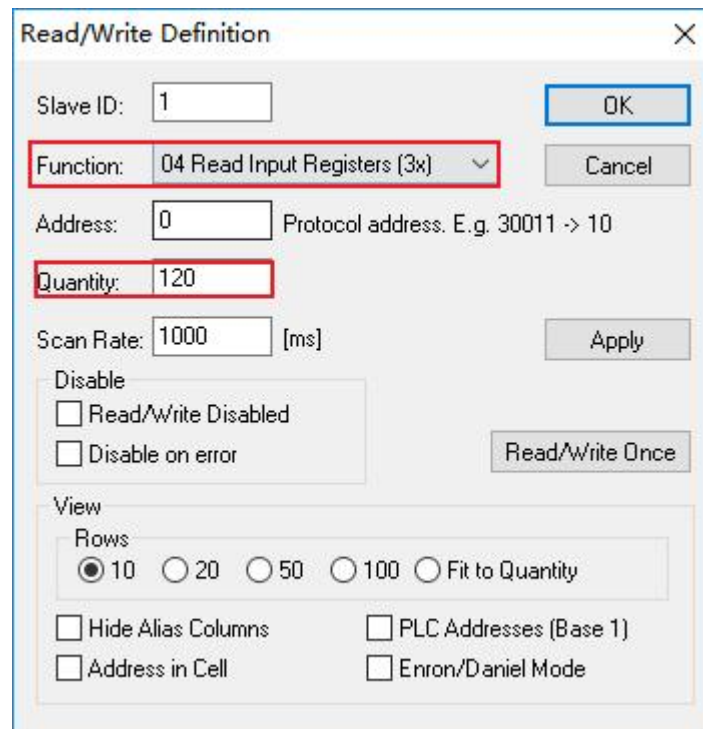
- (1) Modbus Poll software settings are as shown in Section 4.1.1 .
- (2) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station according to the above "2" and "3". This time, Modbus Poll is the master station and the robot software is the slave station.



(3) To modify the Modbus read/write function, right-click "Read/write Definition... (F8)" to enter the read/write definition modification interface.



(4) Enter the read/write definition modification interface, select the "Function" drop-down option and select "0 3 Read Holding Registers (4 x)", read the input register of the lower computer, and set the Quantity to 120, and click the "OK" button to complete the setting.



Read/Write Definition

Slave ID: 1

Function: 04 Read Input Registers (3x)

Address: 0 Protocol address. E.g. 30011 -> 10

Quantity: 120

Scan Rate: 1000 [ms]

Disable

☐ Read/write Disabled

☐ Disable on error

Read/write Once

View

Rows

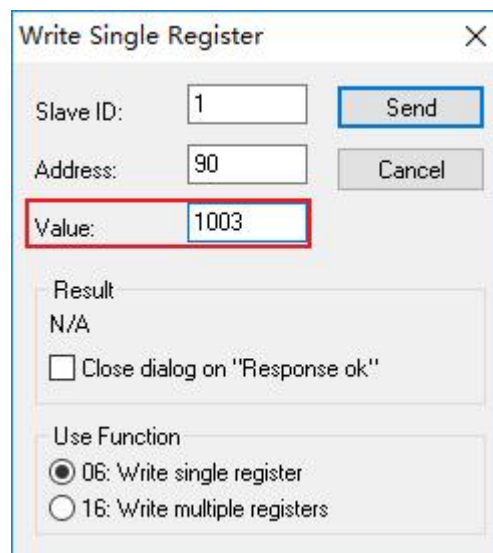
☒ 10 ☐ 20 ☐ 50 ☐ 100 ☐ Fit to Quantity

☐ Hide Alias Columns ☐ PLC Addresses (Base 1)

☐ Address in Cell ☐ Enron/Daniel Mode

(5) Switch to remote teaching mode. The specific operation is shown in 5.2.

(6) Set the speed of robot 1. Change the value of Modbus HoldRegisters address bit 90. When it is 1003, the speed of robot 1 decreases by 1 .



Write Single Register

Slave ID: 1

Address: 90

Value: 1003

Result

N/A

☐ Close dialog on "Response ok"

Use Function

☒ 06: Write single register

☐ 16: Write multiple registers

(7) Check the current speed of robot 1. The specific operation process is shown in Section 1 of Chapter 5. The address bit 10 of InputRegisters is displayed as 9, indicating that the current speed of robot 1 is 9 and the initial speed is 10,

indicating that the speed minus 1 function code is executed successfully.

The screenshot shows the Modbus Poll application window. At the top, there is a menu bar (File, Edit, Connection, Setup, Functions, Display, View, Window, Help) and a toolbar. Below the toolbar, a status bar displays 'Tx = 6: Err = 0: ID = 1: F = 04: SR = 1000ms'. The main area contains a table with 6 columns: 'Alias', '00000', 'Alias', '00010', and 'Alias'. The table has 10 rows, indexed 0 to 9. Row 0 is highlighted in blue. The '00010' column contains the value '9' for row 0. The '00000' column contains values: 3, 0, 0, 0, 0, -7100, 0, 0, 0, 2 for rows 0 through 9 respectively. The 'Alias' columns are empty. At the bottom, there is a status bar with the text 'For Help, press F1.' and the address '[127.0.0.1]: 502'.

	Alias	00000	Alias	00010	Alias
0		3		9	
1		0		0	
2		0		0	
3		0		0	
4		0		0	
5		-7100		0	
6		0		0	
7		0		0	
8		0		0	
9		2		0	

5.4 Jog operation in remote teaching mode

Currently, the jog operation in remote teaching mode supports the existing 4 coordinate systems. In the World, Base and TCP coordinate systems, the addition and subtraction in the X, Y and Z directions are supported; the rotation of the RZ axis is supported. The rotation of the RX and RY axes is not supported yet. In the Joint coordinate system, the addition and subtraction function codes of X, Y, Z, RX, RY and RZ represent the addition and subtraction function codes of axes 1–6 respectively.

After you execute a jog function code, as long as the HoldRegisters address bit 90 changes, the current motion will be stopped immediately regardless of the changed value. If the changed value is a certain function code, it will not be executed and

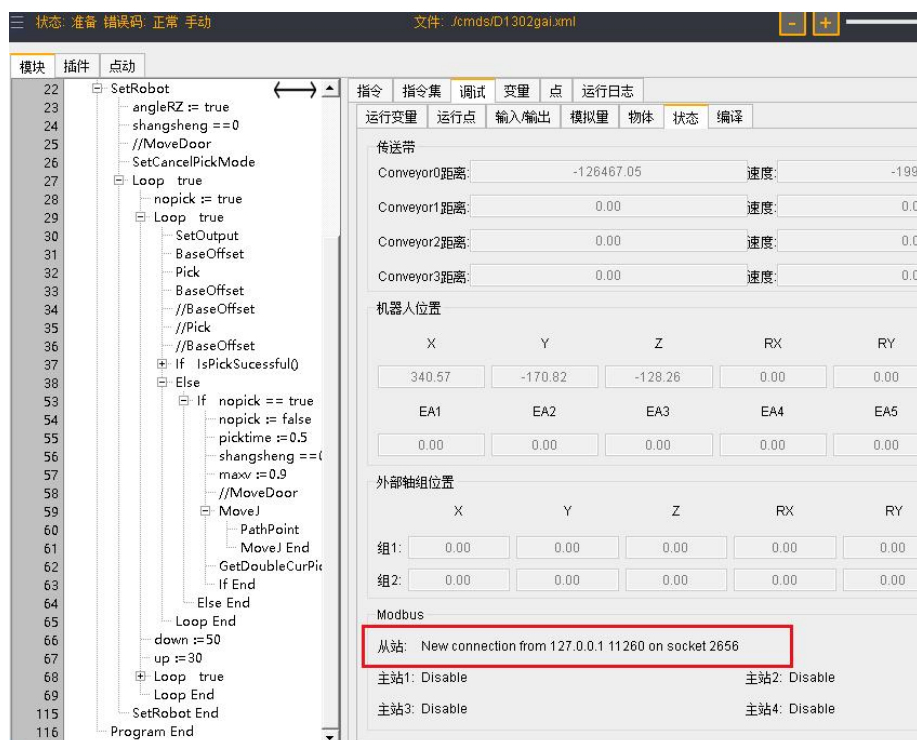
will only be regarded as a stop function for the current motion.

In order to ensure the safety of the jog operation in the remote teaching mode, the maximum operating distance of the single jog function operation of the X, Y, Z, and RZ function codes is 100 mm. The maximum operating distance of the single jog function operation of the axis 1-6 addition and subtraction function codes is 10 mm.

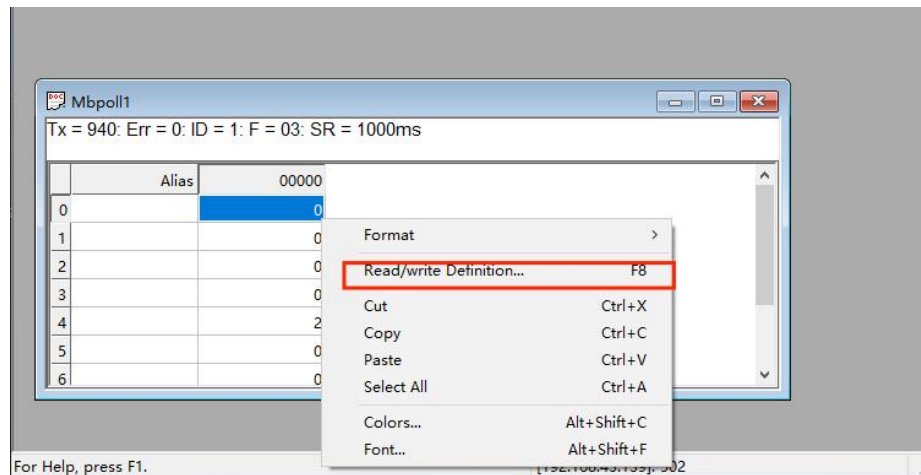
The following takes the X direction minus as an example to show the process of jog operation in remote teaching mode:

(1) Modbus Poll software settings are as shown in Section 4.1.1 .

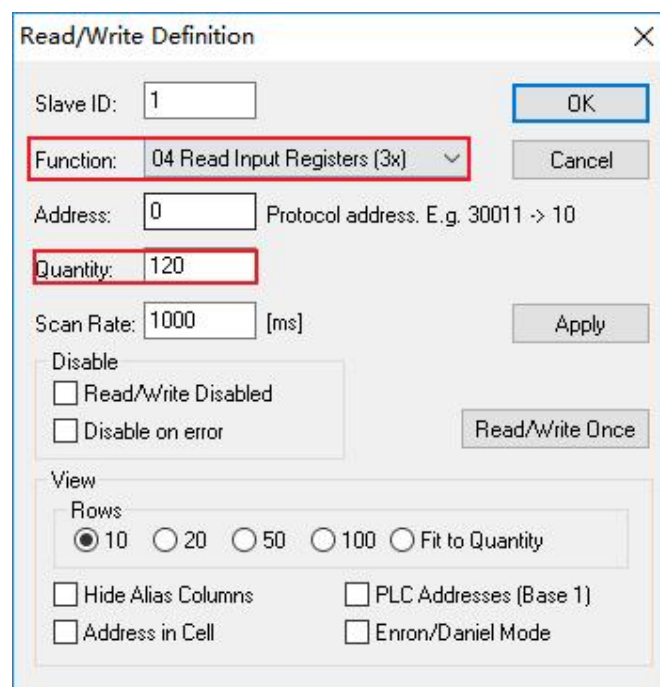
(2) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station according to the above "2" and "3". This time, Modbus Poll is the master station and the robot software is the slave station.



(3) To modify the Modbus read/write function, right-click "Read/write Definition... (F8)" to enter the read/write definition modification interface.



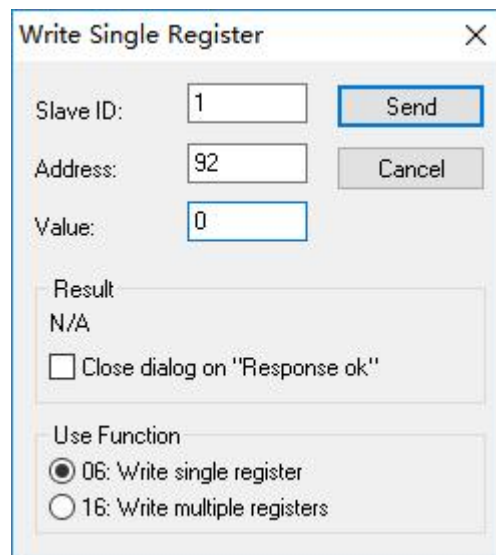
(4) Enter the read/write definition modification interface, select the "Function" drop-down option and select "0 3 Read Holding Registers (4 x)", read the input register of the lower computer, and set the Quantity to 120, and click the "OK" button to complete the setting.



(5) Switch to remote teaching mode. The specific operation is shown in 5.2.

(6) Set the robot speed. The specific operation is shown in 5.3.

(7) Set the robot motion coordinate system. Change the value of Modbus HoldRegisters address bit 92. 0 is the world coordinate system; 1 is the base coordinate system; 2 is the TCP coordinate system; 3 is the Joint coordinate system.



(8) Enable. Change the value of the Modbus HoldRegisters address bit 90. Function code 1001 is the enable switch of robot 1. If robot 1 is not enabled, enable robot 1. If robot 1 is already enabled, turn off the enable.

(9) Check whether it is currently enabled. Check InputRegisters address bits 0 and 1. The specific operation is shown in 4.1. When address bit 0 is 4 and address bit 1 is 2, it means that robot 1 is enabled and in pause status. Or refer to the status bar of YiKingSoft software.

Modbus Poll - [Mbpoll1]

File Edit Connection Setup Functions Display View Window Help

05 06 15 16 17 22 23 TC ? ?

Tx = 6: Err = 0: ID = 1: F = 04: SR = 1000ms

	Alias	00000	Alias	00010	Alias
0		4		9	
1		2		0	
2		0		0	
3		0		0	
4		0		0	
5		-7100		0	
6		0		0	
7		0		0	
8		0		0	
9		2		0	

For Help, press F1. [127.0.0.1]: 502

状态: 暂停 错误码: 正常 示教 文件: %10 W Robot1

(10) Execute the X-direction minus function. Change the value of the Modbus HoldRegisters address bit 90 , and the function code 503 is the X-direction minus operation. After the modification is successful, robot 1 starts to move in the negative X direction. The jog function has safety protection, and a single jog can move up to 100 mm.

(11) Stop the motion. Modify the value of the Modbus HoldRegisters address bit 90 to any value to stop the current motion.

5.5 PTP operation in remote teaching mode

The PTP operation in remote teaching mode supports addition and subtraction in the X, Y, and Z directions; rotation of the RZ axis, but does not currently support wafer handling robots and Mantis-series six-axis robots. The moving distance of the PTP operation is entered in HoldRegisters address bit 91, in units of 0.01 mm. Enter the moving coordinate system in HoldRegisters address bit 92. For specific function codes, see the appendix.

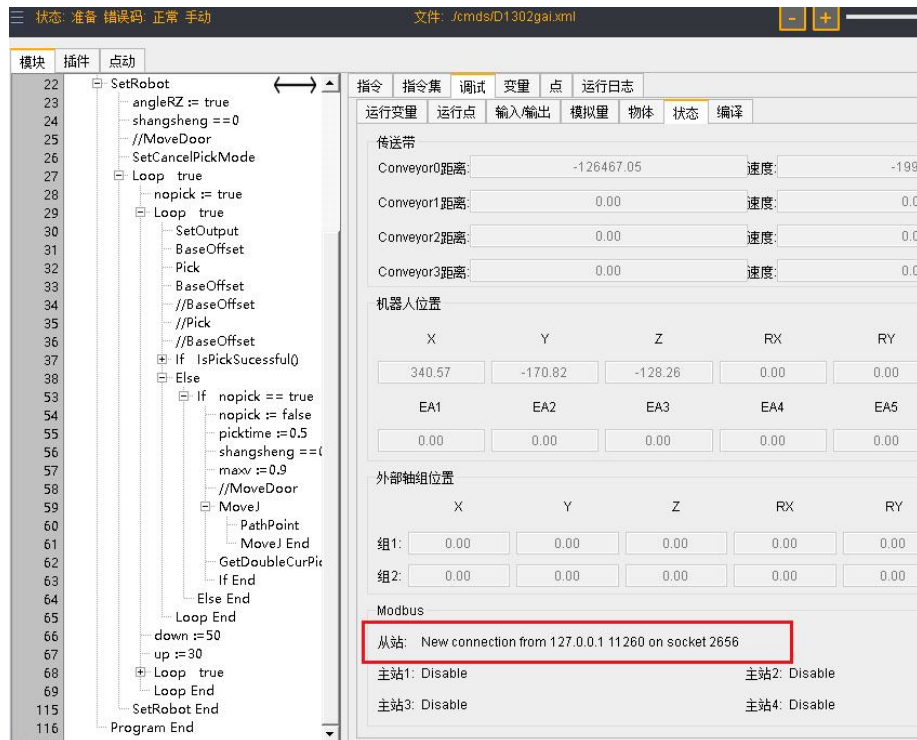
PTP operation in remote teaching mode requires setting the robot working mode to remote teaching mode, setting the PTP moving distance, and then enabling the robot. Use any PTP function code to switch the robot to the running status, and use the corresponding PTP function code again to start PTP motion. During or after the motion, enter any function code to exit the current motion or status and put the robot into the pause status.

If the robot has been enabled and switched to the running status, but you do not want to perform PTP operations and need to exit the running status and change to the paused status, you can enter any code value within the current robot function code range that is not a PTP operation to exit the running status. The following is an operation example of PTPX direction minus 10mm:

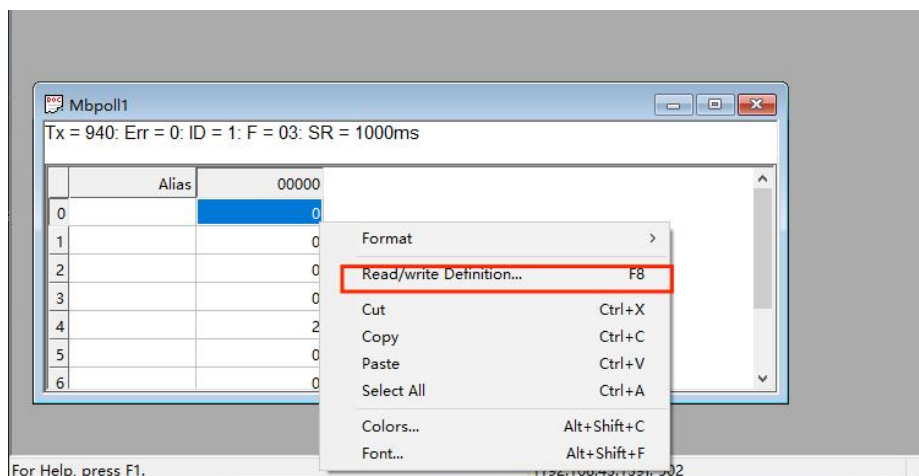
- (1) Modbus Poll software settings are as shown in Section 4.1.1 .
- (2) Open the YiKingSoft software interface. If the software master and slave stations are not set, you need to set the software as the master or slave station

RP-KZ-01-004-C

according to the above "2" and "3". This time, Modbus Poll is the master station and the robot software is the slave station.

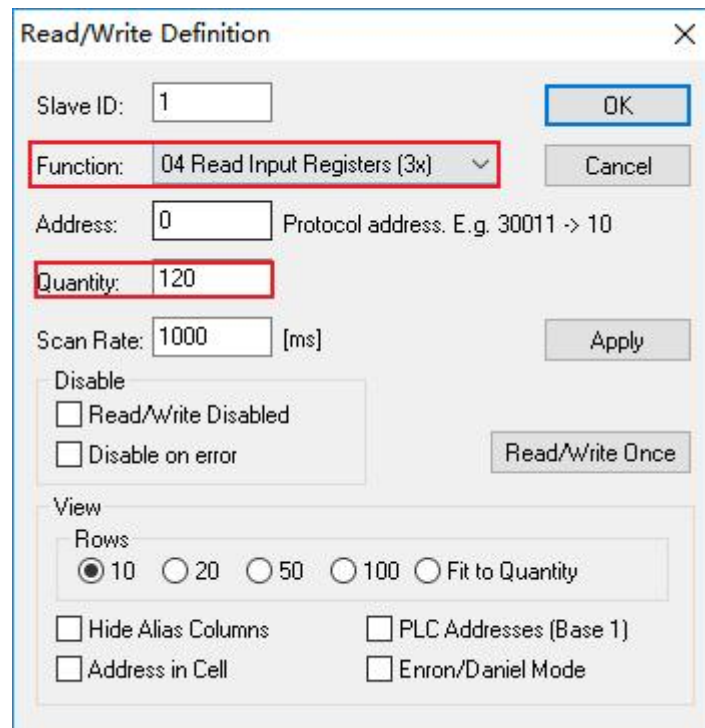


(3) To modify the Modbus read/write function, right-click "Read/write Definition... (F8)" to enter the read/write definition modification interface.



(4) Enter the read/write definition modification interface, select the "Function" drop-down option and select "0 3 Read Holding Registers (4 x)", read the input register of the lower computer, and set the Quantity to 120, and click the "OK" button

to complete the setting.



The image shows a 'Read/Write Definition' dialog box with the following fields and options:

- Slave ID: 1
- Function: 04 Read Input Registers (3x) (highlighted with a red box)
- Address: 0
- Quantity: 120 (highlighted with a red box)
- Scan Rate: 1000 [ms]
- Buttons: OK, Cancel, Apply, Read/Write Once
- Disable section:
 - ☐ Read/Write Disabled
 - ☐ Disable on error
- View section:
 - Rows: ☒ 10, ☐ 20, ☐ 50, ☐ 100, ☐ Fit to Quantity
 - ☐ Hide Alias Columns
 - ☐ Address in Cell
 - ☐ PLC Addresses (Base 1)
 - ☐ Enron/Daniel Mode

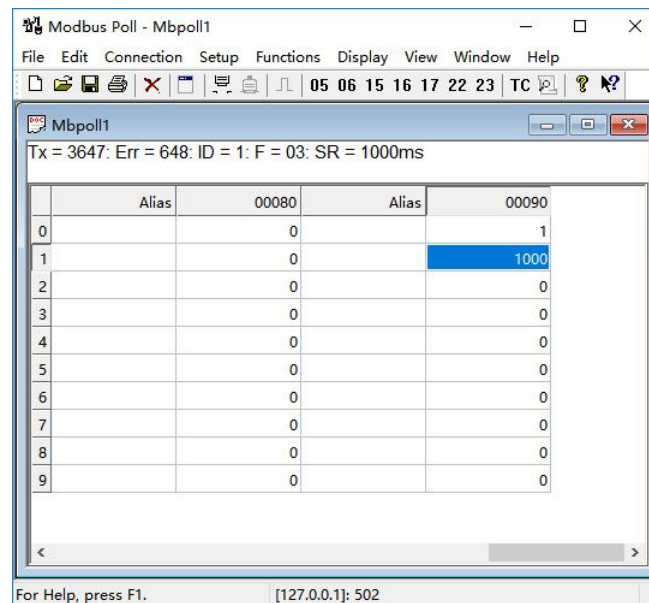
- (5) Switch to remote teaching mode. The specific operation is shown in 5.2.
- (6) Set the robot speed. The specific operation is shown in 5.3.
- (7) Enable. Change the value of the Modbus HoldRegisters address bit 90. Function code 1001 is the enable switch of robot 1. If robot 1 is not enabled, enable robot 1. If robot 1 is already enabled, turn off the enable.
- (8) Check whether it is currently enabled. Check InputRegisters address bits 0 and 1. The specific operation is shown in 4.1. When address bit 0 is 4 and address bit 1 is 2, it means that robot 1 is enabled and in pause status. Or refer to the status bar of YiKingSoft software.

The screenshot shows the Modbus Poll software window. The title bar reads "Modbus Poll - [Mbpoll1]". The menu bar includes File, Edit, Connection, Setup, Functions, Display, View, Window, and Help. The toolbar contains various icons for file operations and communication. Below the toolbar, the status bar displays "Tx = 6: Err = 0: ID = 1: F = 04: SR = 1000ms".

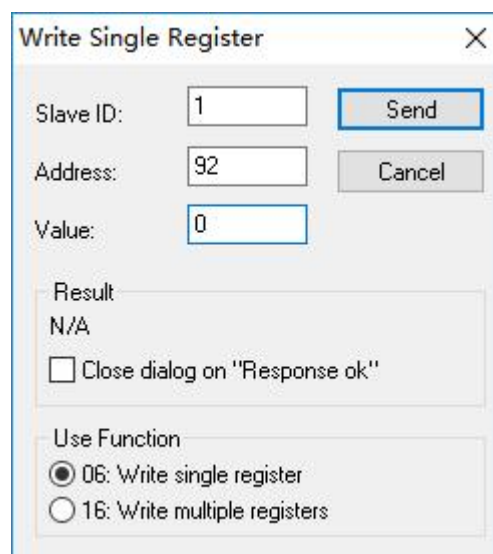
	Alias	00000	Alias	00010	Alias
0		4		9	
1		2		0	
2		0		0	
3		0		0	
4		0		0	
5		-7100		0	
6		0		0	
7		0		0	
8		0		0	
9		2		0	

At the bottom of the window, the status bar shows "For Help, press F1." and "[127.0.0.1]: 502". Below the main window, there is a separate bar with a red box highlighting the text "状态: 暂停" (Status: Pause), followed by "错误码: 正常 示教" (Error code: Normal Teaching), a file name field, and a dropdown menu showing "Robot1".

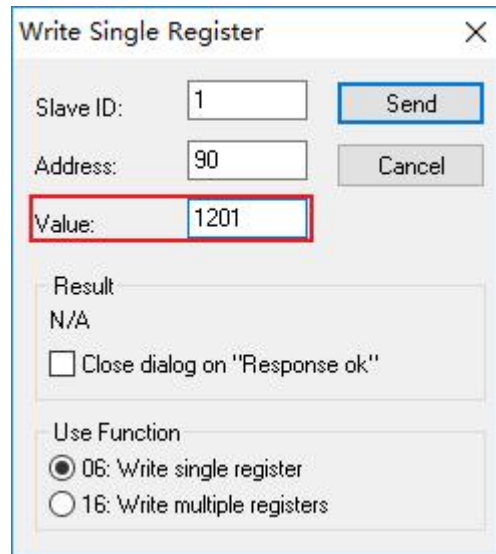
(9) Set the PTP operation moving distance. Set the value of HoldRegisters address bit 91 to 1000, which means a movement of 10 mm.



(10) Set the robot motion coordinate system. Change the value of Modbus HoldRegisters address bit 92. 0 is the world coordinate system; 1 is the base coordinate system; 2 is the TCP coordinate system; 3 is the Joint coordinate system.



(11) Switch the robot status to the running status. Set the value of the HoldRegisters address bit 90 to any PTP operation function code to switch to the running status. Here, function code 1201 is used.



Write Single Register

Slave ID: 1 Send

Address: 90 Cancel

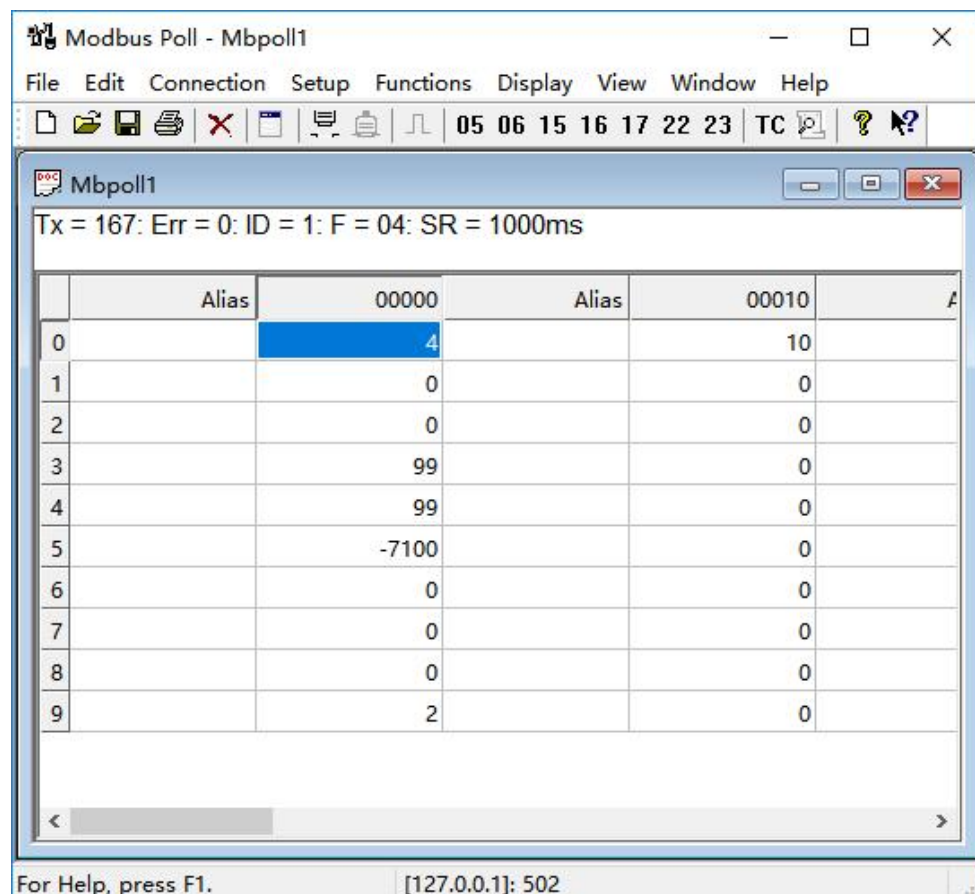
Value: 1201

Result
N/A

☐ Close dialog on "Response ok"

Use Function
☒ 06: Write single register
☐ 16: Write multiple registers

(12) Check the current status. Switch to InputRegisters to check address bits 0 and 1. The specific operation is shown in 5.1. When address bit 0 is 4 and address bit 1 is 0, it means the robot is already in the running status. Or check the YiKingSoft status bar.



Modbus Poll - Mbpoll1

File Edit Connection Setup Functions Display View Window Help

05 06 15 16 17 22 23 TC ? ?

Mbpoll1

Tx = 167: Err = 0: ID = 1: F = 04: SR = 1000ms

	Alias	00000	Alias	00010	
0		4		10	
1		0		0	
2		0		0	
3		99		0	
4		99		0	
5		-7100		0	
6		0		0	
7		0		0	
8		0		0	
9		2		0	

For Help, press F1. [127.0.0.1]: 502



(13) Execute the PTPX direction subtraction operation. You need to execute the 553 function code again. Since the Modbus detects the change of the register edge, you need to first change the code value in address bit 90 to the default value 0, and then use the 1201 function code again to execute the PTP operation.

6. Robot Modbus slave address allocation table

6.1 Bits(Coil)

0-100

6.2 InputBits

0-100

6.3 HoldRegisters

0-79: System register, used for external input start, stop, pause, resume and other functions

80-99: System register robot public area

100-399: Assigned to users to customize communications through commands

0: Robot 1 start command number	Corresponding startup Cmds\ExStartCmd\Cmd_XX.xml
1: Start and stop	The default value is 0. If 1 is received, it will start. If 2 is received, it will stop.
2: Pause recovery	The default value is 0. If 1 is received, it will pause. If 2 is received, it will resume.
3: Error reset	The default value is 0. The error reset 1 is received-reset stop, the error reset 2 is received -command resume.
4: Emergency stop reset	The default value is 0. The emergency stop reset 1 is received-reset stop, The emergency stop reset 2 is received-command resume.
...	
20: Robot 2 start command number	Corresponding startup Cmds\ExStartCmd\Cmd_XX.xml
21: Start and stop	The default value is 0. If 1 is received, it will start. If 2 is received, it will stop.
22: Pause and resume	The default value is 0. If 1 is received, it will pause. If 2 is received, it will resume.

RP-KZ-01-004-C

...	
40: Robot 3 start command number	Corresponding startup Cmds\ExStartCmd\Cmd_XX.xml
41: Start and stop	The default value is 0. If 1 is received, it will start. If 2 is received, it will stop.
42: Pause and resume	The default value is 0. If 1 is received, it will pause. If 2 is received, it will resume.
...	
60: Robot 4 start command number	Corresponding startup Cmds\ExStartCmd\Cmd_XX.xml
61: Start and stop	The default value is 0. If 1 is received, it will start. If 2 is received, it will stop.
62: Pause and resume	The default value is 0. If 1 is received, it will pause. If 2 is received, it will resume.
...	
80: Start all robots at the same time	The default value is 0. If 1 is received, it will start. If 2 is received, it will stop. The start command is the corresponding command stored in register 0.

RP-KZ-01-004-C

...	
90: Modbus	Default value 0
remote teaching	Mode Switching 101
mode jog	Robot 1 jog function code (1001-1999)
function	Enable 1001
	Speed level -1 1002 Speed level +1 1012
	Speed -1 1003 Speed +1 1013
	X- 1101 X+ 1111
	Y- 1102 Y+ 1112
	Z- 1103 Z+ 1113
	RX-1104 RX+1114
	RY- 1105 RY+ 1115
	RZ-1106 RZ+1116
	PTP. X- 1201 PTP. X+ 1211
	PTP. Y- 1202 PTP. Y+ 1212
	PTP. Z- 1203 PTP. Z+ 1213
	PTP. RX- 1204 PTP. RX+ 1214
	PTP. RY- 1205 PTP. RY+ 1215
	PTP. RZ- 1206 PTP. RZ+ 1216

RP-KZ-01-004-C

The jog function code of robot 2 is based on robot	
1 + 1000 (2001-2999)	
Enable 2001	
Speed Level -1 2002	Speed Level +1 2012
Speed -1 2003	Speed +1 2013
X- 2101	X+ 2111
Y- 2102	Y+ 2112
Z- 2103	Z+ 2113
RX-2104	RX+2114
RY- 2105	RY+ 2115
RZ- 2106	RZ+ 2116
PTP. X- 2201	PTP. X+ 2211
PTP. Y- 2202	PTP. Y+ 2212
PTP. Z- 2203	PTP. Z+ 2213
PTP. RX- 2204	PTP. RX+ 2214
PTP. RY- 2205	PTP. RY+ 2215
PTP. RZ- 2206	PTP. RZ+ 2216
The jog function code of robot 3 is based on robot	
2 +1000	

	... The function code of robot 4 is based on robot 3 +1000 ...
91 Set PTP moving step	The unit is 0.01 mm, the maximum value is 10000
92 Set the current coordinate system	0 world coordinate system 1 base coordinate system 2 TCP coordinate system 3 Joint Coordinate System
...	
100 and above are general registers, which can be assigned to variables for programmable use	
...	
400	

6.4 InputRegisters

0-200

0: Status code	StatusCode: NoInit=0 Init=1 SelfChecking=2 Ready=3 Run=4: See register 1 for the running status small status code JogRun=5 BackZero=6 Brake=9 EStop=10 Error=11 SafeRun=12
1: Small status code	The status in this register is valid only when register 0 is in the Run status. RunStatus (0): Running RunPauseStatus (1): Start pausing PauseStatus (2): Pause status, in which you can switch to jog or resume motion GoToPausePos (3): Start running to the pause point

	RunResumeStatus (5): Starts to resume running and automatically switches to RunStatus RobotJOG (7)
2: Alarm code	ErroCode:
3: PosX	Unit 0.1 mm
4: PosY	Unit 0.1 mm
5: PosZ	Unit 0.1 mm
6: RX	Unit 0.1 degree
7: RY	Unit 0.1 degree
8: RZ	Unit 0.1 degree
9: Check manual and automatic	0 is manual, 1 is automatic , 2 is remote teaching, 3 is remote debugging, 4 is remote running
10: Query current speed	unit%
...	
50: Status code	StatusCode: NoInit=0 Initd=1 SelfChecking=2 Ready=3 Run=4: See register 1 for the running state small status code

	JogRun=5 BackZero=6 Brake=9 EStop=10 Error=11 SafeRun=12
51: Minor status code	The status in this register is valid only when register 50 is in the Run status. RunStatus (0): Running RunPauseStatus (1): Start pausing PauseStatus (2): Pause status, in which you can switch to jog or resume motion status GoToPausePos (3): Start running to the pause point RunResumeStatus (5): Starts to resume running and automatically switches to RunStatus RobotJOG (7)
52: Alarm code	ErroCode:
53: PosX	Unit 0.1 mm
54: PosY	Unit 0.1 mm
55: PosZ	Unit 0.1 mm
56: RX	Unit 0.1 degree
57: RY	Unit 0.1 degree

58: RZ	Unit 0.1 degree
59: Check manual and automatic	0 is manual, 1 is automatic , 2 is remote teaching, 3 is remote debugging, 4 is remote running
60: Query current speed	unit%
...	
100: Status code	StatusCode: NoInit=0 Init=1 SelfChecking=2 Ready=3 Run=4 JogRun=5 BackZero=6 Brake=9 EStop=10 Error=11 SafeRun=12
101: Minor status code	The status in this register is valid only when register 100 is in the Run status. RunStatus (0): Running

	RunPauseStatus (1): Start pausing PauseStatus (2): Pause status, in which you can switch to jog or resume motion status GoToPausePos (3): Start running to the pause point RunResumeStatus (5): Starts to resume running and automatically switches to RunStatus RobotJOG (7)
102: Alarm code	ErroCode:
103: PosX	Unit 0.1 mm
104: PosY	Unit 0.1 mm
105: PosZ	Unit 0.1 mm
106: RX	Unit 0.1 degree
107:RY	Unit 0.1 degree
108: RZ	Unit 0.1 degree
109: Check manual and automatic	0 is manual, 1 is automatic , 2 is remote teaching, 3 is remote debugging, 4 is remote running
110: Query current speed	unit%
...	
150: Status code	StatusCode: NoInit=0

	Inited=1 SelfChecking=2 Ready=3 Run=4 JogRun=5 BackZero=6 Brake=9 EStop=10 Error=11 SafeRun=12
151: Minor status code	The status in this register is valid only when register 150 is in the Run status. RunStatus (0): Running RunPauseStatus (1): Start pausing PauseStatus (2): Pause status, in which you can switch to jog or resume motion status GoToPausePos (3): Start running to the pause point RunResumeStatus (5): Starts to resume running and automatically switches to RunStatus RobotJOG (7)
152: Alarm code	ErroCode:
153: PosX	Unit 0.1 mm

154: PosY	Unit 0.1 mm
155: PosZ	Unit 0.1 mm
156: RX	Unit 0.1 degree
157:RY	Unit 0.1 degree
158: RZ	Unit 0.1 degree
159: Check manual and automatic	0 is manual, 1 is automatic , 2 is remote teaching, 3 is remote debugging, 4 is remote running
160: Query current speed	unit%
...	
200	

Revision Record

version number	Modifications	Modified by	Approver	Approver	Change the time
A-0	Drafting				2019\09\18
A-1	InputRegisters deletion status code	Huang Xianrong			2021\07\08
A-2	Re-enter the file and change the file number	Li Changyu			2022.03.22
V1.1	Added the manual and automatic reading function and reset function, and updated the	Jiang Hongliang			2022.09.21

	Modbus TCP content layout.				
V1.2	Change the address where the manual and automatic status is located. The manual and automatic status is located at the input register address	Jiang Hongliang			2022. 09. 23
V1.3	1. Section 4.1: Reading Manual and Automatic Status to Reading Working Status. 2. Added Chapter 5 Modbus Remote Jog. 3. Added the functional description of No. 90, No. 91, and No. 92 in the new appendix HoldRegisters. 4. Added the functions of address bits 9 and 10 in the new appendix InputRegisters. 5. Effective version 2.9.5 and above.	Fu Guocheng			2023. 04. 04