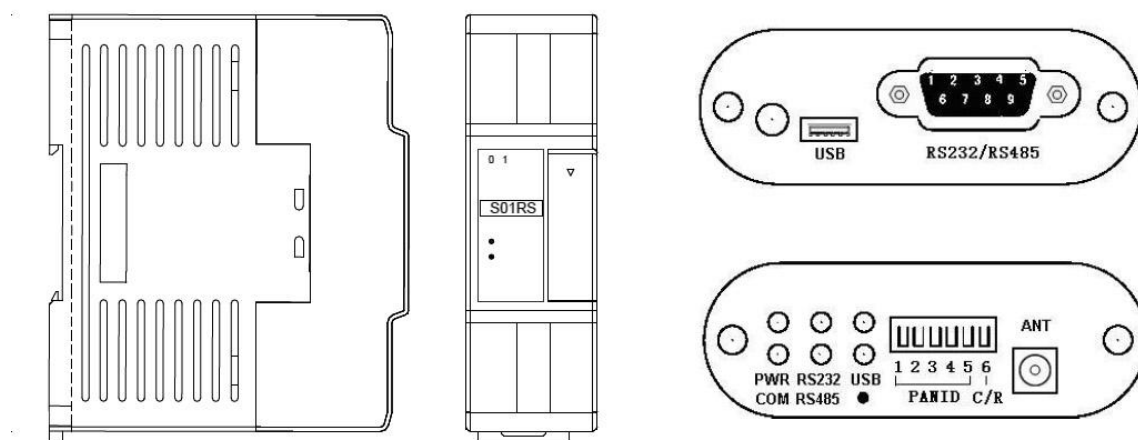


Haiwell PLC User Manual

Programmable Logic Controller

Communication Module User Manual & Application Case



Contents

Communication Expansion Module User Manual

1. Product model list and dimension.....	3
2. Indicator description.....	3
3. Environmental specification for product.....	4
S01RS module.....	4
H01ZB module.....	4
PC2ZB module.....	4
4. Module terminal wiring diagram.....	5
5. S01RS wiring diagram (RS485&RS232).....	5
6. Module connection method.....	5

Haiwell communication expansion module S01RS application example

1. Module power supply.....	6
2. Applicable host PLC.....	6
3. Hardware configuration and communication port number.....	6
4. Communication protocol.....	6
5. Haiwellbus communication example description: Communication between two Haiwell host PLCs.....	7
6. Modbus communication example introduction:.....	9
7. Typical freedom protocol applications:.....	9
8. The system registers of communication timeout time, communication instruction execution interval, communication port character receiving timeout time and application introduction.....	14
9. When PLC is used as slave station, there is no need to write any communication program, and supporting a variety of human-machine interfaces and configuration softwares.....	16
10. how to judge the communication failure and program analysis when PLC is used as slave.....	17
11. Check-code calculator usage introduction.....	17
12. The introduction of supportive baud rate, data format and communication instructions when PLC communication port is used as a master/slave station.....	18
13. PLC communication frequently asked questions.....	18

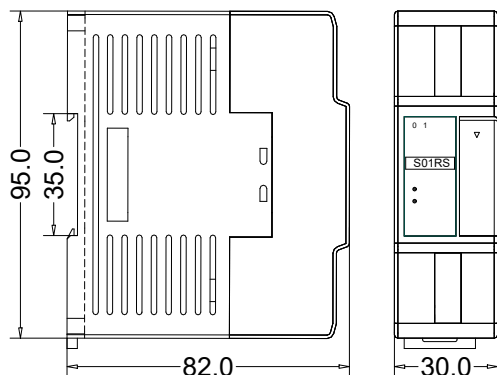
Haiwell wireless zigbee communication modules of H01ZB and PC2ZB application examples

1. Module introduction.....	20
2. Module power supply.....	20
3. Applicable object.....	20
4. Hardware configuration and communication port number.....	20
5. Networking.....	20
6. DIP switch settings.....	21
7. Notes.....	21
8. Driver download and installation of PC2ZB and PC communication settings.....	21
9. Communication between PLCs through H01ZB module.....	22

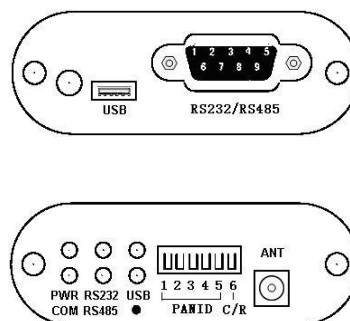
Communication Expansion Module User Manual

1. Product model list and dimension

Model	Power(24VDC)	Dimension
S01RS	0.4VA	30*95*82mm
H01ZB	0.4VA	
PC2ZB	0.1VA	48*70*24mm



S01RS, H01ZB-Dimension figure



PC2ZB-Dimension figure

2. Indicator description

S01RS module

- ① PWR: Power indicator, green. Normally on-power normal; off - power abnormal.
- ② LINK: Normally on in green - Module is connected to host PLC normally; Flicker in green - Module is interacting data with host PLC; Flicker in red - Firmware is incomplete.
- ③ COM indicator digit 0: Indicates that communication port 0 is RS232 or RS485 communication mode, when it is used as a master, it sends command with flickering; when it is used as a slave, it replies data with flickering

H01ZB module

- ① PWR: Power indicator, green. Normally on -power normal; off -power abnormal.
- ② LINK: Normally on in green - Module is connected to host PLC normally; Flicker in green - Module is interacting data with host PLC; Flicker in red - Firmware is incomplete.
- ③ COM: Network indicator, red. Normally on - Joined the network; Normal flickering - Search and set up networking (on 250ms, off 250ms); Fast flickering - Receive data (on 30ms, off 30ms).

PC2ZB module

- ① PWR: Power indicator, green. Normally on -power normal; off -power abnormal.
- ② COM: Network indicator, red. Normally on - Joined the network; Normal flickering - Search and set up networking (on 250ms, off 250ms); Fast flickering - Receive data (on 30ms, off 30ms).
- ③ RS232 / RS485 / USB: Communication indicator, green, the corresponding indicator will flash when using any communication method of them.

3. Environmental specification for product

S01RS module

Item	Environment Specification
Temperature/Humidity	Operating temperature:0~+55℃ Storage temperature:-25~+70℃ Humidity: 5~95%RH, No condensation
Vibration resistance	10~57 HZ, amplitude=0.075mm, 57HZ~150HZ acceleration=1G, 10 times each for X-axis, Y-axis and Z-axis
Impact resistance	15G, duration=11ms, 6 times each for X-axis, Y-axis and Z-axis
Interference immunity	DC EFT:±2500V Surge:±1000V
Over voltage resistance	1500VAC/1min between AC terminal and PE terminal, 500VAC/1min between DC terminal and PE terminal
Insulation impedance	≥ 5MΩbetween AC terminal and all input/output points to PE terminal @500VDC
Earth	The third type of grounding (Cannot be connected with strong electrical system' earth)
Operating environment	Avoid dust, moisture, corrosion, electric shock and external shocks
Isolation mode	No isolation between channels, use optoelectronic isolation for the communication interface and the internal power supply

H01ZB module

Item	Specification
Power supply	Powered by host PLC expansion port
Environment temperature	Operating temperature:0~+55℃ Storage temperature:-25~+70℃ Humidity: 5~95%RH, No condensation
Operating frequency	2.4~2.5GHz
Transmit power	+14dBm
Receiver sensitivity	-95 dBm
Allowed relay(routing) number	Up to 6 by default
Protocol conversion	ZigBee<=>modbus
Maximum communication rate	38400bps
The number of channels	6
PANID range	1~31
The largest network capacity	65535
Reliable transmission distance	150m (No hinder , linear distance)
ZigBee protocol version	ZigBee Pro
Network type	Mesh
Node type	The coordinator, routers
Antenna type	SMA

PC2ZB module

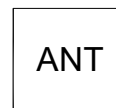
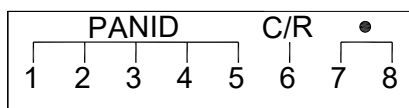
Item	Specification
Power supply	USB port power supply(+5V)
Environment temperature	Operating temperature: 0~+55 ℃; Storage temperature:-25~+70 ℃; Humidity: 5~95% RH, No condensation
Operating frequency	2.4~2.5GHz
Transmit power	+14dBm
Receiver sensitivity	-95 dBm
Allowed relay(routing) number	Up to 6 by default
Protocol conversion	Zigbee<=>modbus
Maximum communication rate	38400bps
The number of channels	6
PANID range	1~31
The largeest network capacity	65535
Reliable transmission distance	150m (No hinder , linear distance)
ZigBee protocol version	ZigBee Pro
Network type	Mesh
Node types	The coordinator, routers
Antenna type	SMA

4. Module terminal wiring diagram

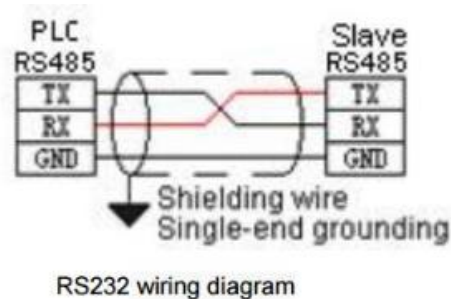
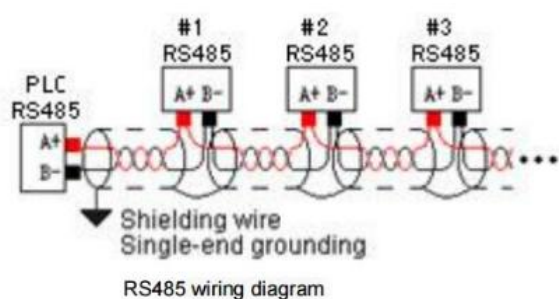
S01RS module:

RS232	Tx
	Rx
	GND
RS485	A+
	B-

H01ZB module:

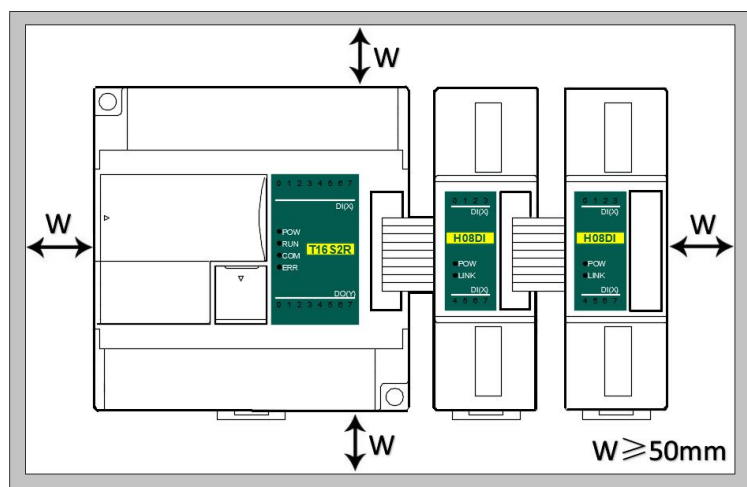


5. S01RS wiring diagram (RS485&RS232)



6. Module connection method

The connection between expansion module and host PLC(or other expansion module) is achieved by Bus. Each expansion module comes with an expansion cable for connecting to the previous module. Open the small flip of the previous module, insert the expansion cable connector of the module which needs to be connected with into the expansion interface of the previous module, and close the small flip of the previous expansion module to reset it after plugging in firmly.



Haiwell communication expansion module S01RS application example

1. Module power supply

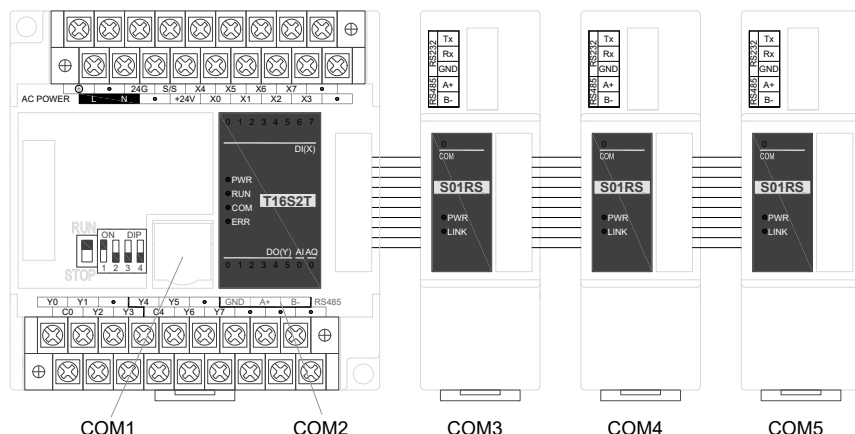
Module is directly hung behind the host PLC by parallel port and powered by parallel port, no need to take external power supply.

2. Applicable host PLC

N, H, T, S series PLC can be expanded with communication modules, the maximum can be expanded with 3 S01RS modules.

3. Hardware configuration and communication port number

Host PLC is built-in 2 communication ports, respectively COM1 for RS232 port, round mouth; COM2 for RS485 port, the A + B-terminal on terminal row. Communication expansion module S01RS not only can be used as 232, but also can be used as 485 , no program definition required. It depends on the external wiring, S01RS can be used as 485 when connected to 485 port or it can be used as 232 when connected to 232 port, but it only can choose one or the other one.

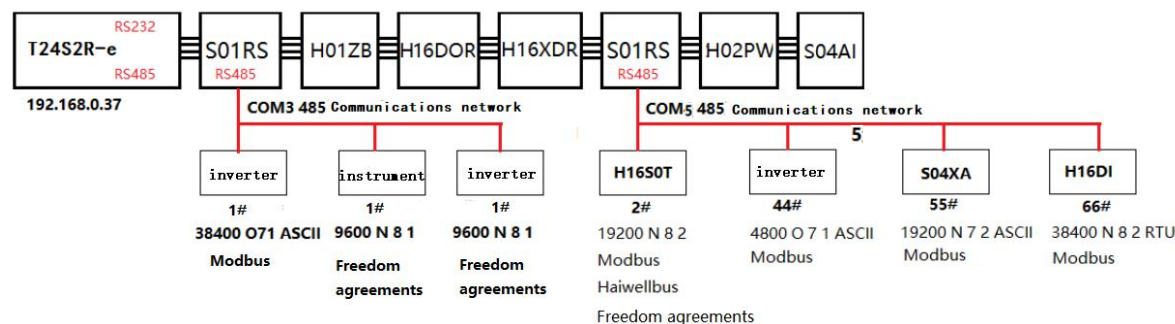


When the communication module is expanded, the communication port closest to the host PLC is COM3, next is COM4 and last is COM5. The position of communication module can be random, and the communication port number can be viewed in the hardware configuration, as shown in the figure below:

Index	Module type	X Component	Y Component	AI Component	AQ Component	Other	Description
0	T16S2T/P(-e)	X0 - X7	Y0 - Y7			COM1-2 HSC0-1 PLS0-1 Port3	CPU module 8*DI 8*DO transistor AC220V power supply 2 channel 200KHz pulse input 2 channel 200KHz pulse output 1 serial port RS232/RS485 support Modbus/Haiwellbus/free protocol Master/Slave
1	S01RS						Analog input module 4*AI analog input. built-in RS485 communication ports support remote
2	S04AI			AI0 - AI3			Digital output module 16*DO transistor. built-in RS485 communication ports support remote
3	H16DOT/P		Y8 - Y23			Port4	1 serial port RS232/RS485 support Modbus/Haiwellbus/free protocol Master/Slave
4	S01RS					Port5	1 serial port RS232/RS485 support Modbus/Haiwellbus/free protocol Master/Slave
5	S01RS						

4. Communication protocol

It is built-in Modbus RTU / ASCII protocol, freedom communication protocol and Haiwellbus high-speed communication protocol of Haiwell company(each port supports the above protocols). One communication port can take different baud rates, different data formats, different communication protocols at the same time, as you can see in the following network diagram, two 485 networks can be normal and efficient communication.



5. Haiwellbus communication example description: Communication between two Haiwell host PLCs

This example shows the communication between two host PLC. It is known that communication baud rate is 19200 between the master and slave PLC, data format is N 8 2 RTU, and slave station number is 2.

- ① Master reads 2 # slave data: for example, read X0 of slave into master M0; read X3 of slave into master M11, etc., as follows:

Index	Component read from slave	Component write to master
1	X0	M10
2	X3	M11
3	V11	V80
4	V12	V81
5	AI0	V20
6	AI1	V21

- ② Master writes 2 # slave data: for example, write V0 of master into slave V100; write V50 of master into slave V102, etc., as follows:

Index	Component read from master	Component write to slave
1	X0	M100
2	X1	M101
3	V0	V100
4	V50	V102
5	Y4	M0
6	Y5	Y0
7	V60	V200
8	V61	V201

Programming ideas:

In the Haiwell PLC programming software, set up Haiwellbus read communication table, special floating-point and 32-bit data occupy two consecutive addresses, if the master wants to write the floating-point or 32-bit data into the slave, it needs to fill two consecutive registers, such as write V11V12 into V80V81, as follows:

Table name: Read 2# slave data

Index	Component read from slave	Component write to master	Component comments
1	X0	M10	
2	X3	M11	
3	V11	V80	
4	V12	V81	
5	AI0	V20	
6	AI1	V21	

Buttons: Append, Insert, Delete, Move up, Move down

Comments:

Password: Confirm password:

Buttons: Help, OK, Cancel

In the Haiwell PLC programming software, set up Haiwellbus write communication table, as follows:

Haiwellbus write table

Table name: writes2 #slave date

Index	Component read from master	Component write to slave	Component comments
1	X0	M100	
2	X1	M101	
3	V0	V100	
4	V50	V102	
5	Y4	M0	
6	Y5	Y0	
7	V60	V200	
8	V61	V201	

Append
Insert
Delete
Move up
Move down

Comments

Password:
Confirm password:

Help OK Cancel

After the table is established, use HWRD and HWWR instructions, and enter the slave address in the "Slave" terminal of the instructions. In this case, the slave PLC address is "2", directly input the table name in the "Table", or double-click the terminal to select the established table. "Port" is the communication port number, as described in the second point above, "2" indicates the master's COM2 for 485 communication port. In this way, the communication program which reads and writes from the slave is completed, as follows:



Judge the communication error code situation through the communication instruction "Out" terminal. "Out" terminal has power indicating that communication is very good without error; "Out" terminal has no power indicating that the communication does not succeed, in this way, just check the slave parameter settings and networking wiring; "Out" terminal flashes to indicate a communication error code, and occasional error code in communications does not matter, communication may be disturbed, therefore, it is possible to check whether the outside uses shielded twisted pair or not, also, multiple slaves need to be in the way of "hand in hand" or add communication isolator S01GL from Haiwell company. The following is an example of a disconnection alarm program for slave communication:



Communication disconnection time can be set according to the actual, the above when M5 is more than 5 seconds in OFF state, T0 coil will indicates communication disconnection alarm.

6. Modbus communication example introduction:

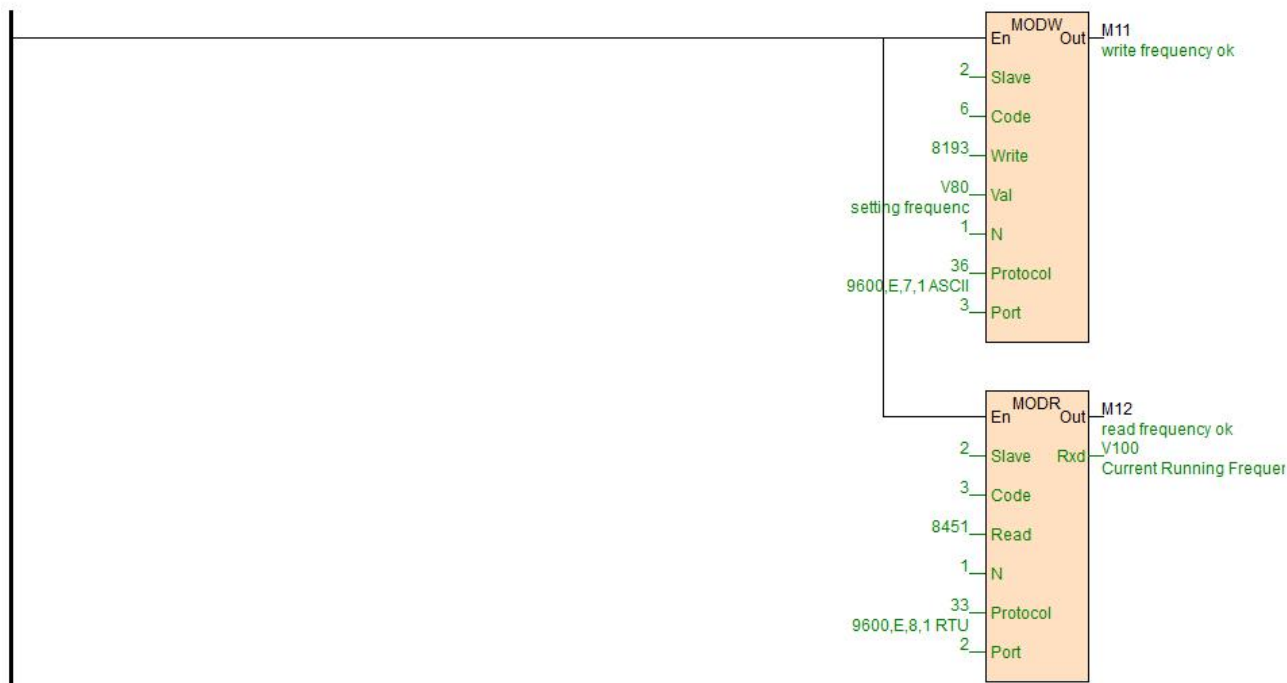
In this example, the host PLC is expanded with a communication expansion module S01RS, communication port is 3, the communication port is connected to a Haiwell thermal resistance module H08RC and a inverter:

- ① H08RC module is built-in RS485 port which can be used for remote IO, supporting Modbus RTU / ASCII protocol, communication parameters is 19200 N 8 2 RTU, station number is 1 #, from the programming software online help - hardware manual - expansion module parameters, we can see the 8 temperature values are stored in Modbus start address 10 ~ 17H, which is decimal 16 ~ 23.
- ② Inverter, 9600 E 7 1 ASCII, station No. 2 #. It is required to read and write the inverter frequency, from the inverter manual we can see that the parameter address of the setting frequency is 2001H, decimal 8193, and the parameter address of the operating frequency is 2103H, decimal 8451. In this way, write the program as follows:

//Network 3 Communication parameters 19200,N,8,2, Slave 1#,eight temperature modbus address are 10-17H.The decimal system are 16-23



//Network 4 inverter,9600 E 7 1 ASCII, Slave 2#.Setting frequency address is 2001H,The decimal system is 8193.Run frequency address is 2103H.The decimal system is 8451.



Port = 3, in this example, it indicates the communication port of S01RS. 8193 can be quickly entered, such as put the mouse on the "Write" terminal, then it can directly enter 0x2001.

7. Typical freedom protocol applications:

Serial communication COMM instruction is divided into high-low byte mode and low byte mode. In accordance with the sending and receiving, it also can achieve that only sending、only receiving、 waiting for data receiving after sending commands. For more detail of instructions introduction, you can refer to the software online help. As for freedom communication, the essence is to

understand the communication protocol, the following we will introduce the typical application of COMM instruction.

7.1 Tn=0, Rn>0, only receive data

Only receive data, for example, PLC communicates with the weighing instrument, it is known that the weighing instrument has 485 port, the baud rate is 9600, data format is E 7 1 ASCII, the weighing instrument generally has two modes, query mode and continuous sending mode. Continuous sending mode, it indicates that the instrument intermittently send the current weight to the communication port, so as long as the PLC side is responsible for receiving data.

It is known that the weighing instrument sends 18 bytes of data to the communication port for each time, and the 18 bytes respectively represent the specific meanings as follows:

Byte order	ASCII symbol	Hexadecimal number	Meaning
1~2	OL	4FH 4CH	Overrange
	ST	53H 54H	Static load
	US	55H 53H	Dynamic load
3	,	2CH	Separator
4~5	NT	4EH 54H	Net weight
	GS	47H 53H	Gross weight
6	,	2CH	Separator
7	+	2BH	Positive sign
	-	2DH	Negative sign
8~14	0~9	30H-39H	Data
	" "	20H	Space
	.	2EH	Decimal point
15~16	Kg	4BH 67H	Kilo
	"T"	54H 20H	Ton
17	CR	0DH	Enter
18	LF	0AH	Line feed

Assuming that the net weight of weighing instrument is 1.23Kg, then the weighing instrument will send 18 bytes of data to the PLC according to the above format, the data is in hexadecimal:

53 54 2C 4E 54 2C 2B 20 20 20 31 2E 32 33 4B 67 0D 0A

The above data indicating the weight is 2B 20 20 20 31 2E 32 33, it represents 1.23 Kg. 2B is the "+" sign, 20 is a space, 2E is the decimal point, 30 ~ 39H is the character 0 ~ 9. So we just use the COMM instruction to receive the 18 bytes back, and parse out the weight we want. In this way, we write the COMM instruction below:

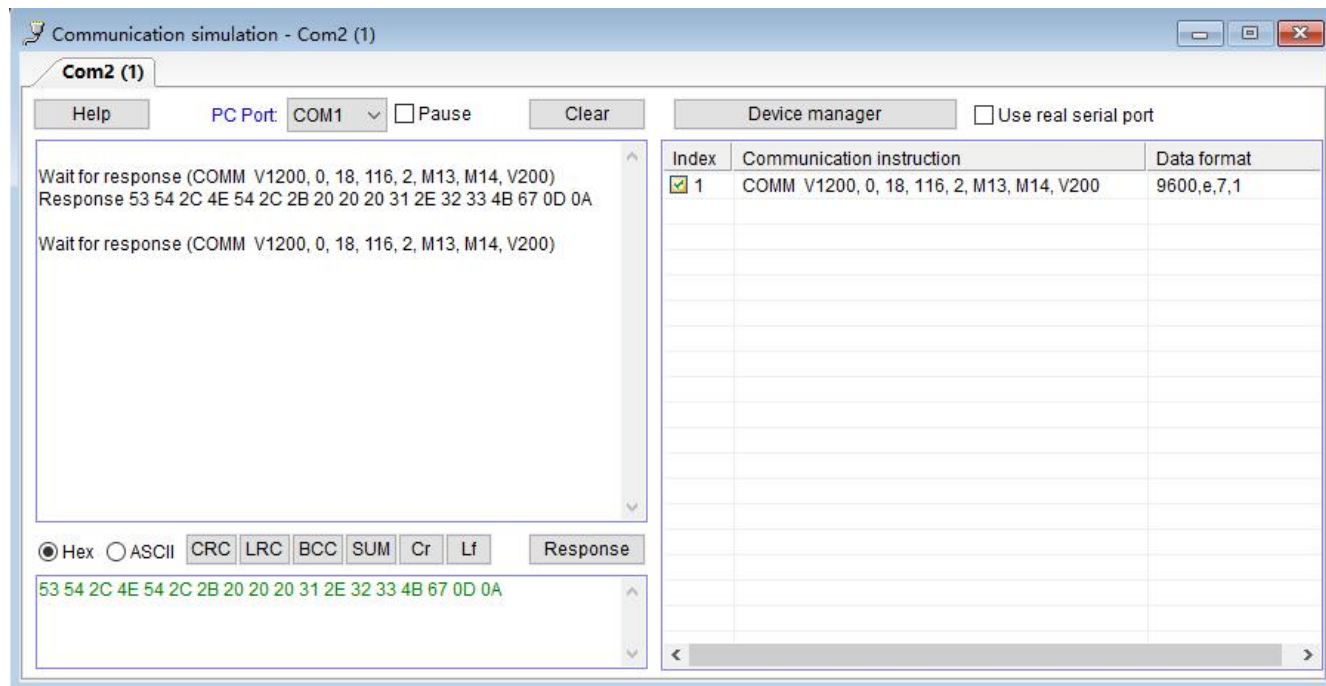


When PLC is running, and COMM instruction is scanned, since Tn = 0, there is no need to send command. Therefore, COMM instruction will be in receiving status according to baud rate and data format defined by protocol, at this moment, when the weighing instrument sends the data of 53 54 2C 4E 54 2C 2B 20 20 20 31 2E 32 33 4B 67 0D 0A, the COMM instruction will receive the data, and store it in the start register of V200, when the data is stored, first stored in the low byte of the register, then stored the high byte of the register, as follows:

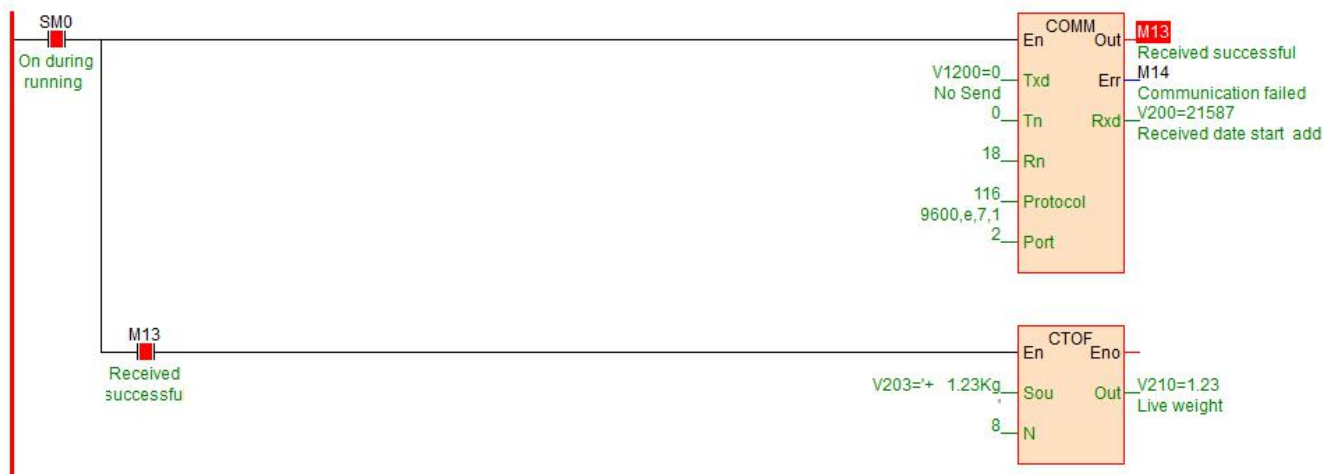
..... 0A0D 674B 3332 2E31 2020 202B 2C54 4E2C 5453
 V208 V207 V206 V205 V204 V203 V202 V201 V200

From the above we can see that we want the data in V203 ~ 206, these four registers, a total of 8 bytes. The next is data analysis,

Haiwell has a very convenient instruction of character to floating point, that is, CTOF. So we write the program as follows, and get the simulation results through communication simulator running:



Click on the "simulation", open the communication simulation in the debug menu, fill in the data in the answering area, click on the "answer" to simulate the communication program, if the serial interface of computer is actually connected to instrumentation, you can check the "use the actual serial port" so that the software can co-simulation with the actual instrument, very convenient. The final simulation program is as follows:



7.2 Tn>0, Rn=0, only send data

For example, when the values of the register V300 in the PLC are respectively different values, then it sends different data contents to the serial port, for example:

V300 = 1, then send the 6 bytes data of V1000 ~ 1002 to the communication port;

V300 = 2, then send the 8 bytes data of V1010 ~ 1013 to the communication port;



7.3 $Tn > 0$, $Rn > 0$, for example, communication with YUDIAN instruments by AIBUS

AI instrument uses hexadecimal data format to represent a variety of instruction codes and data. AI instrument only has two standard communication instruction, one is read instruction, the other is write instruction. As follows:

Read: address code + 52H (82) + parameter code to read + 0 + 0 + check code

Write: address code + 43H (67) + the parameter code to be written + low byte of written data + high byte of written data + check code

Return data: The instrument returns the following 10 bytes of data no matter for writing or reading:

Measured value PV + setting value SV + output value MV and alarm status + parameter value to be read / written + check code

In this case, the instrument address is 1, baud rate is 9600, data format is N 8 1. When reading the current temperature, assume the current temperature is 254.1 °C (2541 = 0x9ED), then send and reply data as follows (why the following data is the content for sending and replying, please refer to the AI instrument manual agreement section):

Send: 81 81 52 00 00 00 53 00

Reply: ED 09 00 00 00 60 00 00 EE 69

In general, read commands are fixed. Haiwell provides a very convenient initialized data table: it is initial register value table, the commands which will be sent are filled in the initial register value table, in this case, the starting address is V1020, the length is 4, a total of 8 bytes, as follows:

..... **00 53 00 00 00 52 81 81**

..... **V1023 V1022 V1021 V1020**

The COMM instruction has two modes, one is high-low byte mode, first, sending the low-byte data of the start address, then, sending the high-byte, and so on. The other is the low-byte mode of COMM.LIB. You can double-click to set the COMM instruction, it only sends the low-byte data of the register. In this example of high-low byte mode, first, send the low-byte 81H of V1020, then send high-byte 81H, send the low-byte 52H of V1021, then send high-byte 00H of V1021 and so on.

The following is the initial register values table of "read AI instrument commands":

Initial register table

Table name: Start component: ☐ Decimal ☒ Hex ☐ Float

Power-off preservation: 1048 Point V1000 - V2047 Length: (0 to 200) ☐ Character

Index	Component	16bits value	32bits value	Component comments
1	V1020	0x8181	0x00528181	
2	V1021	0x0052	0x00000052	
3	V1022	0x0000	0x00530000	
4	V1023	0x0053	0x00000053	

Comments:

Password: Confirm password:

Then according to the regulation of the instrument, we write the COMM instruction, and the simulation is carried out as follows:

Component state table - Status table1

Component	16bits value	32bits value	Component comm
V50	2541	2541	Measurement PV
V51	0x0000	0x60000000	
V52	0x6000	0x00006000	
V53	0x0000	0x69EE0000	
V54	0x69EE	0x000069EE	

COMM En Out

V1020=-32383 Tx d Err M20 read successful

Sending data sta 8 Tn Rxd M21 read failed

10 Rn V50=2541 Measurement PV

120 Protocol

9600,n,8,1 Port

2

Communication simulation - Com2 (1)

Com2 (1)

Help PC Port: COM1 ☐ Pause Device manager ☐ Use real serial port

Send (COMM V1020, 8, 10, 120, 2, M20, M21, V50)
81 81 52 00 00 00 53 00
Wait for response
Response ED 09 00 00 00 60 00 00 EE 69

Send (COMM V1020, 8, 10, 120, 2, M20, M21, V50)
81 81 52 00 00 00 53 00
Wait for response

☒ Hex ☐ ASCII

ED 09 00 00 00 60 00 00 EE 69

Index	Communication instruction	Data format
1	COMM V1020, 8, 10, 120, 2, M20, M21, V50	9600,n,8,1

When PLC is running, COMM instruction is scanned, because $Tn=8 > 0$, COMM instruction will send command to communication port 2 according to the baud rate and data format defined by protocol: 81 81 52 00 00 00 53 00, after sending completed, judge $Rn=10 > 0$, then it turns into the receiving state. After receiving the command, it will reply the data: ED 09 00 00 00 60 00 00 EE 69, the data received by the PLC will be placed in the starting register of V50, first it will be stored in the low byte of V50, and then it will be stored in the high byte of V50, low byte of V51, high byte of V51 and so on, when receiving data of $Rn=10$, OUT terminal is ON, marking the success of this communication

```

..... 69 EE 00 00 60 00 00 00 09 ED
..... V54 V53 V52 V51 V50

```

Received data is arranged as shown above, hexadecimal 09ED is the decimal number 2541, a decimal precision, which is the actual 254.1 °C.

8. The system registers of communication timeout time, communication instruction execution interval, communication port character receiving timeout time and application introduction

8.1 Communication timeout time

The unit is ms, the default is 200, that means 200ms, indicating the time which is waiting for the response and replying data from the slave, after PLC sending commands to communication port. Communication timeout is used for PLC master station, there is no need to use this communication system register when PLC is used as a slave station.

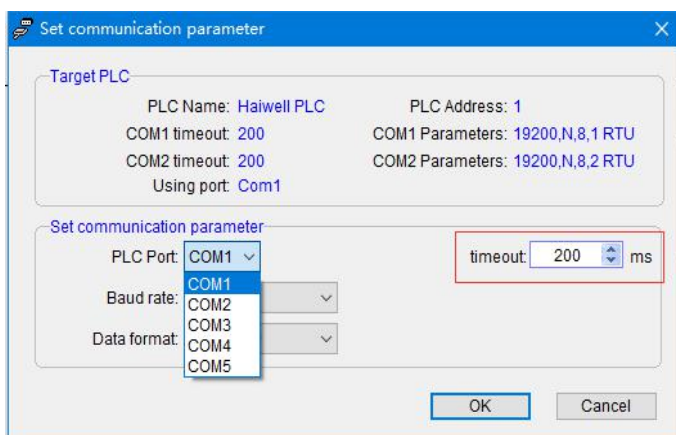
System register	Note	Read/write	Power-off preserve	Default value
SV45	COM1 and Ethernet communication timeout time, unit: ms	R/W	Yes	200
SV47	COM2 communication timeout time, unit: ms	R/W	Yes	200
SV55	COM3 communication timeout time, unit: ms	R/W	Yes	200
SV57	COM4 communication timeout time, unit: ms	R/W	Yes	200
SV59	COM5 communication timeout time, unit: ms	R/W	Yes	200

In general, the value is default, it is generally used in the situation when some slave instruments response data slowly, for example, when slave instrument receives the commands from PLC, it will take 0.5 second to reply the data, at this time, you need to increase the communication timeout time. If you need to increase the communication timeout time, there are two ways for settings:

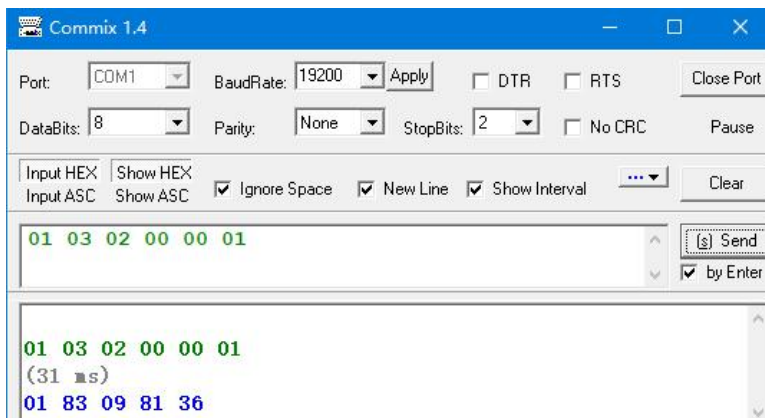
- ① In the PLC program, use MOV assignment instruction to assign values to system registers, such as COM2 communication timeout time setting.



- ② After PLC online, in the menu bar - PLC pull-down option - communication parameters settings-set the communication timeout time.



- ③ Serial debugging tools can record the data's interval time from sending to replying. Therefore, it is recommended that before PLC and instruments communication, you can use the serial port debugging tools to communicate with instruments, so that we can confirm the instrument station number, baud rate, data format, and the Modbus addresses to be read and written, response time and so on. In this way, writing PLC communication instruction will be easier. For example, the following figure of 16ms:



8.2 Communication instruction execution interval

The interval time defaults to 0, PLC will execute next one immediately after one communication instruction has been executed. If some devices are not allowed or can not receive such a fast communication frequency, it need to set communication interval through MOV instruction assignment.

System register	Note	Read/write	Power-off preserve	Default value
SV141	COM1 communication instruction execution interval, unit: ms	R/W	Yes	0
SV833	COM2 communication instruction execution interval, unit: ms	R/W	Yes	0
SV834	COM3 communication instruction execution interval, unit: ms	R/W	Yes	0
SV835	COM4 communication instruction execution interval, unit: ms	R/W	Yes	0
SV836	COM5 communication instruction execution interval, unit: ms	R/W	Yes	0

Note:

- Communication time of S series PLC's all communication ports is set by SV141, CTHN four series can be set separately.
- Communication instruction execution interval is used for PLC master station, there is no need to use this communication system register when PLC is used as a slave station.

8.3 Communication port character receiving timeout time

There will be a situation in the process of communication with instruments, that is, intervals between communication characters of some instruments are not standard, or a frame data is longer and divided into multiple returns, the intervals between character frames are larger, so that the PLC can not receive the complete data frames and the instruction will judge that the communication fails. To this situation, Haiwell conveniently provides a system register used to set the communication port character receiving timeout time. At this time, the character receiving timeout time can be set. Set the communication port character receiving timeout time through MOV instruction assignment.

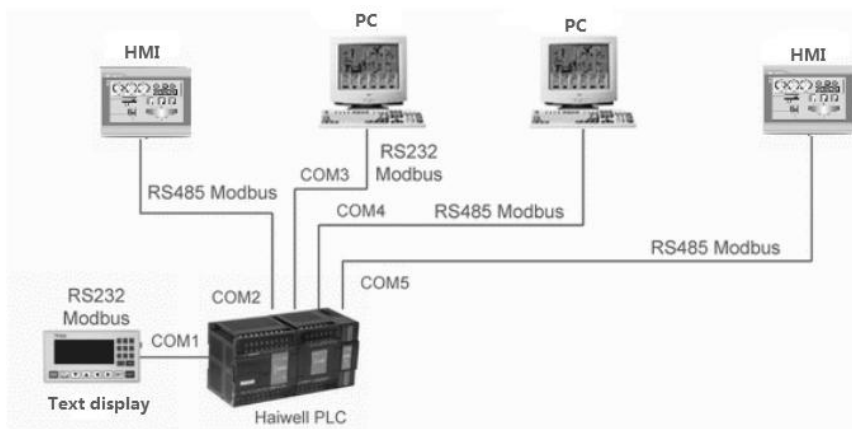
System register	Note	Read/write	Power-off preserve	Default value
SV851	COM1 communication port character receiving timeout time, unit: ms	R/W	Yes	0
SV852	COM2 communication port character receiving timeout time, unit: ms	R/W	Yes	0
SV853	COM3 communication port character receiving timeout time, unit: ms	R/W	Yes	0
SV854	COM4 communication port character receiving timeout time, unit: ms	R/W	Yes	0
SV855	COM5 communication port character receiving timeout time, unit: ms	R/W	Yes	0

Note:

- Four series of CTHN can be set separately, S series does not support setting the communication parameters.
- Communication port character receiving timeout time is used for PLC master station, there is no need to use this communication system register when PLC is used as a slave station.

9. When PLC is used as slave station, there is no need to write any communication program, and supporting a variety of human-machine interfaces and configuration softwares

CPU host which is built-in Ethernet port and two serial ports can be extended to the Ethernet port plus five serial ports, each communication port can be programmed and networked, also it can be used as master or slave. Support the networking ways of 1: N, N: 1, N: N, support a variety of human-machine interfaces and configuration softwares, also it can be networked with any third-party device which has communication functions (such as inverters, instruments, bar-code readers, etc.). The figure below shows the networking way of N:1 when the PLC is used as a slave:



There is no need to write any program when PLC is used as slave, the default parameter is 19200 N 8 2 RTU, station number is 1. For configuration software and touch screen, which has the built-in Haiwell driver can directly select Haiwell driver, if there is no Haiwell driver, you can choose Modicon's Modbus driver. The corresponding Modbus communication code table of Haiwell is as the following figure:

- ① Haiwell PLC bit component table (equivalent to Modbus address type of 0,1, supporting Modbus function code of 1, 2, 5, 15)

Component	Name	Range	Read/Write attribute	Modbus communication address code		Description
				Hexadecimal	Decimal	
X	Digital input	X0~X1023	read	0x0000~0x03FF	0~1023	
Y	Digital output	Y0~Y1023	read/write	0x0600~0x09FF	1536~2559	
M	Auxiliary relay	M0~M12287	read/write	0x0C00~0x3BFF	3072~15359	
T	Timer (coil)	T0~T1023	read/write	0x3C00~0x3FFF	15360~16383	
C	Counter (coil)	C0~C255	read/write	0x4000~0x40FF	16384~16639	
SM	System Status bit	SM0~SM215	all be read/some be wrote	0x4200~0x42D7	16896~17111	
S	Step relay	S0~S2047	read/write	0x7000~0x77FF	28672~30719	

- ② Haiwell PLC register component table (equivalent to Modbus address type of 3, 4, supporting Modbus function code of 3,4,6,16)

Component	Name	Range	Read/Write attribute	Modbus communication address code		Description
				Hexadecimal	Decimal	
CR	Expansion module parameters	CR0~CR255	all be read/some be wrote	0x00~0xFF	0~255	Used when Modbus accesses expansion module parameters
AI	Analog input register	AI0~AI255	read	0x0000~0x00FF	0~255	
AQ	Analog output register	AQ0~AQ255	read/write	0x0100~0x01FF	256~511	
V	Internal data register	V0~V14847	read/write	0x0200~0x3BFF	512~15359	
TV	Current value of timer	TV0~TV1023	read/write	0x3C00~0x3FFF	15360~16383	
CV	Current value of counter	CV0~CV255	read/write	0x4000~0x40FF	16384~16639	Only CV48~CV79 are 32-bit register
SV	System register	SV0~SV900	all be read/some be wrote	0x4400~0x4784	17408~18308	

③ Description:

- Haiwell PLC adopts standard Modbus protocol (supporting RTU and ASCII format) it can communicate with all HMI and configuration softwares that support Modbus protocol.
- Haiwell PLC's Modbus address number starts from 0, some HMI or configuration software starts from 1, if HMI or configuration software's Modbus address starts from 0, then it directly uses the communication address, for example M0 is 0x3072, V0 is 4x0512; If the Modbus address of the HMI or configuration software starts from 1, the component address number needs to be increased by 1, for example, M0 is 0x3073 [3072 + 1] and V0 is 4x0513 [512 + 1]. The first digit of the address is the Modbus protocol component type (0/1 is bit component, 3/4 is register component, type 0/4 means read-write, type 1/3 means read-only), the rest of the digit is the component address number.

10. how to judge the communication failure and program analysis when PLC is used as slave

The communication disconnection alarm program when the master station of PLC communicates with instruments has been introduced in the above example of "Haiwellbus communication example introduction: communication between two Haiwell host PLC", then when the slave station of PLC communicates with the host computer, if communication line is broken, or the host computer parameters have been modified, at this time, as a slave of PLC, how to check the communication disconnection? The following ideas:

The host computer constantly writes M900 as OFF to PLC, the program sets M900 every second. T0 is used for host computer communication failure alarm. If the M900 has a delay of 3 seconds before getting electricity, it means the host computer does not write M900 to be OFF, so that we can judge that the communication between the host computer and the PLC fails.

//Network 1 The host computer reset M900 for to PLC, the program every second set M900



//Network 2 M900 for on 3 seconds on behalf of the host computer and PLC communication failure



11. Check-code calculator usage introduction

If you want to use serial debugging tools or host computer to read the V100 data of Haiwell PLC, what data commands need to be sent?

According to the previous introduction we know that Haiwell PLC is built-in Modbus communication protocol, and Haiwell programming software provides a very convenient tool of check code calculator, the tool can quickly and efficiently calculate the command frame which has been sent. Open Haiwell programming software-Tools-check code calculator, pop-up the following window:

We choose the function code, set the type of components, component number, station number of slave PLC and so on, then click generate, you can get the command: 01 03 02 64 00 01 C4 6D, the command which sent by host computer can help to read the V100 value of PLC .

The check code calculator can also carry on the positional notation conversion at the same time, check the data, calculate the serial data such as CRC, LRC, SUM and other check codes, for communication engineers, this tool is a very convenient programming assistant.

12. The introduction of supportive baud rate, data format and communication instructions when PLC communication port is used as a master/slave station

12.1 PLC communication port used as a master station:

Supported by C / T / H / N series host PLC communication port:

C/T/H/N host PLC communication port	Baud rate	1200、2400、4800、9600、19200、38400、57600、115200
	Data format	N,8,2 RTU、E,8,1 RTU、O,8,1 RTU、N,7,2 ASCII、E,7,1 ASCII、O,7,1 ASCII、N,8, 1 RTU
	Communication instruction	MODR / MODW / HWWR / HWRD / COMM / RCV / XMT
Description: Host PLC' communication port, that is, COM1: RS232 (round-mouth); COM2: RS485 (terminal of A + B-).		

Supported by S01RS communication expansion module:

S01RS communication module	Baud rate	1200、2400、4800、9600、19200、38400、57600
	Data format	N,8,2 RTU、E,8,1 RTU、O,8,1 RTU、N,7,2 ASCII、E,7,1 ASCII、O,7,1 ASCII
	Communication instruction	MODR / MODW / HWWR / HWRD / COMM
Description: 1、S01RS module does not support baud rate115200; 2、If S01RS module uses the MODR/MODW instruction, it will not support the format of N 8 1, so when communicated with the instrument supporting N 8 1 format, you can deal with the problem through the following three methods: Method ①: change instrument to 2-bit stop bits; Method ②: change the calibration method to odd parity check or even parity check; Method ③: send and receive Modbus protocol through COMM instruction. 3, The feature of E/S/B series host PLC communication port (COM1, COM2) and S01RS's is consistent.		

12.2 PLC communication port used as a slave station:

Supported by C / T / H / N series host PLC communication port:

C/T/H/N host PLC communication port	Baud rate	1200、2400、4800、9600、19200、38400、57600、115200
	Data format	N,8,2 RTU、E,8,1 RTU、O,8,1 RTU、N,7,2 ASCII、E,7,1 ASCII、O,7,1 ASCII、N,8, 1 RTU
Description: The default is 19200, N, 8,2 RTU. Generally use the default, without modification.		

Supported by S01RS communication expansion module:

S01RS communication module	Baud rate	1200、2400、4800、9600、19200、38400、57600
	Data format	N,8,2 RTU、E,8,1 RTU、O,8,1 RTU、N,7,2 ASCII、E,7,1 ASCII、O,7,1 ASCII
Description: 1、The default is 19200, N, 8,2 RTU. Generally use the default, without modification. 2、The feature of E/S/B series host PLC communication port (COM1, COM2) and S01RS's is consistent.		

13. PLC communication frequently asked questions

- ① In order to read the register value of instrument, Modbus address is 40001, in Haiwell Modbus, how much to fill in Read of read instruction MODR?

A: Fill in 0. The first digit of the address is the Modbus protocol component type (0/1 is bit component, 3/4 is register component, type 0/4 means read-write, type 1/3 means read-only) , the rest of the digits are component address numbers. Besides, there is no need to add 1 when Haiwell PLC reads Modbus device address, so as for the address 40001, it only fills 0 in Read terminal as Haiwell PLC reading. Similarly, for example, an address is 40387, then as long as fill 386 in Haiwell instructions.

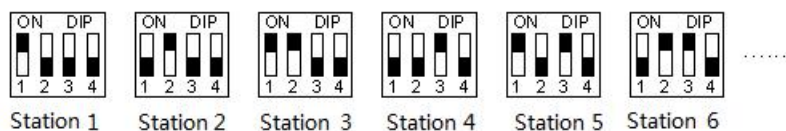


② How to set PLC station number address?

A: Address: 1~254 can be set; PLC address is divided into soft address and hard address, soft address has the highest priority.

Soft address: Through the programming software-PLC-set PLC parameters, check the use of PLC soft address, address range is 1-254;

Hard address: The address is set through the 4-bit DIP switch of module hardware, address range 1-15. Hardware address setting example:



③ Where can we find Haiwell PLC online tutorial? (Haiwell official website-www.haiwell.com)

A: Haiwell official website-Supports-FAQ "PLC online steps described in detail and online problem solution"

④ Where can we find the communication example program between Haiwell PLC and other manufacturers' frequency converter instrument?

A: Haiwell official website-Download-Program example "Haiwell PLC program example.rar"

⑤ Where can we find the communication example program between Haiwell PLC and other configuration software、text or HMI?

A: Haiwell official website-Download-Program example "Communication example between Haiwell PLC and other configuration software, text or HMI.zip"

⑥ Where can we find the communication wiring diagram between Haiwell PLC and other manufacturers' HMI?

A: Haiwell official website-Download-User Manual "communication wiring diagram between Haiwell PLC and other manufacturers' HMI"

Haiwell wireless zigbee communication modules of H01ZB and PC2ZB

application examples

1. Module introduction

ZigBee technology is a short-range, low-power wireless communication technology. Its biggest characteristic is low-power consumption and can be networked, especially the network function with routing, theoretically the communication coverage area of ZigBee can be unlimited expansion. Haiwell has two models for zigbee module, namely PC2ZB and H01ZB.

PC2ZB: It is a type of wireless communication expansion module that used on PC terminal such as computer, HMI, text, etc. Through this module, HaiwellHappy can carry out the operations of online and read-write in wireless mode with the host PLC which is extended with H01ZB.

H01ZB: It is an expansion module for wireless communication between host PLCs. Host PLCs that have expanded this module can use the instructions of MODR/ MODW/HWWR/HWRD to achieve read and write communications in wireless mode. One host PLC can be expanded up to 3 communication ports (the type of expansion module model can be H01ZB or S01RS and other communication expansion modules). H01ZB can only be connected to PLC by parallel port. The communication between PLC and host computer is generally a combination of PC2ZB + H01ZB. When host PLC communicates with host computer wirelessly, it needs to hang a H01ZB module behind each host PLC.

2. Module power supply

H01ZB is hung behind the host PLC through parallel port, and powered by parallel port, no external power supply is required. PC2ZB, USB interface, 5V power supply.

3. Applicable object

H01ZB: N, H, T, S series PLC can be expanded with H01ZB modules, and they can be expanded up to 3 H01ZB modules.

PC2ZB: Built-in USB, 232,485 interfaces, it can access the computer, touch screen, text and so on.

4. Hardware configuration and communication port number

When the communication module is expanded, the communication port which is the closest to the host PLC is COM3, next is COM4, last is COM5. At this time, the position of communication modules can be random. The communication port number can be viewed in the hardware configuration as shown in the figure below:

Index	Module type	X Component	Y Component	AI Component	AQ Component	Other	Description
0	T16S2T/P(-e)	X0 - X7	Y0 - Y7			COM1-2 HSC0-1 PLS0-1	CPU module 8*DI 8*DO transistor AC220V power supply 2 channel 200KHz pulse input 2 channel 200KHz pulse output
1	H01ZB					Port3	ZigBee wireless communication support Modbus/Haiwellbus/free protocol Master/Slave
2	H40XDR	X8 - X27	Y8 - Y27				Digital I/O module 20*DI 20*DO relay, built-in RS485 communication ports support remote
3	S01RS					Port4	1 serial port RS232/RS485 support Modbus/Haiwellbus/free protocol Master/Slave
4	H04DT			AI0 - AI3			Analog input module 4 road DS18xx series temperature sensor
5	H01ZB					Port5	ZigBee wireless communication support Modbus/Haiwellbus/free protocol Master/Slave

It can be seen from this configuration that both H01ZB and S01RS are communication expansion modules, which will occupy the number of communication modules, and the total number of communication modules can be expanded up to three.

5. Networking

5.1 Networking steps of PC2ZB+H01ZB

- ① Connect PC2ZB module to PC-side through the USB/RS232/RS485 mode. Among them, when connected through RS232/RS485, USB port still need to be connected, its role is to provide operating voltage of +5V.
- ② Configure the address of host PLC (through the host PLC' DIP switch), so that the address value is only in the network.
- ③ Select any H01ZB or PC2ZB module in the network to act as the Zigbee coordinator: The method is to dial the 6th bit of the DIP switch to "ON". While the 6th bit of the DIP switch from other modules in the same network is dialled to "OFF" as a router. See the sixth point of schematic diagram and introduction.
- ④ Dial the DIP switches' 1~5 bit of all H01ZB and PC2ZB modules to the same value in the network (used as the network

PANID, ranging from 1 to 31). When users need to set up a number of different networks, it depends on the DIP switch values to distinguish.

- ⑤ After power on, the module will automatically search network. The COM LED of the module blinks periodically when the module does not establish / join in the network; the COM LED stays on when the module establishes / joins in the network.
- ⑥ In this way, it can use HaiwellHappy of PC to carry out online and read-write operations for host PLC.

5.2 Networking steps between host PLCs by connecting with H01ZB module

- ① Hang the H01ZB module on the expansion slot of host PLC(or the expansion slots of other modules hung behind host PLC).
- ② Configure the address of host PLC (through the host PLC' DIP switch), so that the address value is only in the network.
- ③ Select any H01ZB module in the network to act as the Zigbee coordinator: The method is to dial the 6th bit of the DIP switch to "ON". While the 6th bit of the DIP switch from other modules in the same network is dialled to "OFF" as a router. See the section 7 of schematic diagram and introduction.
- ④ Dial the DIP switches' 1~5 bit of all H01ZB modules to the same value in the network (used as the network PANID, ranging from 1 to 31). When users need to set up a number of different networks, it depends on the DIP switch values to distinguish.
- ⑤ After power on, the module will automatically search network. The COM LED of the module blinks periodically when the module does not establish / join in the network; the COM LED stays on when the module establishes / joins in the network.
- ⑥ Using the communication instructions (MODR / MODW) of host PLC to read and write other host PLCs within the network.

6. DIP switch settings

As shown in the figure below, Figure 6-1 shows the 6-bit dial codes of PC2ZB. Figure 6-2 shows the 8-bit dial codes of H01ZB, the effective number of bits is 1 ~ 6, 7 and 8 is reserved. Therefore, the following will introduce the module of dial codes 1~6 function.

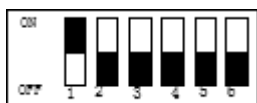


Figure 6-1



Figure 6-2

The first five bit 1~5 of DIP switch are used for setting network ID. The sixth bit of DIP switch is coordinator/router switch, when dialing ON, this module is coordinator, the other modules in the same network can only be OFF as router.

When the user wants to set the network ID, the modification method is as follows: The black part in the figure shows the position of DIP switch. When one of dial codes is set to ON, indicating the bit is set to 1, and when set to OFF, the bit is set to 0. The 1st bit in the above figure is ON, while the 2nd~5th bits are OFF, they are represented in binary system when reflected in the ID number: The 1st bit of DIP switch indicates the 0 bit in binary (b0) and the 5th bit of DIP switch indicates the 4th bit in binary (b4), thus, the first five bits of DIP switch can be used to indicate the number of binary from 00000~11111, converting binary number to decimal number, which can indicate network ID number. The position of the DIP switch as shown in the above figure means 00001, that is, the decimal number is 1, indicating that the network ID number is 1; the valid network ID is 1 to 31.

7. Notes

- ① When networking, it should carry out step by step in strict accordance with the above 5th point.
- ② Only one coordinator can be used in a network, when the module whose the 6th bit of DIP switch is set to "ON" as the coordinator, other modules must keep the 6th bit of DIP switch in the "OFF" state (at this time, the network identity is a router) .
- ③ Support the read-write and access to each other for multiple host PLCs within the same network, but when there are two or more than two host PLCs at the same time visiting the same host PLC, it will cause abnormal communication.

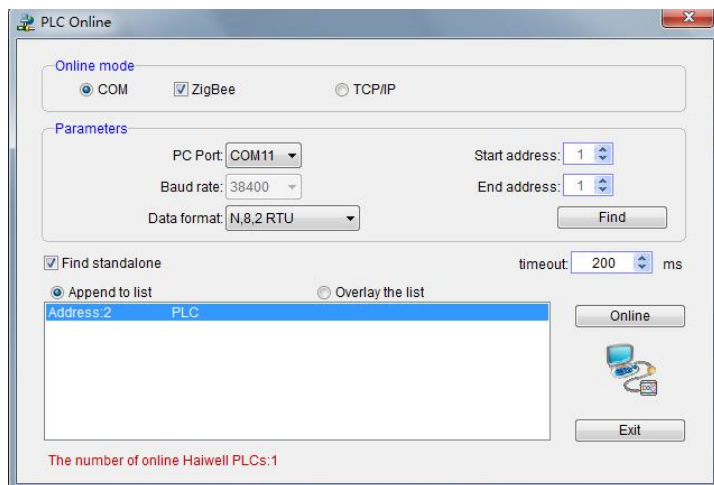
8. Driver download and installation of PC2ZB and PC communication settings

- ① PC2ZB module is connected with PC through USB / RS232 / RS485, when connecting with PC via USB, USB provides the power for the module to work and generates a serial port. When generating the serial port, it needs to install the driver.
- ② Driver download: You can visit Haiwell official website: www.haiwell.com - Software - "Haiwell PC2ZB communication module driver. rar".

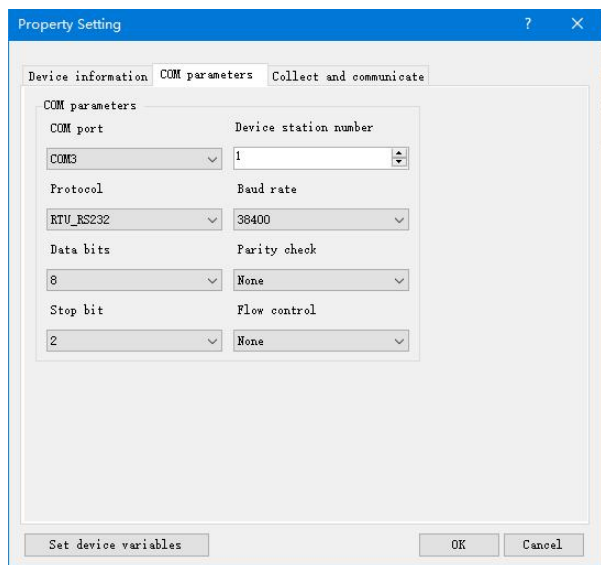


After the driver is installed, if insert PC2ZB, it will have one more COM port, when you want to achieve wireless Zigbee communication through the PC configuration or programming software, just select the new COM port.

When online through the PC2ZB, it needs to select Zigbee, the default baud rate is 38400, the data format is N 8 2RTU.



Therefore, for example, when PC configuration wants to achieve wireless communication via PC2ZB , it needs to set according to the default baud rate 38400 and data format N 8 2RTU. For example, in order to set COM port parameters on Haiwell SCADA, it is necessary to set the baud rate to 38400.



9. Communication between PLCs through H01ZB module

For wireless Zigbee communication between PLCs, you can use MODR / MODW / HWWR / HWRD instructions to achieve wireless read and write communication. The same as the wired 485' communication program, support for Modbus, Haiwellbus. Please refer to "Haiwell communication expansion module S01RS application example", here not repeat them.