SAVCH Programmable Logic Controller

Communication Module

User Manual & Application Case

创无限 Innovate for more Win forever 赢久远
SAVCH

User Manual

Contents

Chapter 1 User Manual

1.1 Products receiving

All products have been performed with strict test and inspection. After receiving the inverters, the following checks shall be performed.

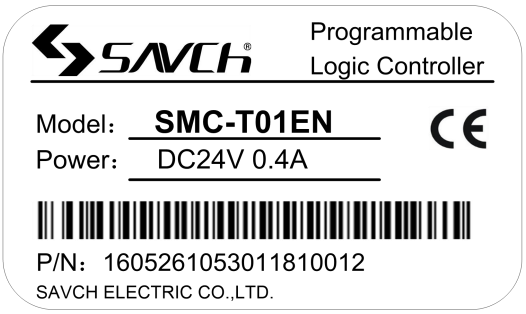
●To check that SAVCH inverter, an instruction manual is inside of the package

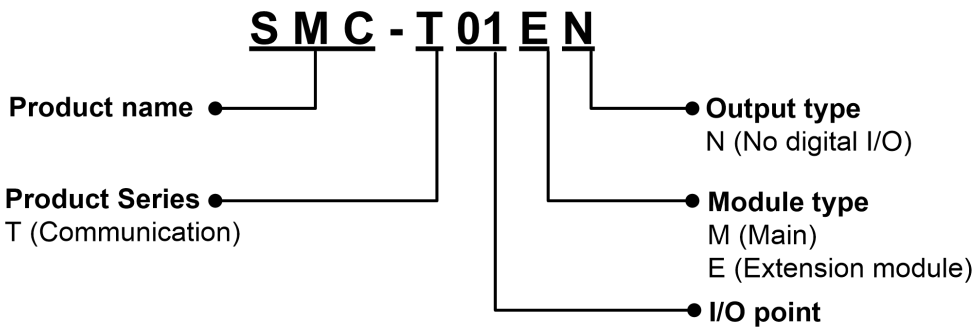
●To check whether model number correspond with model your purchase order.

●To check whether there are damaged parts during transportation and delivering. If there are, do not connect with power supply.

If any of the above checkpoints are not satisfactory, contact your SAVCH ELECTRIC representative for a quick resolution.

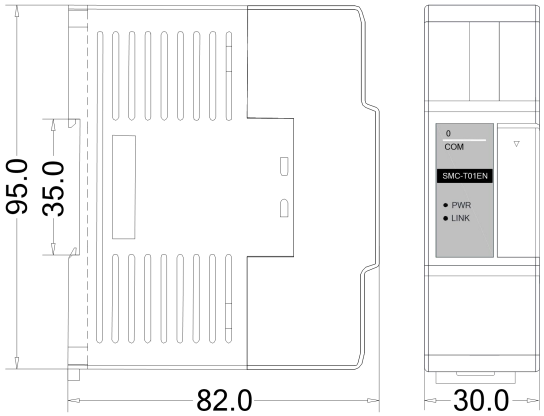
1.2 Model description



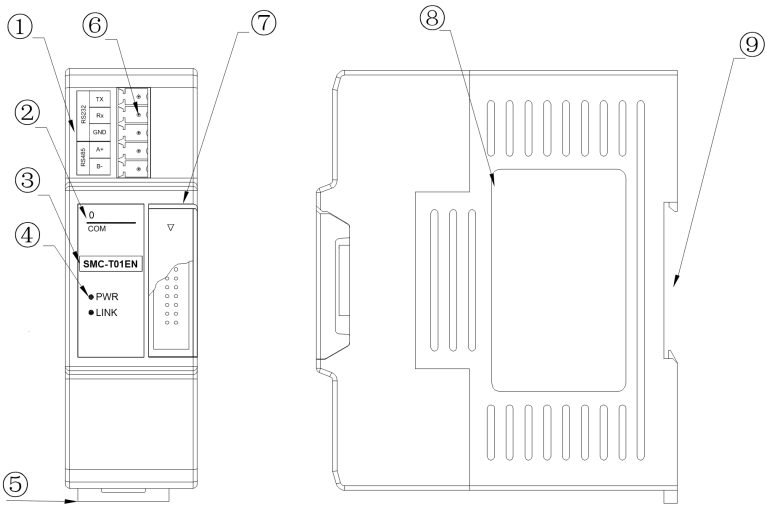


1.3 Product model list & dimensions

Model	Power Consumption	Dimension
SMC-T01EN	0.4VA	30×95×82mm



1.4 Front / side view



1. Terminal definition	6. Pluggable terminal
2. Communication status indicator	7. Module expansion port
3. Model	8. Module nameplate
4. PWR:Power indicator, LINK: Module communication indicator	9. 35mm DIN rail
5. DIN rail mounting slot	

1.5 Indicator description

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

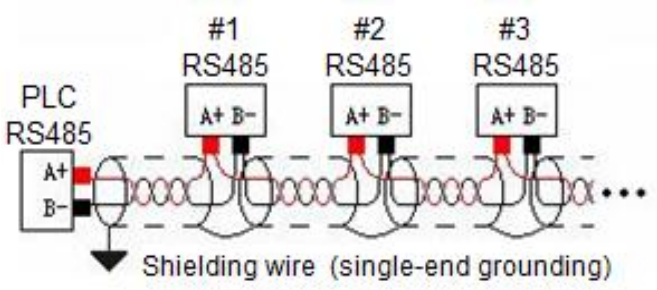
1.6 Environmental specification for product

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

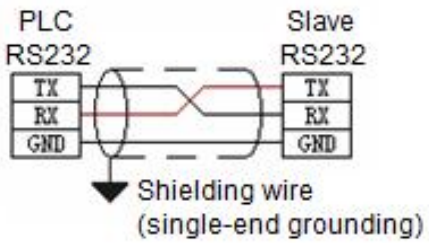
1.7 Module terminal wiring diagram

RS232	Tx
	Rx
	GND
RS485	A+
	B-

1.8 Wiring diagram (RS485&RS232)



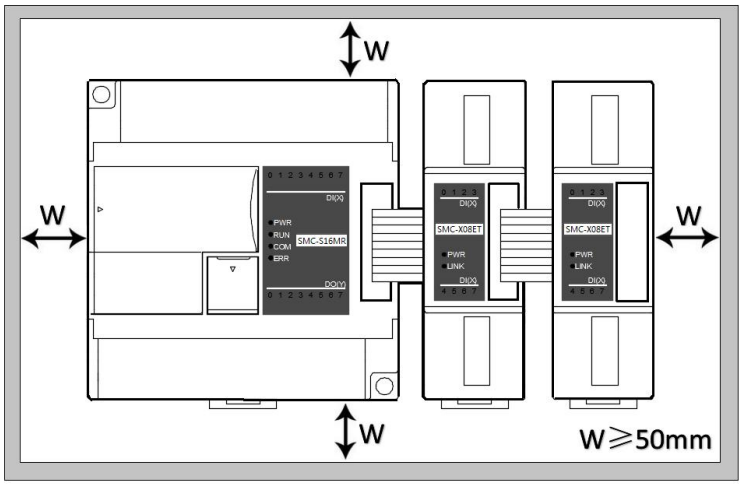
RS485 Wiring diagram



RS232 Wiring diagram

1.9 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.



Chapter 2 Application Case

2.1 Module power supply

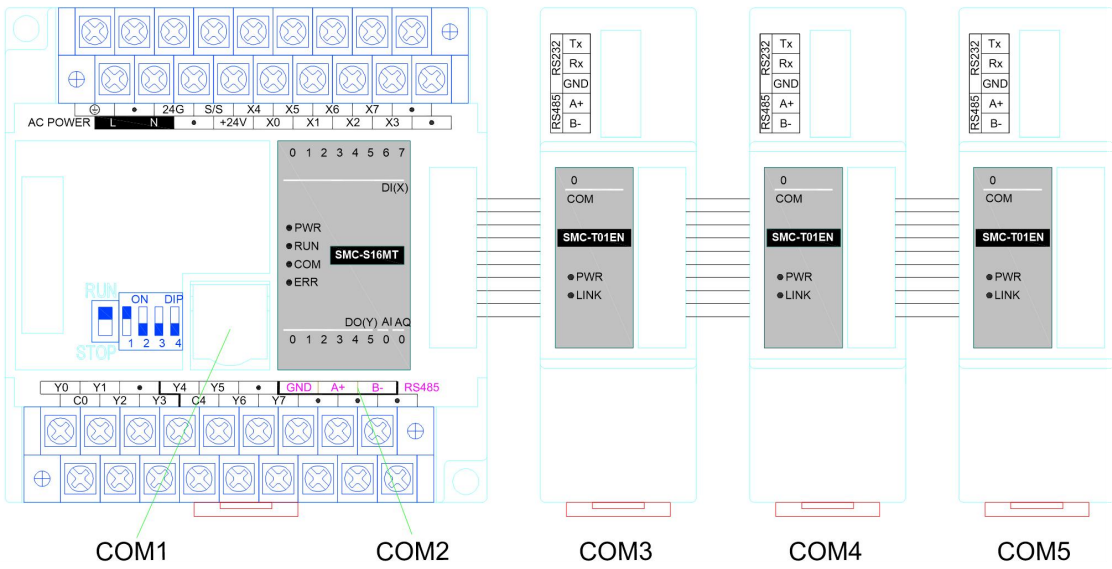
Module is directly hung behind the host PLC by parallel port and powered by parallel port, no need to take external power supply.

2.2 Applicable host PLC

S/H/M series PLC can be expanded with communication modules, the maximum can be expanded with 3 SMC-T01EN modules.

2.3 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 SMC-T01EN not only can be used as 232, but also can be used as 485, no program definition required. It depends on the external wiring, SMC-T01EN 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.

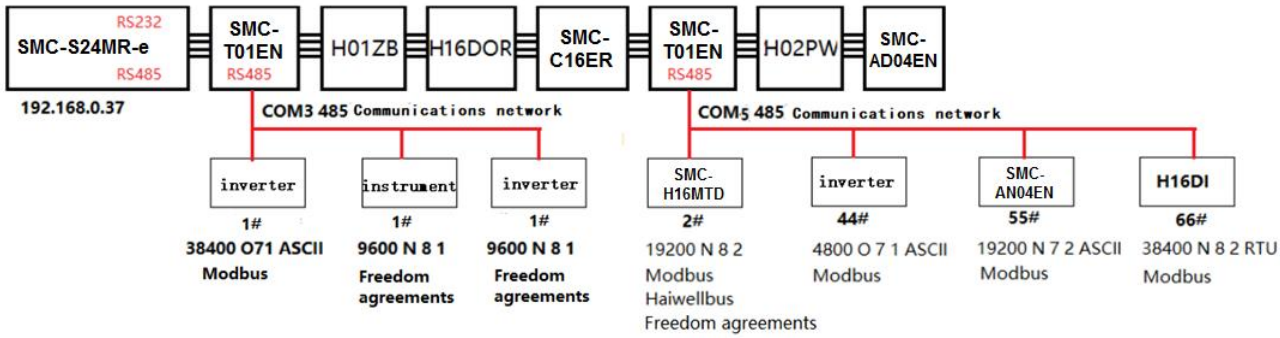


When the communication module is expanded, the communication port closest to the host PLC is COM3, next is COM4 and last is COM5. The position of communication module can be random, and the communication port number can be viewed in the hardware configuration, as shown in the figure below:

Index	Module type	X Component	Y Component	AI Component	AQ Component	Other	Description
0	SMC-S16MT(-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 c
1	SMC-T01EN					Port3	1 serial port RS232/RS485 support Modbus/Haiwellbus/free protocol Master/Slave
2	SMC-AD04EN			AI0 - AI3			Analog input module 4*AI analog input. built-in RS485 communication ports support remot
3	SMC-Y08ET		Y8-Y15				Digital output module 16*DO transistor. built-in RS485 communication ports support remo
4	SMC-T01EN					Port4	1 serial port RS232/RS485 support Modbus/Haiwellbus/free protocol Master/Slave
5	SMC-T01EN					Port5	1 serial port RS232/RS485 support Modbus/Haiwellbus/free protocol Master/Slave

2.4 Communication protocol

It is built-in Modbus RTU / ASCII protocol, freedom communication protocol and SAVCH bus high-speed communication protocol of SAVCH 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.



2.5 SAVCH bus communication example description: Communication between two SAVCH 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.

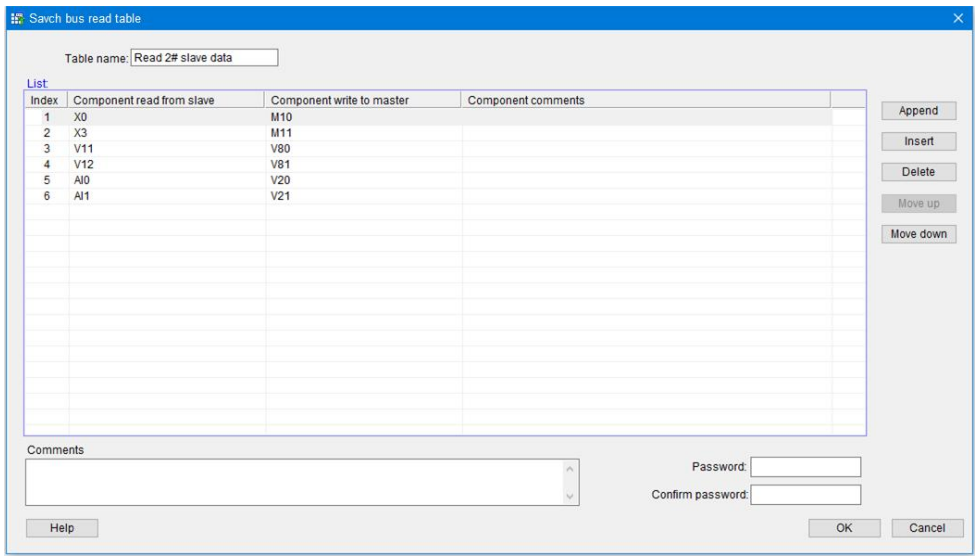
① Master reads 2 # slave data: for example, read X0 of slave into master M0; read X3 of slave into master M11, etc., as follows:

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

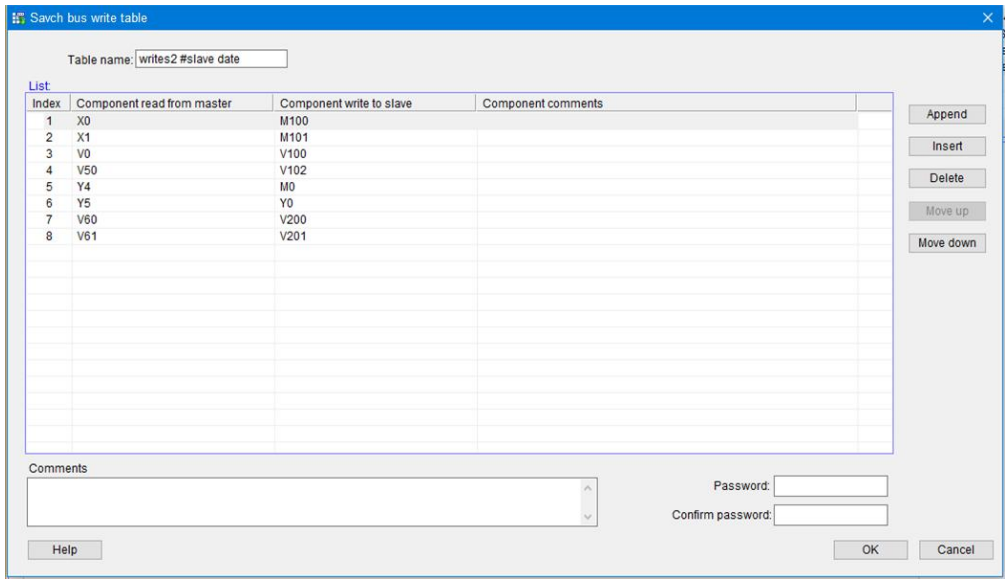
② Master writes 2 # slave data: for example, write V0 of master into slave V100; write V50 of master into slave V102, etc., as follows:

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 SAVCH PLC programming software, set up SAVCH bus 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:



In the SAVCH PLC programming software, set up SAVCH bus write communication table, as follows:



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". The following is an example of a disconnection alarm program for slave communication:

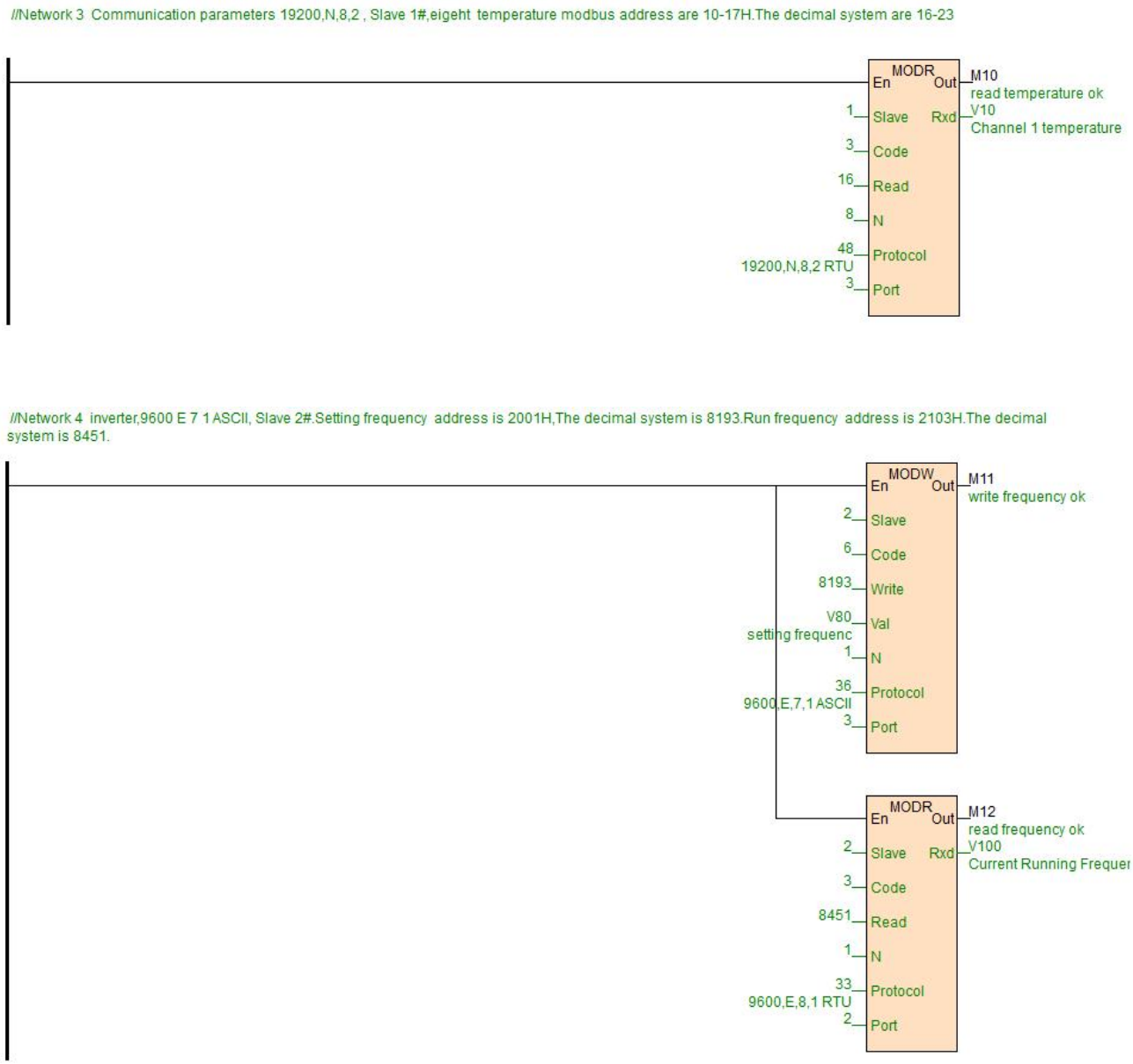
Communication disconnection time can be set according to the actual, the above when M5 is more than 5 seconds in OFF state, T0 coil will indicates communication disconnection alarm.

2.6 Modbus communication example introduction:

In this example, the host PLC is expanded with a communication expansion module SMC-T01EN, communication port is 3, the communication port is connected to a SAVCH thermal resistance module SMC-TR08EN and a inverter:

① SMC-TR08EN 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:



Port = 3, in this example, it indicates the communication port of SMC-T01EN. 8193 can be quickly entered, such as put the mouse on the "Write" terminal, then it can directly enter 0x2001.

2.7 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.

2.7.1 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：

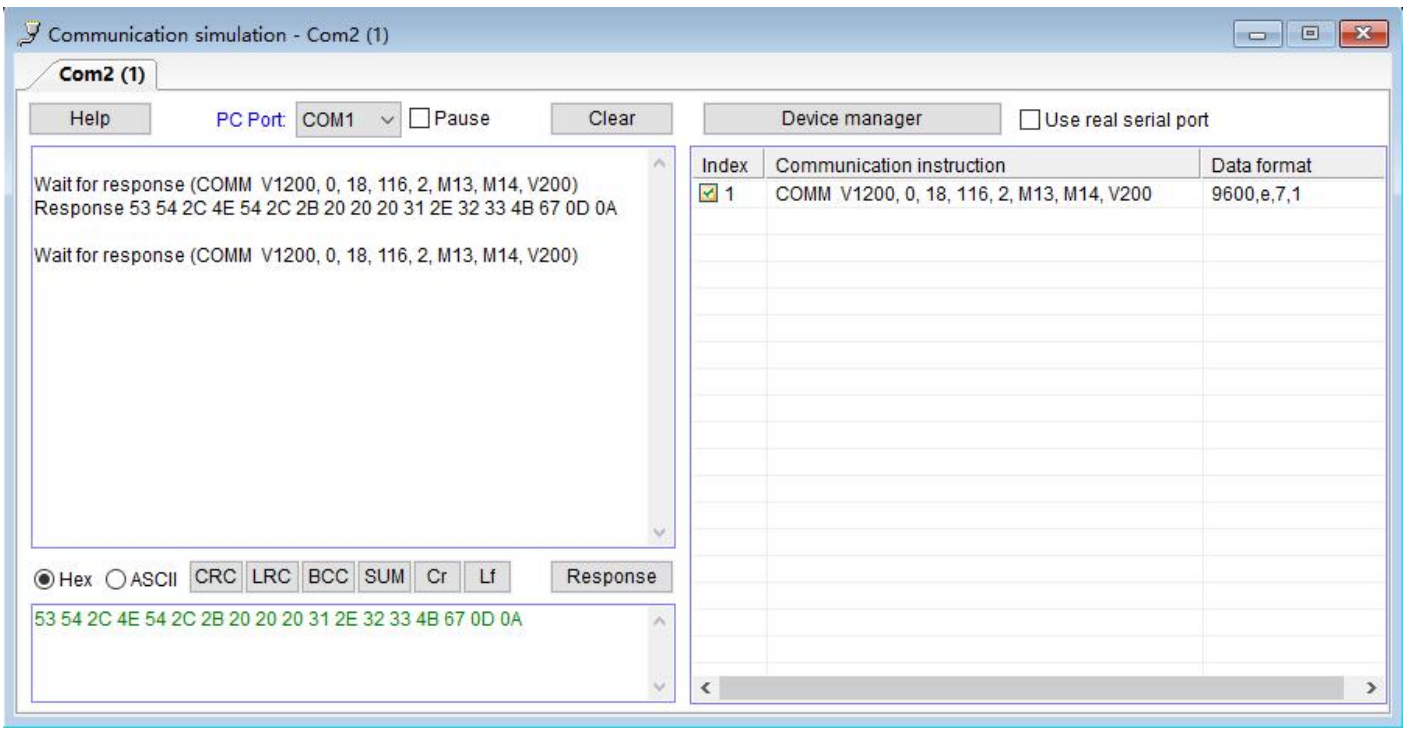
53 54 2C 4E 54 2C 2B 20 20 20 31 2E 32 33 4B 67 0D 0A

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:

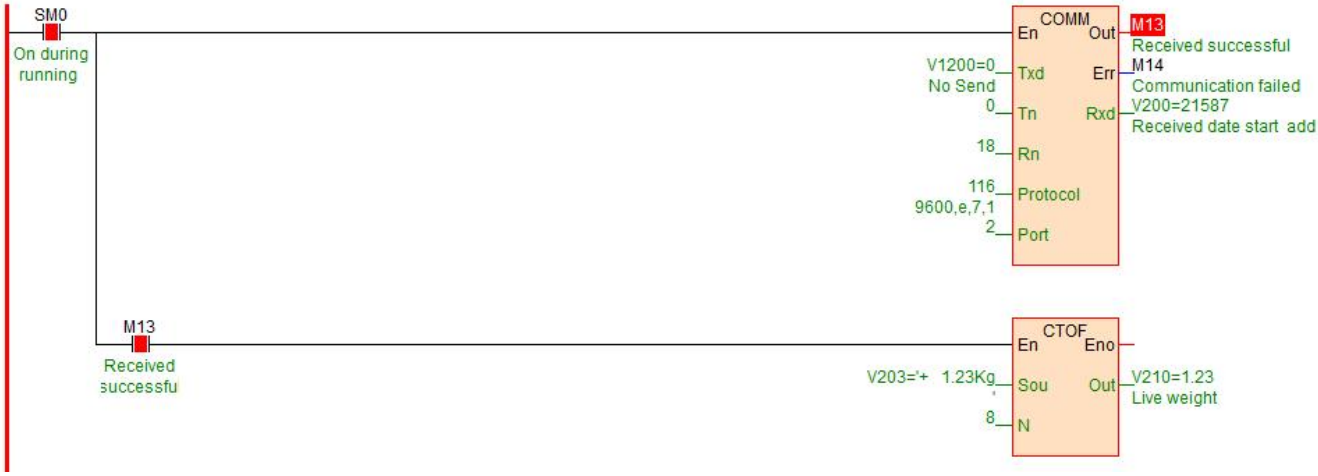
.....	0A0D	674B	3332	2E31	2020	202B	2C54	4E2C	5453
.....	V208	V207	V206	V205	V204	V203	V202	V201	V200

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, SAVCH 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:

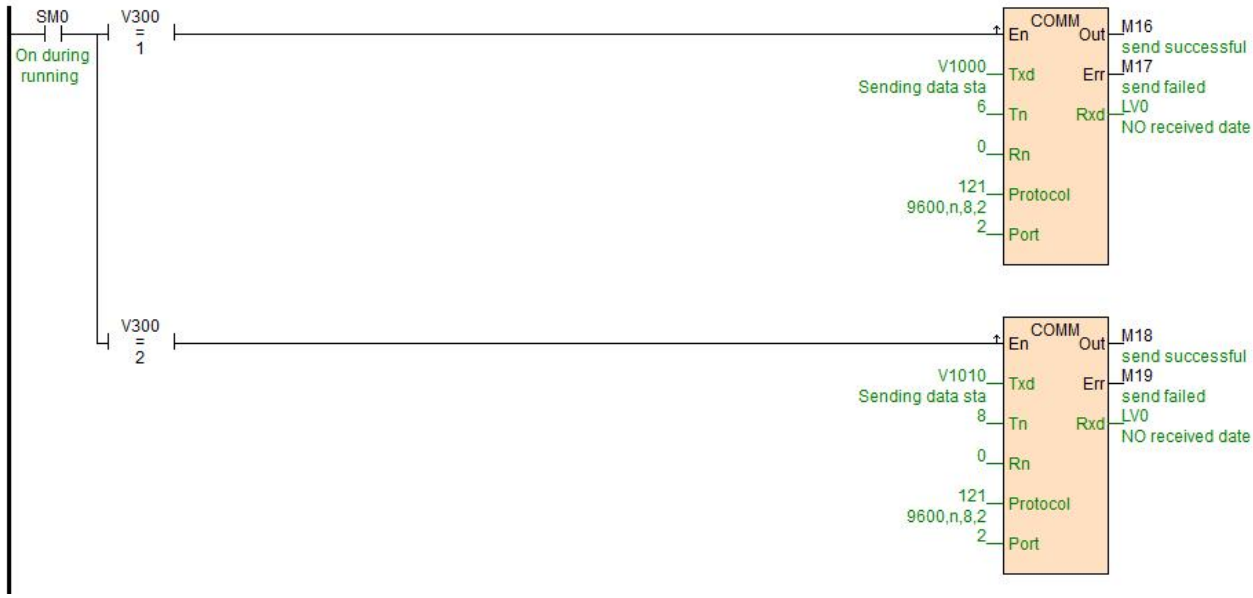


2.7.2 Tn＞0,Rn=0,only send data

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;



2.7.3 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 ℃ (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

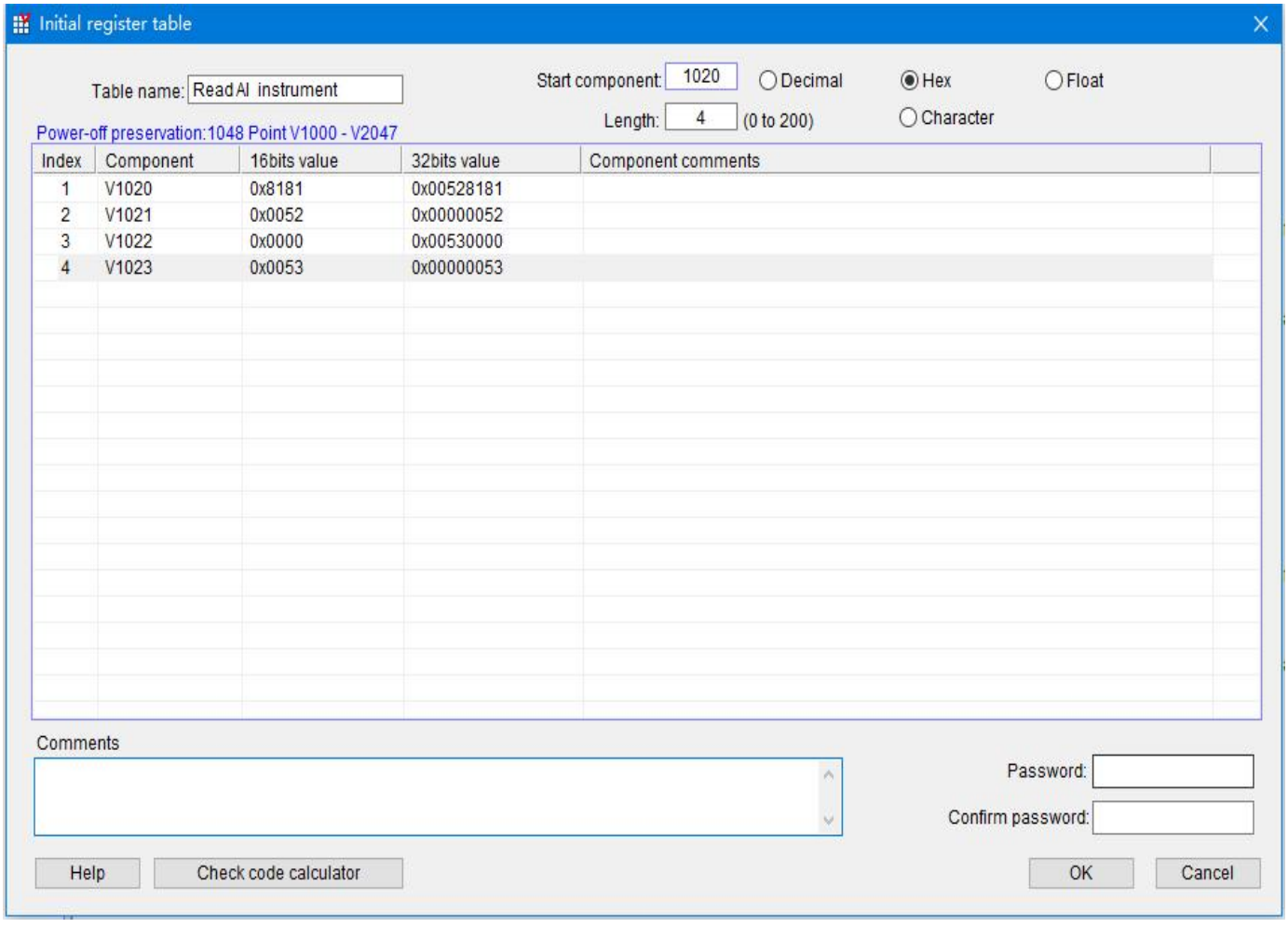
In general, read commands are fixed. SAVCH provides a very convenient initialized data table: it is initial register value table, the commands which will be sent are filled in the initial register value table, in this case, the starting address is V1020, the length is 4, a total of 8 bytes, as follows:

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

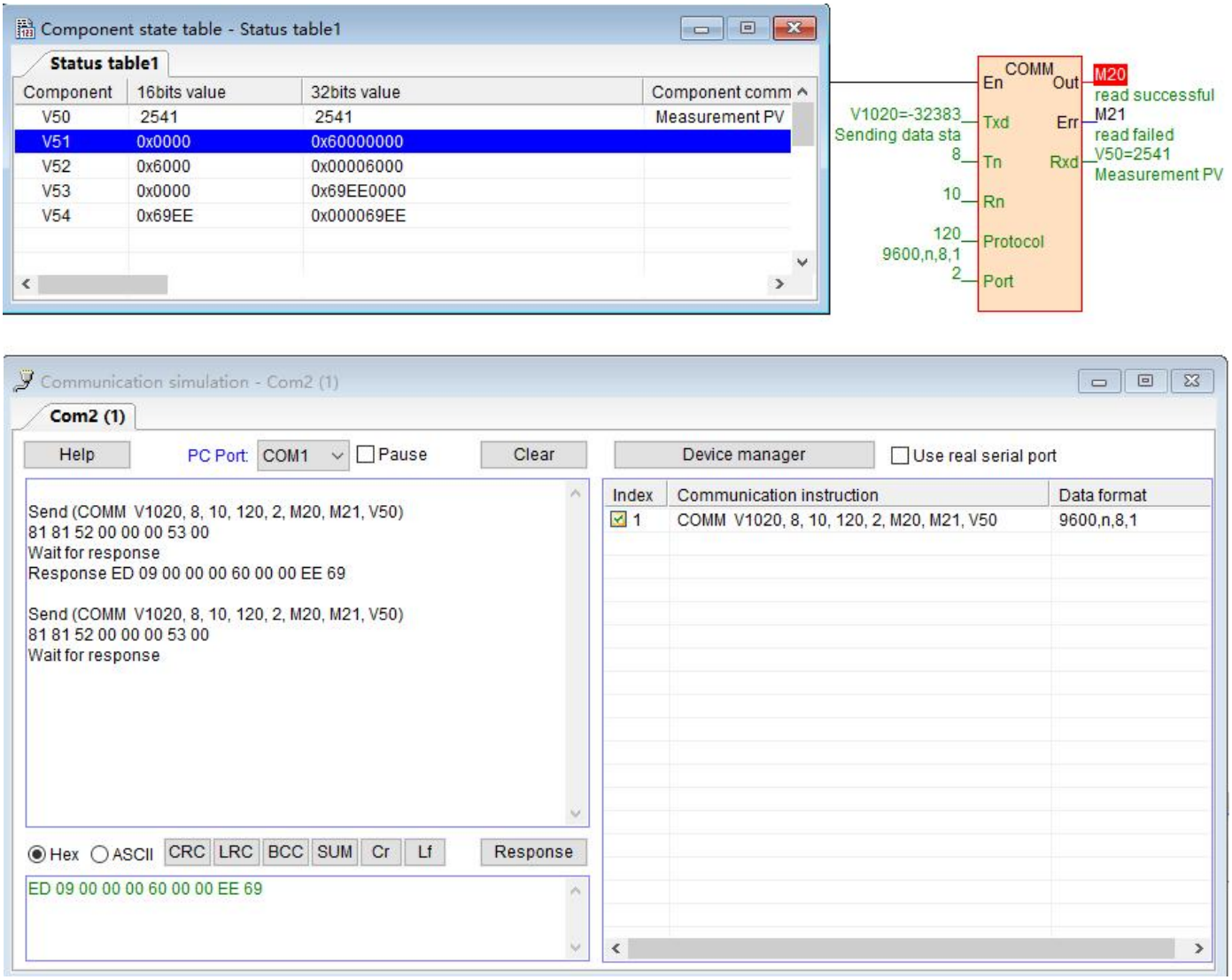
..... V1023 V1022 V1021 V1020

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":



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



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 ℃.

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

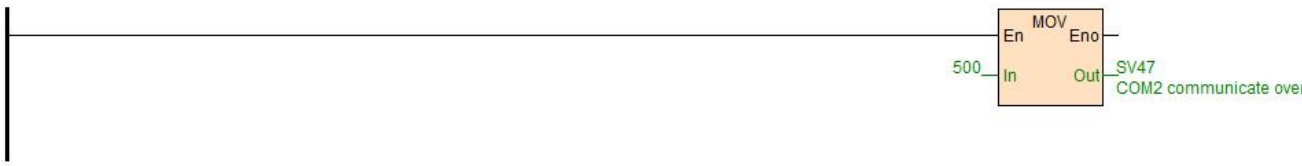
2.8.1 Communication timeout time

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.

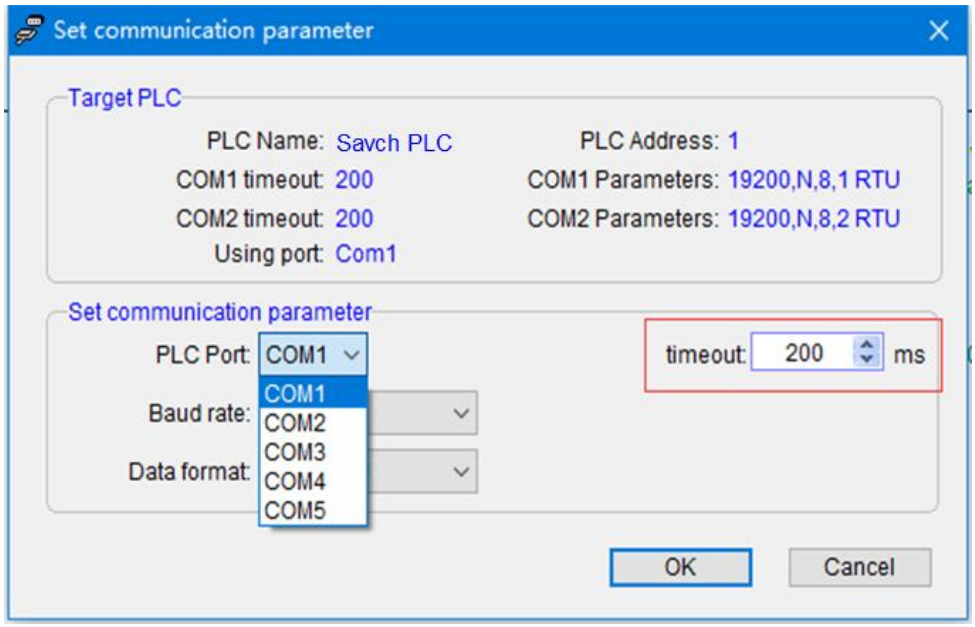
System register	Note	Read/write	Power-off preserve	Default value
SV45	COM1 and Ethernet communication timeout time, unit: ms	R/W	Yes	200
SV47	COM2 communication timeout time, unit: ms	R/W	Yes	200
SV55	COM3 communication timeout time, unit: ms	R/W	Yes	200
SV57	COM4 communication timeout time, unit: ms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

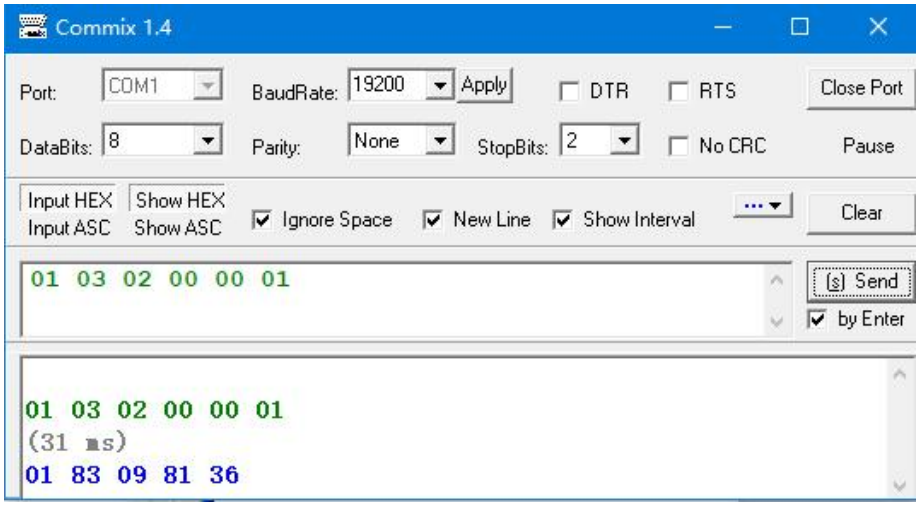
① In the PLC program, use MOV assignment instruction to assign values to system registers, such as COM2 communication timeout time setting.



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:



2.8.2 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 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.

2.8.3 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, SAVCH 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.

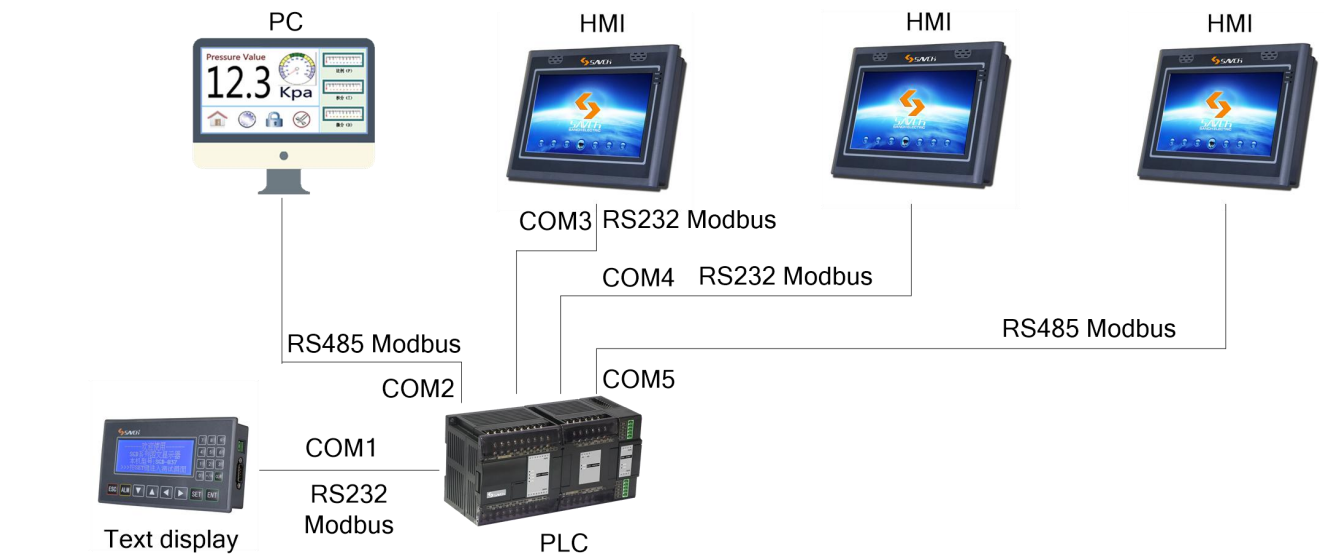
System register	Note	Read/write	Power-off preserve	Default value
SV851	COM1 communication port character receiving timeout time, unit: ms	R/W	Yes	0
SV852	COM2 communication port character receiving timeout time, unit: ms	R/W	Yes	0
SV853	COM3 communication port character receiving timeout time, unit: 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 E/H/M can be set separately.
- 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.

2.9 When PLC is used as slave station, there is no need to 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 SAVCH driver can directly select SAVCH driver, if there is no SAVCH driver, you can choose Modicon's Modbus driver. The corresponding Modbus communication code table of SAVCH is as the following figure:

① SAVCH 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 SAVCH 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:

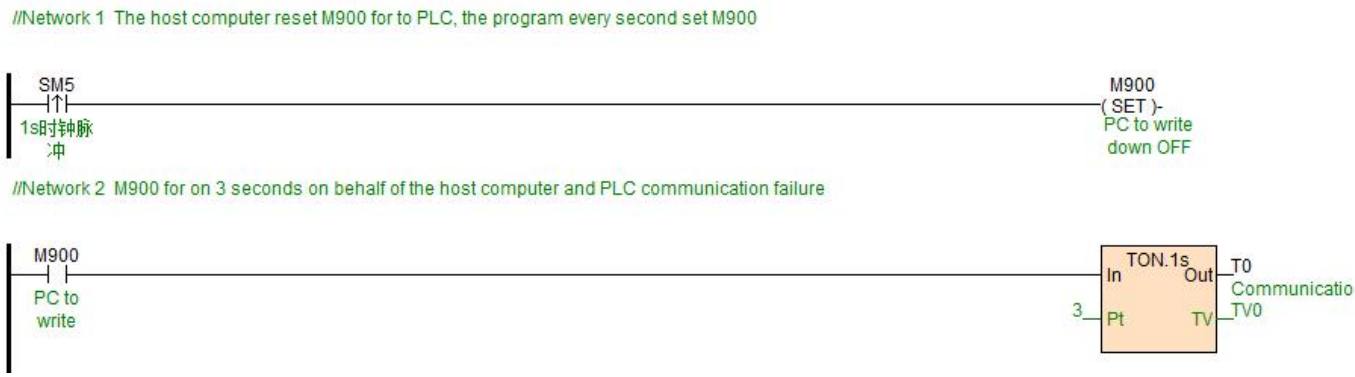
SAVCH PLC adopts standard Modbus protocol (supporting RTU and ASCII format) it can communicate with all HMI and configuration softwares that support Modbus protocol.

SAVCH 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.

2.10 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 "SAVCH bus communication example introduction: communication between two SAVCH 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:

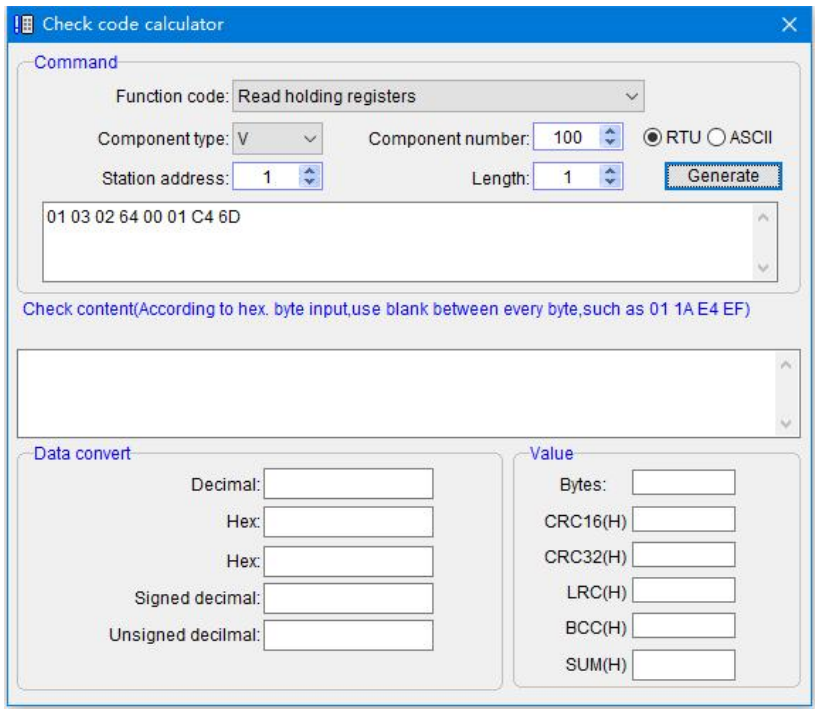
The host computer constantly writes M900 as OFF to PLC, the program sets M900 every second. T0 is used for host computer communication failure alarm. If the M900 has a delay of 3 seconds before getting electricity, it means the host computer does not write M900 to be OFF, so that we can judge that the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



2.11 Check-code calculator usage introduction

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

According to the previous introduction we know that SAVCH PLC is built-in Modbus communication protocol, and SAVCH 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 SAVCH 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.

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

2.12.1 PLC communication port used as a master station:

Supported by S/H/M series host PLC communication port:

S/H/M 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 SMC-T01EN communication expansion module:

SMC-T01EN 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. SMC-T01EN module does not support baud rate115200; 2. If SMC-T01EN 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 S/H/M series host PLC communication port (COM1, COM2) and SMC-T01EN's is consistent.		

2.12.2 PLC communication port used as a slave station:

Supported by S/H/M series host PLC communication port:

S/H/M 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 SMC-T01EN communication expansion module:

SMC-T01EN 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 S/H/M series host PLC communication port (COM1, COM2) and SMC-T01EN's is consistent.		

2.13 PLC communication frequently asked questions

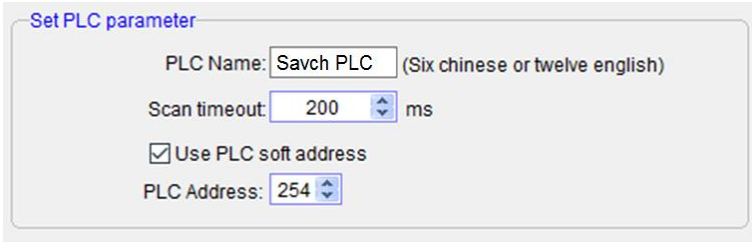
① In order to read the register value of instrument, Modbus address is 40001, in SAVCH Modbus, how much to fill in Read of read instruction MODR?

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 SAVCH PLC reads Modbus device address, so as for the address 40001, it only fills 0 in Read terminal as SAVCH PLC reading. Similarly, for example, an address is 40387, then as long as fill 386 in SAVCH instructions.

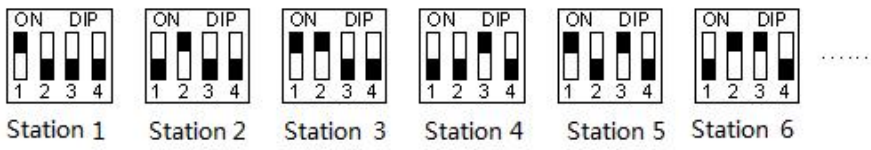
② How to set PLC station number address?

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 SAVCH PLC online tutorial? (SAVCH official website-www.SAVCH.com)

A: SAVCH official website-Supports-FAQ “’PLC online steps described in detail and online problem solution”

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

A: SAVCH official website-Download-Program example “SAVCH PLC program example.rar”

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

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

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

A: SAVCH official website-Download-User Manual “communication wiring diagram between SAVCH PLC and other manufacturers' HMI”

■ Innovate for more | Win forever

■ Industry intelligence | Energy saving | Green power

Qualification

Designed by Taiwan savch electric

Received ISO9001 and CE certificate