

Ethernet Multi-channel Analog Signal Isolated Data Acquisition Transmitter

The Internet of Things/Local Area Network Analog Isolated Data Acquisition Transmitter
SY AD 08-RJ45

Features:

- Directly convert 4-20mA and other analog data into communication digital data for Ethernet and Can bus.
- Support 10M Ethernet RJ45 interface, support Modbus TCP communication protocol.
- Support RS 485/RS 232 serial ports extended, support Modbus RTU protocol.
- Measuring precision is more than 0.05%, do calibration on the precision of modules by computer is available.
- 8-channel analog input with common ground, 3000VDC isolation between I/O terminal and the system.
- Wide range auxiliary power supply wide range: 8 - 50VDC
- Unique external form, high reliability, programmable, DIN Rail 35 standard installation.
- User can programme to set address, baud rate, etc for the modules.

Applications:

- Monitoring, measuring and remotely controlling the industrial field through local area network.
- Applied in the industrial intelligent system for Ethernet, RS232/485 can bus
- Gathering 0-10V/4-20mA signal and converting into Ethernet RJ45 output.
- Sensor signal A/D conversion, network control and long distance transmission.
- Ethernet multi-channel analog data acquisition, monitoring and long distance transmission.
- Meters and instruments data acquisition and recording
- Monitoring operating data from electric equipments, transportation and other equipments system.
- Achieve the network operation in Environment protection equipments, security, medical equipments and other equipments.

Chapter I Generalization

SUNYUAN SY AD 08-RJ45 series products achieve the analog data acquisition between sensor and host machine to detect analog signals or monitor remote equipments. Through setting the software, it can be applied in various types of sensor including: analog signal input, analog signal output and digital signal input/output. SY AD08-RJ45 series products can be applied in RS232/485 bus and RJ45 port in Ethernet industrial automatic control system, 4-20mA/0-10V analog measuring, monitoring and controlling, 0-1mA/0-30mV/small signal measuring, industrial site signal isolation and long distance transmission, etc.

The products have the following functions: power supply isolation, input and output signal isolation, linearization, A/D conversion and RS485 serial or Ethernet communication. Each serial port can connect at most 256 pieces of SYAD08-RJ45 series modules, the communication methods adopt serial port ASCII character communication protocol or Modbus RTU protocol and Ethernet Modbus TCP protocol, its command set is compatible with ADAM modules, the configuration parameter can be set by users, and the module can operate well in the same can bus with the controlling modules from other companies to make it easy to programme through computer.

SY AD08-RJ45 module is an intelligent monitoring and controlling system based on MCU, all the IP address, interface number, calibration value, ID address, baud rate, data format, states and other configuration information which set by user are stored in the nonvolatile storage device.

SY AC08-RJ45 series products are designed and developed by following industrial standards, which can withstand 3000VDC isolation power. The signal input/output are isolated, the modules have high anti-interference performance, high reliability. Operating temperature range: - 45°C ~ +85°C.

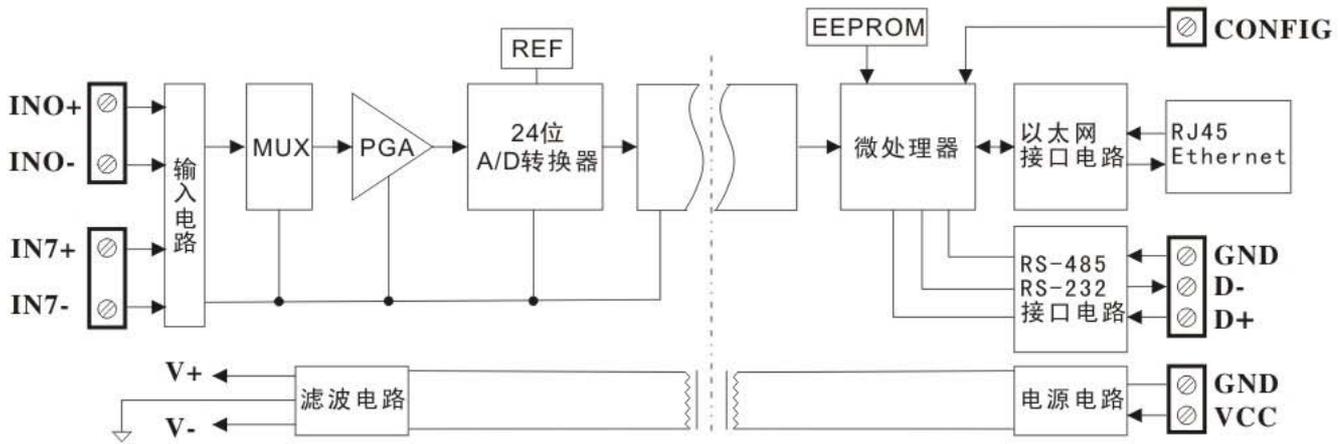


Fig.1 SY AD 08-RJ45 Product Functional Block Diagram

SY AD 08-RJ45 Function:

SY AD 08-RJ45 signal isolated data acquisition module can be used to measure 8-channel voltage or current signals, it supports ASCII protocol for serial ports, standard Modbus RTU protocol and Ethernet Modbus TCP protocol.

1. Analog Signal Input

24-bit data acquisition precision, the signal input range of the product has been calibrated before leaving the factory. User can also programme the module to do calibration. For the details about current or voltage input range, please check the model selection part for reference. In measuring multi-channel signal, the input of each channel must be the same.

2. Communication

(1) Serial Port Communication:

Communication Interface: one-channel standard RS485 communication port and one channel standard RS232 communication port.

Communication Protocol: support ASCII character protocol, Modbus RTU communication protocol. User can select the protocol through programming. It can achieve network communication with various types of PLC, RTU and computer monitoring system.

Data Format: 10 bit. One standard bit, eight data bit, one stop bit.

Communication Address: (00H-FFH) and baud rate (300bps, 600bps, 1200bps, 2400bps, 4800bps, 9600bps, 19200bps, 38400bps) can be set. The network communication distance is upto 1200 meters, the connection is realized by double ground shield cable. High anti-interference of the communication interface: ±15KV ESD protection, and the response time is less than 100ms.

(2) Ethernet Communication

Communication Interface: one channel standard 10M Ethernet interface.

Communication Protocol: support Modbus TCP protocol and operate well with various brand of PLC, computer monitoring system in network communication.

3. Anti-interference

Available to set checksum according to requirement. TVS inside the module can actively restrain all kinds of surge pulse and protect the module, and the digital filter inside can also restrain the power frequency interference from power grid.

Model Selection:

SY AD 08 - RJ45 - U_□ / A_□

Input Voltage/Current Signal

U1: 0-5V	A1: 0-1mA
U2: 0-10V	A2: 0-10mA
U3: 0-75mV	A3: 0-20mA
U4: 0-2.5V	A4: 4-20mA
U5: 0-±5V	A5: 0-±1mA
U6: 0-±10V	A6: 0-±10mA
U7: 0-±100mV	A7: 0-±20mA
U8: Customized	A8: Customized

Note: SY AD 08-RJ45 module supports RS232/RS485 and Ethernet interface, users can choose the corresponding communication interface based on the operating conditions. But RS232/485 Interfaces and Ethernet RJ45 Interface cannot operate simultaneously, otherwise, interferes with each other.

- E.g.1:** Model No: SY AD 08-RJ45-A4; refers to 8-channel 4-20mA signal input, Ethernet and RS232/RS485 output.
- E.g.2:** Model No: SY AD 08-RJ45-A7; refers to 8-channel 0-±20mA signal input, Ethernet and RS232/RS485 output.
- E.g.3:** Model No: SY AD 08-RJ45-U1; refers to 8-channel 0-5V signal input, Ethernet and RS232/RS485 output.

SY AD 08-RJ45 General Parameters:

(typical @ +25°C, Vs: 24VDC)

Input type: Current Input/ Voltage Input

Precision: 0.05%

Input offset: ±0.1 uA/°C

Temperature Drift: ±15 ppm/°C (±30 ppm/°C, Max)

Input Impedance: 50Ω (4-20mA/0-20mA/0-±20mA current input)

100Ω (0-10mA/0-±10mA current input)

1KΩ (0-1mA/0-±1mA current input)

More than 1MΩ(voltage input)

Bandwidth: -3 dB 10 Hz

Conversion Rate: 10 Sps

CMR: 120 dB (1kΩ Source Imbalance @ 50/60 Hz)

NMR: 60 dB (1kΩ Source Imbalance @ 50/60 Hz)

Input Terminal Protection: over-voltage protection, over-current protection

Communication:

Serial Port:

support standard Modbus RTU protocol and special ASCII character communication protocol, RS232/RS485 interface.

Baud Rate: (300 bps、600 bps、1200 bps、2400 bps、4800 bps、9600 bps、19200 bps、38400bps) ,can be set through software).

Address (0x00~0xFF) can be set through software

Ethernet:

Support standard Modbus TCP protocol, RJ4 network interface.

IP address and terminal number can be set through software.

Communication Response Time: 100 ms Max.

Operating Power Supply: +8 — 50 VDC wide power supply range, with internal anti-inverse connection and over-voltage protection circuit.

Power Consumption: less than 2W

Operating Temp.: - 45 — +80°C

Operating Humidity: 10 — 90% (no condensation)

Storage Temp.: - 55 — +85°C

Storage Humidity: 10 — 95% (no condensation)

Isolation Withstand Voltage : 3KVDC between input and output every minute, stray current 1mA, output and power are common-ground.

Endure Impact Volt: 3KVAC, 1.2/50us (peak value)

Dimension: 83 x 37 x 51 mm

Weight: about 72g

Outside View & Pin Definition:

PIN	Name	Description
1	IN0+	Input channel 0 positive terminal
2	IN0-	Input channel 0 negative terminal
3	IN1+	Input channel 1 positive terminal
4	IN1-	Input channel 1 negative terminal
5	IN2+	Input channel 2 positive terminal
6	IN2-	Input channel 2 negative terminal
7	IN3+	Input channel 3 positive terminal
8	IN3-	Input channel 3 negative terminal
9	IN4+	Input channel 4 positive terminal
10	IN4-	Input channel 4 negative terminal
11	IN5+	Input channel 5 positive terminal
12	IN5-	Input channel 5 negative terminal
13	IN6+	Input channel 6 positive terminal
14	IN6-	Input channel 6 negative terminal
15	VCC	Power supply +
16	GND	Power supply GND
17	D+	RS485/232 positive terminal
18	D-	RS485/232 negative terminal
19	GND	GND
20	CONFIG	Configuration pin
21	IN7+	Input channel 7 positive terminal
22	IN7-	Input channel 7 negative terminal
23	RJ-45	Ethernet interface



Fig. 2 SY AD 08-RJ45 Outside View

Connection Diagram:

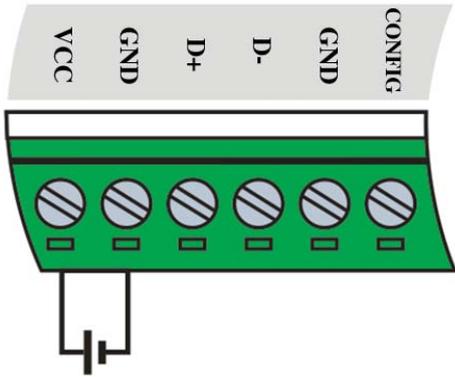
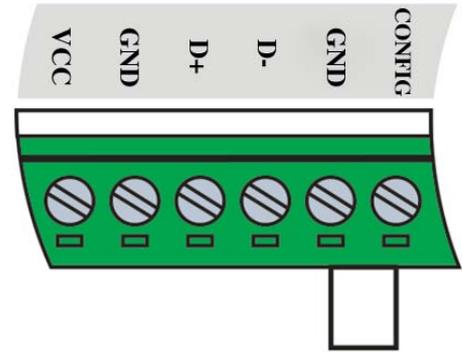


Fig.3 Power Supply Connection Diagram



将CONFIG管脚短路接到地线(GND管脚)后, 再接通电源, 此时模块进入配置状态

Fig.4 Configuration Mode Connection Diagram

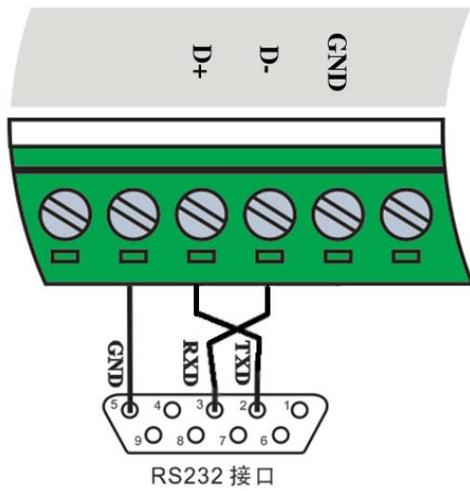


Fig. 5 RS232 Interface Connection Diagram

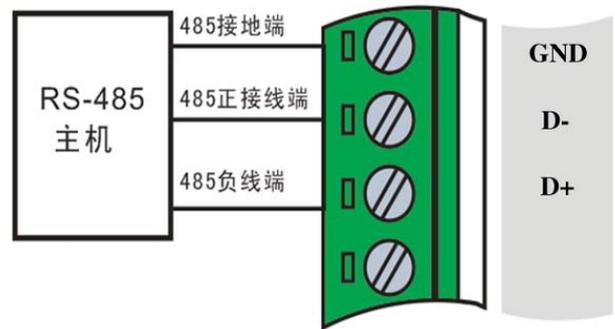


Fig. 6 RS485 Interface Connection Diagram

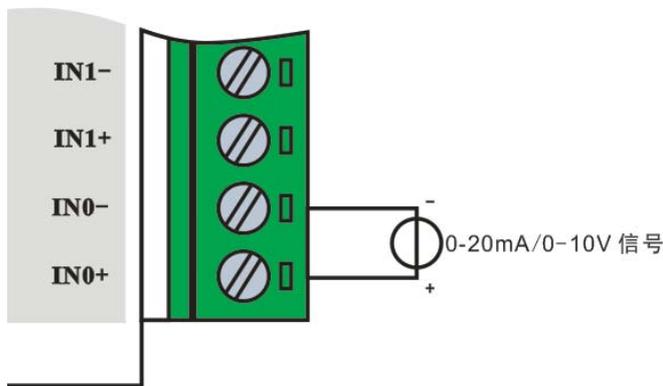


Fig. 7 Input Signal Connection Diagram

Initialization of SY AD08-RJ45 Module

(1) For all the SYAD 08T-RJ45 series modules, it must be allocated an unique IP address code in RS232/RS485 network, the address code is hexadecimal digits and the value range of address code should between 0x00 and 0xFF. All the SYAD08-RJ45 series modules have the same one set of initial setting from factory, as below:

Address code is 01H

Baud rate: 9600bps

Checksum: disable

ASCII character communication protocol

As the new module address code is the same, their addresses will be contradictory with other modules, so when users set up the system, the re-configuration for each analog output module address must be done. Users can modify SYAD08-RJ45 series module's address through configuration commands after connecting SYAD08-RJ45 series module power cable and RS232/RS485 communication cable. Baud rate, calibration, status, and communication protocols can also be adjusted according to user's requirements. Before the adjusting process, one must set the module to be the default state firstly; otherwise it can not be modified.

(2) SY AD08-RJ45 series modules must be allocated an unique address IP and terminal number in Ethernet network. All the SYAD08-RJ45 series modules have the same one set of initial network setting from factory, as below:

IP address is: 192.168.0.80

Terminal number: 80

Modbus TCP communication protocol.

As the new module IP address is the same, their IP addresses will be contradictory with other network IP addresses, so when users set up the system, the re-configuration for each module IP address and terminal number must be done. Users can modify SYAD08-RJ45 series module's address through configuration commands after connecting SYAD08-RJ45 series module power cable and communication cable. The communication protocols can also be adjusted according to user's requirements. Before the adjusting process, one must set the module to be the default state firstly; otherwise it can not be modified.

SY AD08-RJ45 series module has a Pin marked CONFIG. Connect CONFIG Pin to the ground (GND) and then connect power supply. In this way, the module is in default status.

Default States:

Let the modules be in default states, under default states, it supports ASCII character and Modbus TCP communication protocol, the module setting is as following:

(1)Serial Ports:

Address code: 00H

Baud rate: 9600 bps

Checksum: disable

ASCII Character communication protocol

(2)Ethernet:

IP address is: 192.168.0.80

Terminal number: 80

Modbus TCP communication protocol.

At this time, users can modify the module parameters by configuring the command. If the exact configuration of a module is uncertain, the module could be in default state by installing the jumper, then reconfigure the module. If user requires the module to be set to MODBUS RTU communication protocol, see the chapter of MODBUS communication protocol section for reference.

Chapter II ASCII Character Communication Protocol

Brief introduction

When the controller communicates on the MODBUS under ASCII character protocol mode, every 8-bit of a message is transmitted as two ASCII bytes. E.g.: Transmit the number 34: it transmits 0x33 (ASCII code of 3) and 0x34 (ASCII code of 4) separately. The advantage is that the interval of sending the bytes is only 1s without error.

Character Protocol Command Set

The command is formed by a series of characters, e.g.: first code, address ID, variable, selectable calibration and bytes and one end fent mark (**cr**) shows the end of command. SYAD08-RJ45 series module is not support broadcast address, so the host only control one module one time.

Format of each ASCII code's byte: (Leading Code)(Addr.)(Command)[data][checksum](cr)

Leading code)	the first letter of a command, every command requires a leading code as %, \$, #, @, etc.	1- C
(Addr.)	address code of module, the default value range in the following is 00~FF (hexadecimal system).	2- C
(Command)	command code/variable value	1- C
[data]	command parameter	variable length
[checksum]	calibration value, selectable parameter, it requires only in the operation of checksum	2- C
(cr)	a control code character used as the end code. (cr) as end code of the enter the value is 0x0D	1- C

Calibrate and check the communication between host and module is correct or not. When checksum is enabled, both command and response must be added additional checksum parameter which occupies 2 bytes. The checksum and characters should be before enter key mark after command or response character.

Calculation method: Get the sum of ASCII code value of all the prior issued bytes. This sum and hexadecimal digits 0xFF is two characters (the hexadecimal number).

E.g.: Prohibit checksum

```
User command    $022(cr)
Module response !02000600 (cr)
Enable checksum
User command    $022B8 (cr)
Module response !02000640AD (cr)
'$' = 0x24    '0' = 0x30    '2' = 0x32
B8=(0x24+0x30+0x32+0x32) AND 0xFF
'!' = 0x21    '0' = 0x30    '2' = 0x32    '4' = 0x34    '6' = 0x36
AD=(0x21+0x30+0x32+0x30+0x30+0x30+0x36+0x34+0x30) AND 0xFF
```

Commonly Used Analog Input Commands

1. Read analog input module data
2. Read the input data of analog channel N
3. Module allocation
4. Read allocation states
5. Offset calibration
6. Full scale calibration
7. Read module name
8. Enable or prohibit channel command
9. Read channel states
10. Read serial ports communication protocol states
11. Set serial ports communication protocol
12. Read the network port number
13. Set the network port number
14. Read the network IP address
15. Set the network IP address

Response of Commands

Response message depends on a variety of commands. Response also consists of several characters, including leading code, variables and end tags. There are two leading codes of response signal: '!' or '>' indicates a valid command and '?' means invalid. By checking the response information, user can monitor whether the command is valid or not.

Note: 1. User needs to ensure the address he uses is correct in a command. If one use the wrong address, the object module will not response.

2. The command must be entered in capital letters.

1. Read Analog Input Module Data

Description: Based on the current data format, the command is to read analog input value of all channels from the module.

Command Syntax: **#AA(cr)**

Parameter Description: # d velimiter character
AA module address, value range 00H~FFH(hexadecimal)
(cr) end character, carriage return(0DH)

Response: **>(data)(cr)** command is valid.
?AA(cr) invalid command or illegal operation

Parameter Description:
> delimiter character
(data) represent the data of all channels which been read by the command. The data format is in Engineering Units, percent of FSR, complement of hexadecimal or ohms. For details, check article 3 of Command Set.
(cr) end character, carriage return (0DH).

More instructions: If there is a syntax error or communication error or address does not exist, the object module does not respond. If certain channel is closed, the data read out as space character.

E.g.: User command #23(cr)
Module response >+4. 7653+4. 7653+4. 7653+4. 7653+4. 7653+4. 7653+4. 7653+4. 7653 (cr)
Description: On the module with address 23H, input is +4. 7653V (data format is engineering unit):
Channel 0-7: +04. 7653V

2. Read the Data of Analog Input Module in Channel N

Description: Read the analog input data of channel N from the module at the current allocated data format.

Command Syntax: #AAN(cr)

Parameter Description: # delimiter character
AA module address, value range 00H~FFH
N channel 0 or 1
(cr) end character, carriage return(0DH)

Response: >(data)(cr) command is valid.
?AA(cr) invalid command or illegal operation or channel is closed.

Parameter Description: > delimiter character
(data) data read in channel N. The data format is in Engineering Units, PERCENT OF FSR, complement of hexadecimal, or ohms. Refer to Article 3 in Command Set.
(cr) end character, carriage return(0DH)

More instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command #230(cr)
Module response >+04.632 (cr)

Description: The input current of channel 0 with address 23H module is +04.632mA (data format is engineering unit).

3. Module Allocation

Description: Set address, input range, baud rate, data format, checksum status for an analog input module. Configuration information is stored in nonvolatile memory EEPROM.

Syntax: %AANN TTCCFF(cr)

Parameter Description:
% delimiter character.
AA module address, value range 00H~FFH.
NN hexadecimal address, NN is from 00 to FF.
TT hexadecimal represents type code.
CC hexadecimal represents baud rate code. **Refer to table 2**

Baud Rate Code	Baud Rate
01	300bps
02	600bps
03	1200bps
04	2400bps
05	4800bps
06	9600bps
07	19200bps
08	38400bps

Table 2 Baud Rate Code

FF The 8-bit of hexadecimal is used to represent data format and checksum. (Note: From bits2 to bits 5 is unnecessary to be set to 0.)

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit2	Bit 1	Bit 0
------	-------	-------	-------	-------	------	-------	-------

Table 3 Data Format, Checksum Code

Bit7: Reserved bits should be set to 0
Bit6: checksum status: 0 represents prohibit; 1 represents permission
Bit5-bit2: it's unnecessary and used to be set to 0
Bit1-bit0: data format bit
00: Engineering Units

01: % of FSR

10: complement of hexadecimal (Twos complement)

(cr) end character, carriage return(0DH).

Response: **!AA(cr)** command is valid.

?AA(cr) invalid command or illegal operation, no jumper allocation before baud rate checksum status.

Parameter Description: **!** delimiter code indicates valid command.

? delimiter code indicates invalid command.

AA indicates input Module address

(cr) end character, carriage return(0DH)

More instructions:

Note 3. If need to reallocate module, user should install jumper to let the module be in default status. Now Module address is 00H,that is AA=00H,NN = new address. Otherwise it will respond error signal.

Note 4. Once the command is executed, the new address will be set as response information and be send back to the host machine by the module after reallocation.

If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **%0011000600(cr)**

Module response **!11(cr)**

Description: **%** Delimiter character

00 Indicate the analog module original address which user wants to allocate is 00H.

11 Indicate the hexadecimal address of the new address is 11H.

00 Type code

06 Means baud rate 9600 bps

00 Indicate data format is engineering unit and the checksum is forbidden.

4. Read the Allocation States

Description: Read the configuration information of the specified module

Command Syntax: **\$AA2(cr)**

Parameter Description:

\$ delimiter character

AA module address,value range 00H~FFH

2 read the configuration status

(cr) end character, carriage return (0DH)

Response: **!AATTCCFF(cr)** command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description :

! delimiter character

AA input module address

TT Type code

CC baud rate code (see table 2)

FF table 3 for reference

(cr) end character,upper machine enter key(0DH)

More instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$302(cr)**

Module response **!300F0600(cr)**

Description: **!** delimiter character

30 analog input module address is 30H

00 input type code

06 baud rate 9600 bps.

00 data format is engineering unit and the checksum is forbidden.

5. Offset Calibration

Description: Adjust the zero offset of channel N

Command Syntax : **\$AA1N(cr)**

Parameter Description :

\$ Delimiter character

AA Module address,value range 00H~FFH

1 Offset calibration

N Channel code 0~7

(cr) End character,upper computer enter key(0DH).

Response: !AA(cr) command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description:

! delimiter code means valid command

? delimiter code means invalid command

AA input Module address

(cr) end character,upper computer enter key(0DH).

More instructions: The product has been calibrated before delivery, and the user can directly use it without calibration.

When calibrating an analog input, offset calibration should be done firstly, and then calibrate the gain.

During the calibration, analog input module needs to be connected with the appropriate input signal on the channel which is to be calibrated. Different input range requires different input voltage or current. See the calibration module section for reference.

If there is a command syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command \$2310(cr)

Module response !23(cr)

Description : Offset calibrating channel 0 with address 23H

6. Gain Calibration Command

Description: calibrate the gain of channel N

Command Syntax: \$AA0N(cr)

Parameter Description:

\$ Delimiter character

AA Module address,value range 00H~FFH

0 Gain calibration command

N Channel code 0~7

(cr) End character,host computer enter key(0DH)

Response: !AA(cr) command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description: ! delimiter code means valid command

? delimiter code means invalid command

AA input Module address

(cr) end character, carriage return(0DH)

More instructions: The product has been calibrated before delivery, and the user can directly use it without calibration. When calibrating an analog input, offset calibration should be done firstly, and then calibrate the gain.During the calibration, analog input module needs to be connected with the appropriate input signal on the channel which as to be calibrated. Different input range requires different input voltage or current. See the calibration module section. If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command \$2303(cr)

Module response !23(cr)

Description: gain calibration of channel 3 with address 23H module.

7. Read Module Name

Description: Read module name

Command Syntax: \$AAM(cr)

Parameter Description:

\$ Delimiter character

AA Module address, value range 00~FF

M Read module name

(cr) End character,host machine enter key(0DH)

Response: !AA(Module Name)(cr) command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description:

! delimiter character:command is valid.

? delimiter character:command is invalid.

AA input Module address

(Module Name) module name could be SYAD08-RJ45 etc. Means the module type user used.

(cr) end character, host machine enter key(0DH)

More instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$08M(cr)**
Module response **!08SY AD08-RJ45 (cr)**

Description: The module in address 08H is SY AD08-RJ45.

8. Enable or Prohibit Channel Command

Description: Enable or prohibit the data acquisition channel of the module

Command Syntax : **\$AA5VV(cr)**

Parameter Description :

\$ Delimiter character
AA Module address, value range:00H~FFH (hexadecimal system)
5 The command means enable or prohibit the data acquisition channel of the module
VV Two bids (hexadecimal number), 7 to 0 bit of the binary expression represents 7~0 channel.
Bit is 0: prohibit the channel
Bit is 1: enabled the channel

(cr) end character, host machine enter key(0DH)

Response: **!AA(cr)** command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description: **!** delimiter character: command is valid.

? delimiter character: command is invalid.

AA input Module address

(cr) end character, host machine enter key (0DH).

More instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$08537(cr)**
Module response **!08 (cr)**

Description: set object channel value to 0x37.

3 is 0011,represents enable channel 5 and channel 4, prohibit channel 7 and channel 6.

7 is 0011,represents enable channel 2 and channel 1, channel 0, prohibit channel 3.

9. Read the Channel States

Description: Read the channel status command

Command Syntax : **\$AA6(cr)**

Parameter Description:

\$ Delimiter character
AA Module address,value range 00H~FFH.
6 Read the channel status
(cr) End character, host machine enter key(0DH)

Response: **!AAVV(cr)** command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description :

! delimiter character :command is valid.
? delimiter character: command is invalid.
AA input Module address
VV two 16 bits (hexadecimal), 7 to 0 bit of the binary expression represents 7~0 channel.
Bit is 0: prohibit the channel
Bit is 1: enabled the channel

(cr) end character,host machine enter key(0DH).

More instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$186 (cr)**
Module response **!18FF (cr)**

Description : the current channel status is 0xFF.

0xFF is 1111and 1111, means all the channels of the module with address 18H have been enabled.

10. Read Serial Ports Communication Protocol States

Function: read the communication protocol states of the module.

Command Syntax: **\$AAP(cr)**

Parameter Description:

- \$** Delimiter character
- AA** Module address,value range 00H~FFH.
- P** key words of communication protocol of serial ports
- (cr)** End character, host machine enter key(0DH)

Response: **!AAPV(cr)** command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description :

- !** delimiter character: command is valid.
- ?** delimiter character: command is invalid.
- AA** target Module address
- P** key words of communication protocol of serial ports
- V** Code number of protocol, (0 or 1)
 - 0:** Character protocol which defines by command set
 - 1:** MODBUS RTU Protocol
- (cr)** end character, host machine enter key(0DH)

More Instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$00P(cr)**

Module response **!00P1 (cr)**

Description: set the communication protocol to MODBUS Protocol

11. Set Serial Ports Communication Protocol

Function: set the communication protocol to ASCII character communication protocol or MODBUS RTU protocol.

Command Syntax: **\$AAPV(cr)**

Parameter Description:

- \$** Delimiter character
- AA** Module address,value range 00H~FFH.
- P** Set communication protocol
- V** Code number of protocol, (0 or 1)
 - 0:** Character protocol which defines by command set
 - 1:** MODBUS RTU Protocol
- (cr)** End character, host machine enter key(0DH)

Response: **!AA(cr)** command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description :

- !** delimiter character :command is valid.
- ?** delimiter character: command is invalid.
- AA** input Module address
- (cr)** end character, host machine enter key(0DH)

More Instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

The command for setting communication protocol is valid must be in the allocation status.

E.g.: User command **\$00P1(cr)**

Module response **!00 (cr)**

Description: set the communication protocol to MODBUS Protocol

E.g.: User command **\$00P0(cr)**

Module response **!00 (cr)**

Description: set the communication protocol to character protocol which defines by command set.

12. Read the Network Port Number

Function: read port number of the Ethernet interface.

Command Syntax: **\$AAW(cr)**

Parameter Description:

- \$** Delimiter character
- AA** Target module address,value range 00H~FFH (hexadecimal).
- W** Key words of network port numbers.
- (cr)** End character, host machine enter key(0DH)

Response: **!AAxxxx(cr)** command is valid.
?AA(cr) invalid command or illegal operation

Parameter Description :

! delimiter character :command is valid.
? delimiter character: command is invalid.
AA target module address.
W Key words of network port numbers.
xxxx represents 16-bit digit with hexadecimal format.
(cr) end character, host machine enter key(0DH)

More Instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$00W(cr)**
Module response **!00W0050(cr)**

Description: the port number of current network port is 0x50, that is 80.

13. Set the Network Port Number

Function: set port number of the Ethernet interface.

Command Syntax: **\$AAWxxxx(cr)**

Parameter Description:

\$ Delimiter character
AA Target module address,value range 00H~FFH (hexadecimal).
W Key words of network port numbers.
xxxx Represents 16-bit digit with hexadecimal format.
(cr) End character, host machine enter key(0DH)

Response: **!AA(cr)** command is valid.
?AA(cr) invalid command or illegal operation

Parameter Description:

! delimiter character :command is valid.
? delimiter character: command is invalid.
AA target module address
(cr) end character, host machine enter key(0DH)

More Instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$00W0050(cr)**
Module response **!00(cr)**

Description: set the port number of current network port to 0x50, that is 80.

14. Read the Network IP Address

Function: read network IP address of the module.

Command Syntax: **\$AAD(cr)**

Parameter Description:

\$ Delimiter character
AA Module address,value range 00H~FFH.
D Key words of network IP
(cr) End character, host machine enter key(0DH)

Response: **!AAD:xx-yy-zz-nn(cr)** command is valid.
?AA(cr) invalid command or illegal operation

Parameter Description:

! delimiter character: command is valid.
? delimiter character: command is invalid.
: delimiter character
- delimiter character
AA target Module address
D key words of network IP
: delimiter character
xx represent the 31st-24th order digits of the IP address (two hexadecimal digits).
yy represent the 23rd-16th order digits of the IP address (two hexadecimal digits).
zz represent the 15th-8th order digits of the IP address (two hexadecimal digits).
nn represent the 7thrd-0 order digits of the IP address (two hexadecimal digits).
(cr) end character, host machine enter key(0DH).

More Instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$00D(cr)**
 Module response **!00D: C0-A8-00-50 (cr)**

Description: the IP address of current module is 0xC0A80050, that is 192.168.0.80.

15. Set the Network IP Address

Function: set network IP address of the module.

Command Syntax: **\$AAD:xx-yy-zz-nn(cr)**

Parameter Description:

- \$** Delimiter character
- :** delimiter character
- delimiter character
- AA** Module address, value range 00H~FFH.
- D** Key words of network IP.
- xx** represent the 31st-24th order digits of the IP address (two hexadecimal digits).
- yy** represent the 23rd-16th order digits of the IP address (two hexadecimal digits).
- zz** represent the 15th-8th order digits of the IP address (two hexadecimal digits).
- nn** represent the 7thrd-0 order digits of the IP address (two hexadecimal digits).
- (cr)** End character, host machine enter key(0DH)

Response: **!AA(cr)** command is valid.

?AA(cr) invalid command or illegal operation

Parameter Description:

- !** delimiter character: command is valid.
- ?** delimiter character: command is invalid.
- (cr)** end character, host machine enter key(0DH)

More Instructions: If there is a syntax error or communication error or address does not exist, the module does not respond.

E.g.: User command **\$18W(cr)**
 Module response **!180050(cr)**

Description: set the IP address of current module to 0xC0A80050, that is 192.168.0.80.

Input Range and Data Format

There are 4 types of output data format are used in analog input module:

- 00: Engineering Units
- 01: % of FSR
- 10: Complement hexadecimal
- 11: Ohms (only in thermocouple)

Input range	Data format	Span +	Zero	Span -	Res.
A1: 0-1mA A5: 0-±1mA	Engineering Unit	+1.0000	±0.0000	-1.0000	0.1uA
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB
A2: 0-10mA A6: 0-±10mA	Engineering Unit	+10.000	±00.000	-10.000	1uA
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB
A3: 0-20mA A4: 4-20mA A7: 0-±20mA	Engineering Unit	+20.000	±00.000	-20.000	1uA
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB
U1: 0-5V U5: 0-±5V	Engineering Unit	+5.0000	±0.0000	-5.0000	0.1mV
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB

U2: 0-10V U6: 0±10V	Engineering Unit	+10.000	±00.000	-10.000	1mV
	% of FSR	+100.00	±000.00	-100.00	0.01%
	Complement of hexadecimal	7FFFFFFF	000000	800000	1LSB
U3: 0-75mV	Engineering Unit	+75.000	±00.000	-75.000	1uV
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB
U4: 0-2.5V	Engineering Unit	+2.5000	±0.0000	-2.5000	0.1mV
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB
U7: 0-100mV	Engineering Unit	+100.00	±000.00	-100.00	0.01mV
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB
A8: customized U8: customized	Engineering Unit	+100.00	±000.00	-100.00	0.01%
	% of FSR	+100.00	±000.00	-100.00	0.01%
	complement of hexadecimal	7FFFFFFF	000000	800000	1LSB

Table 4 Input Range and Data Format

Application Cases

1. input range is 4~20mA, input is 4 mA:

user command		#01(cr)
Engineering Unit	Module response	>+04.000(cr)
% of FSR	Module response	>+020.00(cr)
complement of hexadecimal	Module response	>199999(cr)

2. Input range is ±10V, output is 2.5V

user command		#01(cr)
Engineering Unit	Module response	>+02.500(cr)
% of FSR	Module response	>+025.00(cr)
complement of hexadecimal	Module response	>1FFFFFF(cr)

3. Conversion relation between hexadecimal data and analog

When the input of module is current/voltage. The conversion formula is as following.

$$X/7FFFFFFF=Xin/Xf$$

Instructions: X: represents the corresponding hexadecimal data of the analog.

Xin: represents the analog signal the channel input.

Xf: represents the analog of the scale span of product, like, 20mA.

4. The input type is voltage value, Xin is the voltage input Vin, Xf is full span voltage input.

5. If the input is negative voltage/current, read the value, get its reverse plus one, then use the formula above to calculate according analog value.

Chapter III MODBUS RTU Protocol

Introduction

MODBUS protocol defines the information structure which could be identified and used by controller. When communicating on the MODBUS network, the protocol allows each controller to know its own device address, and identify the data addressing to it, then decide the effective type and take out data contained in the information. The controller can also organize the response information and use the MODBUS protocol to send out this information.

The controller takes use of master-subordinate technique. That means, only one device (the master) can initiate transmission (the query), and other devices (subordinate ones) respond according to data queried by the master device. A typical master device: host and programmable instrument. Typical subordinate device: programmable controllers.

Master device can communicate with one subordinate device, but also communicate with all subordinate devices by broadcast. For the former situation, the device will respond one message; for the latter, the device will not respond. MODBUS protocol establishes a master query format: device (or broadcast) address, function code, all the data to be sent, and an error detection field.

Response message of subordinate device is also constituted by the MODBUS protocol, including the domain to be act, any data to be returned, and an error detection field. If a error occurred during message reception, or the subordinate device can not perform its command, the subordinate device will create an error message and sent it out as a response
MODBUS transmission: ASCII and RTU

Communication in ASCII mode, each 8-byte in a message is sent as two ASCII character. It uses the LRC error detection, the advantage of which is that the interval of sending the character is only 1s without error; while in RTU mode, every 8 bit byte contains two 4 bit of hexadecimal characters. It uses the CRC error detection, and the advantage is at the same baud rate, it can transmit more data than ASCII.

About ASCII mode and RTU data format, more information can be found in the relevant MODBUS communication protocol:

	ASCII Mode	RTU Mode
Code system	<ul style="list-style-type: none"> hexadecimal,ASCII character 0~9, A~F every ASCII byte consists of one hexadecimal character 	<ul style="list-style-type: none"> 8 bit binary system,hexadecimal0~9,A~F Every 8 bit consists of one or two hexadecimal character
Data bit	<ul style="list-style-type: none"> 1 start bit 7 data bits,the smallest significance bit significance bit sent firstly 1 parity check bit,none if there is no calibration 1 stop bit (with calibration),2 Bits (without calibration) 	<ul style="list-style-type: none"> 1 start bit 8 data bits,the smallest significance bit significance bit sent firstly 1 parity check bit,none if there is no calibration 1 stop bit (with calibration),2 Bits (without calibration)
Error detection	LRC(Longitudinal rigmarole check)	CRC(circulate rigmarole check)

Note: The module only supports MODBUS RTU transmission, not support MODBUS ASCII transmission.

1. Set MODBUS RTU Mode

The device use ASCII command set as default protocol. To use MODBUS RTU, follow these steps to set:

1. Short circuit CONFIG (the 3rd pin) and GND (the 4th pin)
2. Correctly link Power and other interface cables
3. Switch on power supply, module is in default state automatically, communication address is **00**, baud rate is **9600**.
4. The module is to complete initialization, about 1 minutes
5. Send command **\$00P1(cr)**(00 represents corresponding module address,check set communication protocol for reference to set other protocols), check response, if it is **!00(cr)**, set correctly.
6. Switch off power supply, cut off the connecting between CONFIG and GND
7. Module has been set to MODBUS RTU protocol

2.Data Format under MODBUS RTU

Host Machine Inquiry

Host query message specifies the start address of the register to be read and the quantity of registers. The start address of the register addressing is 0000. Checking format is as following:

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	Error Check(CRC)
01	03	00	00	00	08	44 0C

E.g.:Host send query information Tx: 01 03 00 00 00 08 44 0C

- 01 device address
- 03 function code
- 00 00 the start address of register addressing
- 00 08 quantity of register
- 44 0C CRC check code

3. Subordinate Device Response

Register data in the response information of subordinate device corresponds to the two bytes of each register; the first data byte is high, the second is the low data. Response format is as following:

Slave Address	Function	Byte Count	Data Hi(Register 40001)	Data Lo(Register 40001)	Data Hi(Register 40002)	Data Lo(Register 40002)	Data Lo(Register 40008)	Error Check(CRC)
01	03	10	19	99	00	00	02	9E 68

E.g.: Subordinate device response information Rx: 01 03 10 19 99 00 00 00 00 00 00 00 00 00 04 00 00 00 00 87 69

- 01 device address
- 03 function code
- 10 the quantity of bytes
- 19 99 the first register data
- 00 00 the second register data
-
-
- 87 69 CRC check code

The below picture is the register information checked by MODSCAN software and its instruction

<p>The screenshot shows the ModScan software interface. Fields include: Address: 0001, Length: 2, Device Id: 1, MODBUS Point Type: 03: HOLDING REGISTER, Number of Polls: 39, Valid Slave Responses: 39, and a Reset Ctrs button. A data window shows: 40001: <07FFH>, 40002: <0333H></p>	<p>Instructions:</p> <ul style="list-style-type: none"> Address is the start address of the register Device ID is module address Length is the number of registers been read MODBUS Point Type :MODBUS function type selection 40001: corresponds to register data, analog input value of channel 0 40002: corresponds to register data, analog input value of channel 1
---	--

4. Instructions of Register

Address 4X	Data	R/W	Description
40001	IN0	Read only	Measured Value of Channel 0
40002	IN1	Read only	Measured Value of Channel 1
40211	Module Name	Read only	Higher bits: 0x40 Lower bits: 0x21
40221	Channel Status	Read/Write	Higher bits: 0x00 Lower bits: Channel Status (0x03)

Form 5 MODBUS RTU Register Instructions

Chapter IV Modbus TCP Communication Protocol

Introduction

Modbus TCP Protocol is another version of Modbus protocol, which is developed in 1999 to allow internet users to access to Ethernet equipments. Modbus TCP Protocol is open, easy to learn and understand, so it become one the world leading industrial Ethernet Protocols at present.

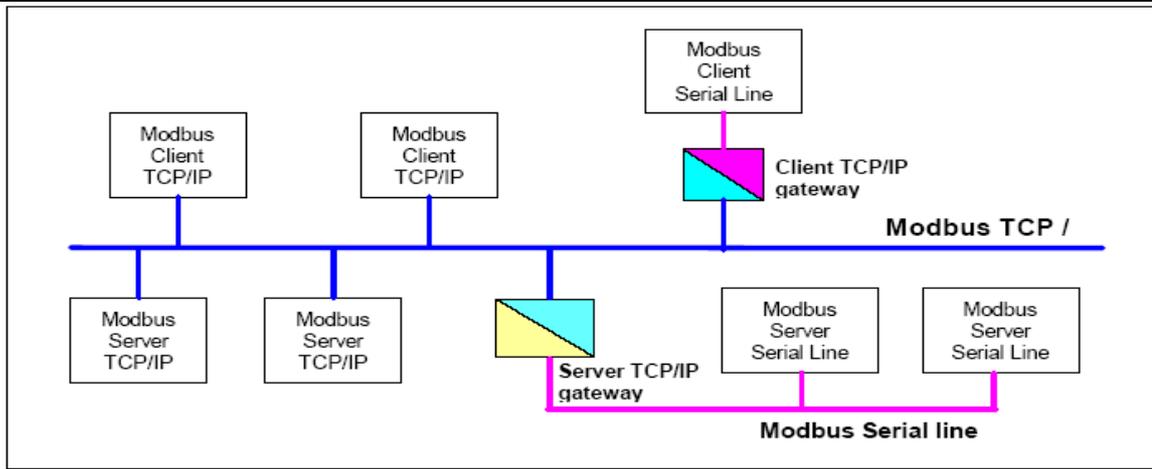


Fig.8 Industrial Ethernet Equipments Network

The protocol defines a message structure which can be recognized by the controller regardless of the network used in communication. It describes the process for controllers' request to access to other equipments, such as, response to the request of other equipments, how to detect errors and record them. It formulates the message domain format and the universal format of contents.

In Modbus network communication, the protocol decides that every controller must know its equipments address, recognize the message sent based on address and decide taking what kinds of actions. If response required, the controller will generate feedback information and send through Modbus Protocol. In other network, the messages with Modbus Protocol are converted into frame or package structures used in that network. The conversion also diversifies the methods to solve the problems on bit address, routing path and error detection.

1. Modbus TCP Mode

MBAP Header	Function Code	Data Quantity	Data 1	Data n
-------------	---------------	---------------	--------	-------	--------

Protocol Head Functional Domain Data Domain

Frame Alignment: In Modbus TCP mode, the address of a module is determined by the IP address, so there is no longer has address domain contents. And TCP network is the reliable data transmission network, so there is no longer has calibration data. But because the inconformity of the arriving order between the data from IP network and the data expected, a data serial number is added. And the Modbus TCP carries the Modbus protocol, so an address domain is added in the leading data.

MBAP Header and Protocol Head include the domains below:

Domain	Length	Description	Client Server	Server
Transaction Elements Identifier	2 bytes	MODBUS request/ response code	Client server startup	Server re-copy from the request it received
Protocol Identifier	2 bytes	0=MODBUS Protocol	Client server startup	Server re-copy from the request it received
Length	2 bytes	The quantity of the bytes below	Client server startup(request)	Server (Response)startup
Unit Identifier	1 byte	Serial link identifier	Client server startup	Server re-copy from the request it received

Header format has 7 bytes.

Transaction Process Identifier: used for affairs process and match operation. In response, Modbus server copies transaction process identifier of request.

Protocol Identifier: used in multiple channels in the system. Identify the Modbus protocol.

Length: length domain is the byte count of next domain, which includes unit identifiers and the data domain.

Unit Identifier: the route of the system uses the domain. Achieve the connection between MODBUS/MODBUS and sub-webs of serial links through the gateway between Ethernet TCP-IP network and MODBUS serial links. MODBUS client machine sets the domain in request, the server must use the same value to return to this domain in response.

2. SY Support Commands

At present, all the Ethernet Data Acquisition Modules for SUNYUAN support MODBUS TCP Protocol. The function codes (hexadecimal) supported mainly include:

- 03 Read holding register
- 04 Read input register
- 06 Set single holding register
- 10 Set multiple holding register
- 41 SY customized command

3. Command and Address Mapping Table

(1) Read Holding Register

Function Code: 03

Instruction: Read holding register value

Data Explanation: the contents read: 16-bit integers or signs.

Address	Description	Explanation
40000-40001	Zero channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40002-40003	The first channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40004-40005	The second channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40006-40007	The third channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40008-40009	The fourth channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40010-40011	The fifth channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40012-40013	The sixth channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40014-40015	The seventh channel zero calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40016-40017	<Hold>	
40018-40019	<Hold>	
40020-40021	<Hold>	
40022-40023	<Hold>	
40024-40025	<Hold>	
40026-40027	<Hold>	
40028-40029	<Hold>	
40030-40031	<Hold>	
40032-40033	Zero channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40034-40035	The first channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40036-40037	The second channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40038-40039	The third channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40040-40041	The fourth channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40042-40043	The fifth channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40044-40045	The sixth channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40046-40047	The seventh channel slope calibration coefficient	Hexadecimal Format (valid for low 24-bit)
40048-40049	<Hold>	

(2) Read Input Register

Function Code: 04

Instruction: read input data

Data Explanation: the contents read: 16-bit integers or unsigned integers

Address	Description	Explanation
30000	The first channel analog input	16-bit analog input value (2bytes)
30001	The second channel analog input	16-bit analog input value (2bytes)
30002	The third channel analog input	16-bit analog input value (2bytes)
30003	The fourth channel analog input	16-bit analog input value (2bytes)
30004	The fifth channel analog input	16-bit analog input value (2bytes)
30005	The sixth channel analog input	16-bit analog input value (2bytes)
30006	The seventh channel analog input	16-bit analog input value (2bytes)
30007	The eighth channel analog input	16-bit analog input value (2bytes)
30008	<Hold>	
30009	<Hold>	
30010	<Hold>	
30011	<Hold>	
30012	<Hold>	
30013	<Hold>	
30014	<Hold>	
30015	<Hold>	

MODBUS Request

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Function Code	1byte	0x04
Leading Code	2bytes	0x0000 to 0x000F
Read Quantity	2bytes	1 to 16 (0x0010)

MODBUS Response

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Function Code	1byte	0x04
Byte Counts	1byte	2n
Input States	2bytes	

Error Response

Domain Name	Byte Count	Value
MBAP Header	7byte	
Function Code	1byte	0x04+0x80
Error Code	1byte	0x1 or 0x2

E.g.:

i .Request: read analog value input

00 00 00 00 00 06 00 04 00 01 00 02

Response:

00 00 00 00 00 07 00 04 04 00 00 FF FF

(3) Set Single Holding Register

Function Code: 06

MODBUS Request

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Function Code	1byte	0x06
Set Address	2bytes	0x0000 to 0x007F
Set Contents	2bytes	0x0000 to 0xFFFF

MODBUS Response

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Function Code	1byte	0x06
Set Address	2bytes	0x0040 to 0x007F
Set Contents	2bytes	0x0000 to 0xFFFF

Error Response

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Function Code	1byte	0x06+0x80
Error Code	1byte	0x1 or 0x2

E.g.:

i .Request: set communication protocol type
00 00 00 00 00 06 00 06 00 44 00 02
Response:
00 00 00 00 00 06 00 06 00 44 00 02

(4) Set multiple holding registers

Function Code: 10

MODBUS Request

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Function Code	1byte	0x10
Set leading address	2byte	0x0040 to 0x007F
Set length	2byte	0x0000 to 0x0100
Byte Counts	1byte	2n
Set Contents	2nbytes	

MODBUS Response

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Functional Code	1byte	0x10
Set leading address	2bytes	0x0040 to 0x00FF
Set length	2bytes	0x0000 to 0x0100

Error Response

Domain Name	Byte Count	Value
MBAP Header	7bytes	
Function Code	1byte	0x10+0x80
Error Code	1byte	0x1 or 0x2

E.g.:

i .Request: Set module ASCII ID/Baud rate
00 00 00 00 00 0B 00 10 00 40 00 02 04 30 31 00 36
Response:
00 00 00 00 00 06 00 10 00 40 00 02

(5) Set Zero-Gain Register

Function Code: 41

Instruction: SY Extended function code.

Format: 00 00 00 00 00 04 00 41 xx yy

xx: Sub-function code

01: zero calibration function

00: gain calibration function

yy: Channel Code 0-15 (hexadecimal)

E.g.:

Channe5 calibration (Zero calibration first, then gain calibration)

Zero:

Request: 00 00 00 00 00 04 00 41 01 05

Response: 00 00 00 00 00 04 00 41 01 05

Gain:

Request: 00 00 00 00 00 04 00 41 00 05

Response: 00 00 00 00 00 04 00 41 00 05

Chapter V Mode Calibration

The calibration of modules must be done based on ASCII character communication protocol. In the MODBUS protocol, the module does not support calibration.

The product has been calibrated before delivery, and the user can directly use it without calibration. User could recalibrate the module via the calibration function of the product. During the calibration, analog input module needs to be connected with the appropriate input signal on the channel which is to be calibrated. Different input range requires different input signal. In order to improve calibration accuracy, it is recommended to use the following equipments:

1. A stable output, low noise DC voltage/current signal source.
2. One five and a half bits or a higher-precision voltage/current measuring instruments to monitor the accuracy of the input signal.

The Calibration Process

1. Select the input channel to be calibrated and connect to the corresponding input signal according to the input range of the module. For SYAD 08 RJ-45 series Modules, Zero is calibrated when the input is 0; full scale is calibrated when input 120% of FSR. For example, when the input is 4-20mA, zero calibration: 0mA input; full scale calibration: 24mA. When the input is 0-5V, zero calibration: 0V input; full scale calibration: 6V.
2. Input zero signal to the analog input module which needs to calibrate, usually it is 0mA or 0V.
3. When the signal is stable, send ASCII character or MODBUS TCP offset calibration commands to the analog input module.

ASCII character: \$AA0N

Modbus TCP: 00 00 00 00 00 04 00 41 01 0N

(AA represents current ID code of the module, N represents the channel code currently being calibrated, 0-7).

4. Input current or voltage signal of 120% of FSR to the analog input module which needs to calibrate.

5. When the signal is stable, send ASCII character or MODBUS TCP gain calibration commands to the analog input module.

ASCII character: \$AA0N

Modbus TCP: 00 00 00 00 00 04 00 41 00 0N

(AA represents current ID code of the module, N represents the channel code currently being calibrated, 0-7)

6. Complete the Calibration.

Appendix: Dimension (Unit: mm)

