

95HXX Series Gas Transmitter Protocol V2.1

2017-08-02

Serial Port Setting Parameters: 9600, none, 8, 1.

All data is read and written in accordance with the MODBUS-RTU format.

Wiring rules

Red line: DC12~24V
Black line: DC Ground Wire (GND)
Yellow line: RS485A
Blue line: RS485B

1, Read Register (03 Code)

Host read register data format:

Address (1 byte) + function code (1 byte) + register address high byte + register address low byte + Read word high byte + read word low byte + CRC high byte check + CRC low byte check.

Slave reply data format:

Address (1 byte) + Function code (1 byte) + Return byte number (n) + Data 1 + Data 2+ Data 3+... + Data (n) + CRC high byte check + CRC low word Section check.

Remarks:

The readable registers of this transmitter can be read in two lengths. The fast mode reads only 2 words or 4 words at a time (suitable for more transmitters in the bus), and the full mode reads 10 words at a time.

E.g

Send: 01 03 00 00 00 02 C40B
 register byte CRC check

Return: 01 03 04 48 E1 28 42 23 94
 Concentration value CRC check

48 E1 28 42 Convert to single precision floating point number
42.220001

Send: 01 03 00 00 00 04 44 09
 register byte CRC check

Return: 01 03 08 48 E1 28 42 09 00 03 02
9D C5
 Concentration value Gas type Unit Alarm Decimal places
CRC check

Send: 01 03 00 00 00 0A C5 CD
 register byte CRC check

Return: 01 03 14 E1 FA 29 42 09 00 03 02
00 00 20 41

Concentration value Gas type Unit Alarm Decimal places
Measure range

00 00 00 3F 00 00 00 40 42 37
Low Alarm High Alarm CRC check

Gas concentration, range, Low alarm value and high alarm value are floating point numbers

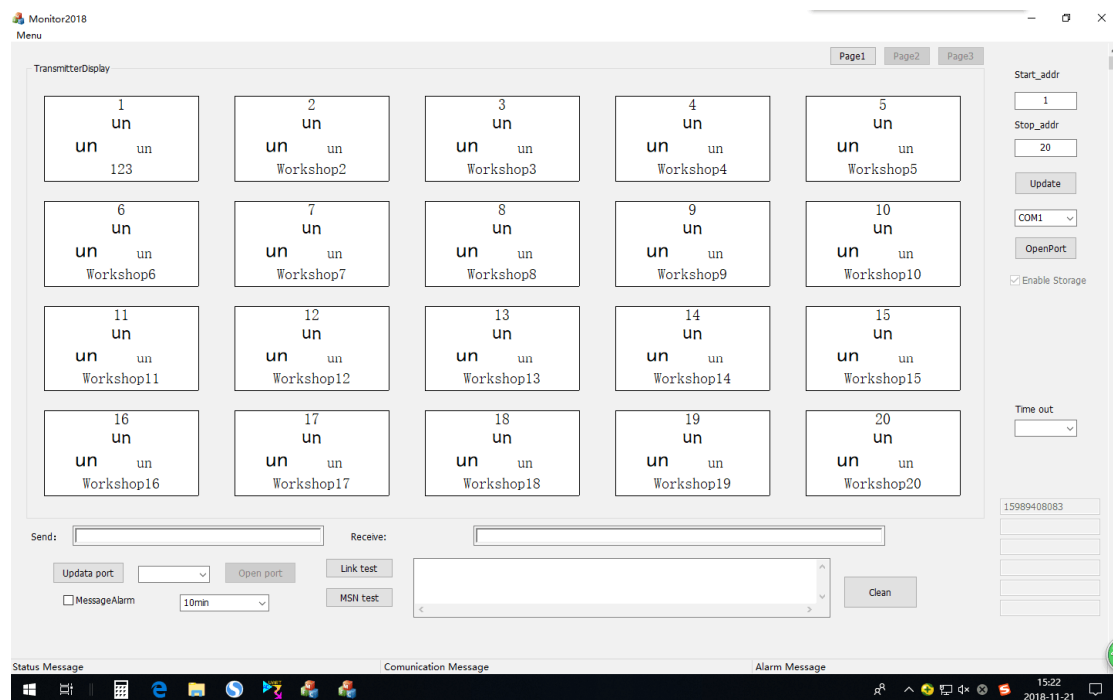
Use the union format when fetching data.

Set a union format variable

```
union
{
    char data[4];
    float F;
}Gas_Concentrate;// Define gas solubility register
```

When taking the number

```
Gas_Concentrate.data[3] = Data[0];
Gas_Concentrate.data[2] = Data[1];
Gas_Concentrate.data[1] = Data[2];
Gas_Concentrate.data[0] = Data[3];
Gas_Concentrate.F      It is the gas concentration we want.
```



Gas type vector table

Number	Gas type	Chinese	Number	Gas type	Chinese
--------	----------	---------	--------	----------	---------

0	N2	氮气	20	COS	氧硫化碳
1	CO	一氧化碳	21	CS2	二硫化碳
2	H2S	硫化氢	22	F2	氟气
3	CL2	氯气	23	HBr	溴化氢
4	NH3	氨气	24	HCL	氯化氢
5	SO2	二氧化硫	25	HF	氟化氢
6	H2	氢气	26	O3	臭氧
7	EX	可燃气体	27	PH3	磷化氢
8	NO	一氧化氮	28	SiH4	硅烷
9	NO2	二氧化氮	29	VOC	挥发性气体
10	C2H4O	乙醛	30	CH2O	甲醛
11	COCL2	光气	31	O2	氧气
12	AsH3	砷化氢	32	C4H8S	四氢噻吩
13	Br2	溴气	50	CO2	二氧化碳
14	C2H4O	环氧乙烷	51	CH4	甲烷
15	C2H2	乙炔	52	SF6	六氟化硫
16	C2H4	乙烯	53	CH3Br	溴甲烷
17	C2H6O	乙醇	54	CH4S	甲硫醇
18	CH3SH	甲硫醇			
19	CLO2	二氧化氯	100	Temp	温度
			101	Humi	湿度
			102	PRA	气压

The type of gas sent during data communication is reduced by 1000. For example, carbon monoxide is represented by 0x01. Hydrogen sulfide is represented by 0X02.

Unit vector table

Numerical value	Symbol
0	PPM
1	PPB
2	%
3	%LEL
4	Mg/m3
5	°C
6	RH%
7	Kpa

Alarm vector table

Numerical value	Symbol
0	no connection
1	normal

2	Low alarm
3	High alarm

In order to be compatible with other companies' modbus data acquisition protocols, the address offset function has been added to allow the customer's offset address to be 40000 (0x9C40).

Send 01 03 00 00 00 02 C4 0B And 01 03 9C 40 00 02 EB 8F return data is the same;

Send 01 03 00 00 00 04 44 09 And 01 03 9C 40 00 04 6B 8D return data is the same;

Send 01 03 00 00 00 0A C5 CD And 01 03 9C 40 00 0A EA 49 return data is the same.

2, Write register (06 yards)

Some parameters and functions of the transmitter are open to the outside world, allowing users to modify it themselves.

For example, the customer changes the address of the transmitter:

The host should modify the address of the transmitter with address 0X01 to 0X02, and the transmitted data is as follows (hexadecimal)

```

01          06          00          00
Address      function code  Register high byte  Register low byte
00          02          08          0B
Data high byte  Data low byte  High parity      Low calibration
  
```

The return data of the slave to the command is as follows

```

01          06          00          00
Address      function code  Register high byte  Register low byte
00          02          08          0B
Data high byte  Data low byte  High parity      Low calibration
  
```

If there are no exceptions, the returned data is exactly the same as the received data.

Otherwise, the error is reported - the error data also follows the MODBUS protocol.

06 Description of each register address under the code

Address	function	Remark
0x0012	Modify transmitter address	Call the input data
0x0008	Modify low alarm value	Call the input data
0x000A	Modify high alarm value	Call the input data
0x000E	Zero calibration	Do not call the input data
0x0000	Range	Call the input data

	calibration	
--	-------------	--

Note: The data written to the transmitter is 16-bit integer data, and the data read is floating point.

Example:

- 1, The initial address of the transmitter is 1, and the command sent to modify 2 is 0100060120002A80E.
- 2, The transmitter's low alarm value is changed to 200. The command sent is 0106000800C8099E.
- 3, The high alarm value of the transmitter is changed to 500. The command sent is 0106000a01F4A9DF.
- 4, The command for zero calibration of the transmitter is 0106000E0000E809
- 5, The transmitter's concentration is calibrated to 500. The command sent is 0106000001F489DD.

CRC Check form

```
/* CRC High byte value table */
const unsigned char auchCRChi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;
/* CRC Low byte value table*/
```

```

const unsigned char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
0x43, 0x83, 0x41, 0x81, 0x80, 0x40
} ;

```

Remark:

LED There are many options when entering the menu with the remote control with display model.

F-01 to F-12

Function	description	Remarks	Defaults
F-01	Transmitter address	RS485 signal transmitter address, set range with remote control 1-200	1
F-02	Low alarm value		
F-03	High alarm value		
F-04	Zero calibration	Calibration in clean air or high purity nitrogen	
F-05	4mA Fine tuning	4mA Fine current	50

		output, ranging from 3.8 to 4.2	
F-06	Programmer reserved		
F-07	Target concentration		
F-08	Calibration enable		
F-09	Production Date		
F-10	Communication protocol selection		
F-11	Serial data verification method		
F-12	Transfer order of floating point values		
F-13	Restore factory settings		

Thanks for visiting our sisco.com! Any questions, please contact us...

>>> Email: sales@sisco.com

>>> Phone: ++86 773-363-7977

>>> Website: www.sisco.com