7-in-1 air quality detection module

M701 product specification

Version: 3.0

M701 7-in-1 sensor module is a cost-effective digital serial port output sensor module, using 485/UART output mode, set CO2, formaldehyde, TVOC, laser dust PM2.5,PM10 particles, temperature, humidity in one, a variety of parameters will be in the form of RS485 digital interface unified output.It can carry out real-time comprehensive detection of the environment, has good stability and is very convenient for customers to use.

Application areas:

- Hotel room air quality monitoring
- Agricultural greenhouses, outdoor breeding place environmental monitoring
- Fresh air ventilation system
- Air purifier, air conditioning
- Air quality monitoring equipment
- Ventilation control system for kitchen and bathroom
- Smart home equipment

Main features:

- Carbon dioxide output at the same time. Formaldehyde.TVOC.PM2.5. PM10. Temperature.
- High sensitivity and stable data
- 485 output mode, can be connected to multiple sensors
- The temperature is accurate to 0.1℃, the humidity is accurate to 0.1%
- Automatic output of seven sensor monitoring data every 2 seconds through RS485/UART signal

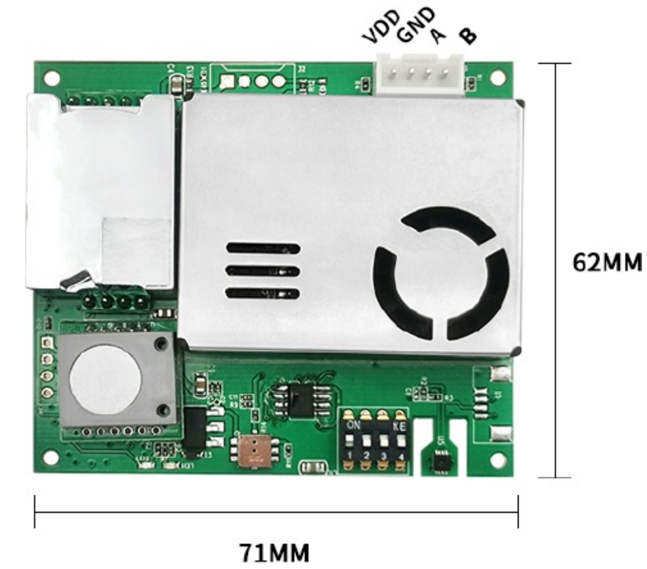
Specifications:

class Don’t	Measurement resolution	Measuring range	Accuracy of measurement
CO_2	1ppm	400 PPM to 5000 PPM	Plus or minus 3% + 50 PPM or plus or minus 10%
CH_2O	1 mu g/m3	0 ~ 2000 mu g/m after	Plus or minus 10%
TVOC	1 mu g/m3	0 ~ 2000 mu g/m after	Plus or minus 25%
PM2.5	1 mu g/m3	0 ~ 999 mu g/m after	Plus or minus or plus or minus 10 10%
PM10	1 mu g/m3	0 ~ 1000 mu g/m after after	Plus or minus or plus or minus 10 10%

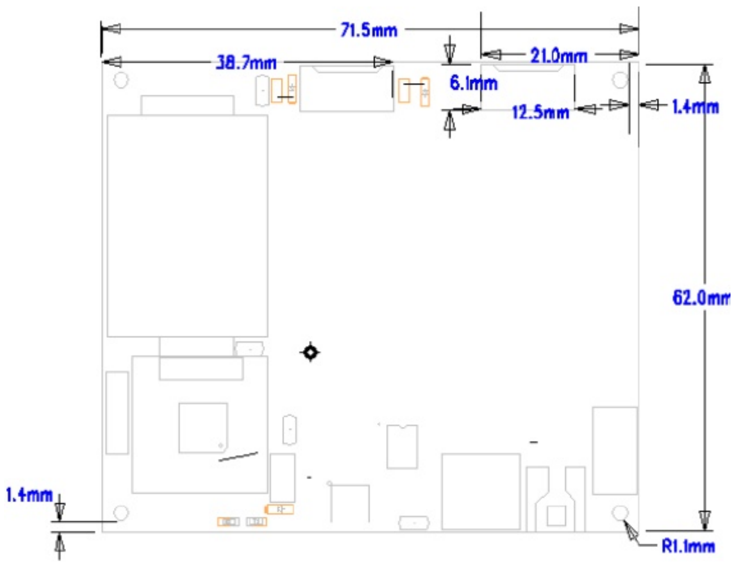
Temperature	0.1 ℃	- 40 ℃ ~ 100 ℃	+ / - 1 ℃
Humidity	0.1 %	0 ~ 100%	Plus or minus 3% RH
Physical interface	Mother XH2.54 seat		
The output data	RS485		
Working voltage	5.0 + / - 0.2 VDC		
Working current	500 ma or less (CO_2Transient needs 300 mA)		
Warm up time	2 minutes (CO only$_2$,CH_2O and TVOC need to be preheated, other parameters will be displayed when powered on)		
Working temperature	0 ℃ to 50 ℃		
Working humidity	95% RH or less		
Overall dimensions	71 * 62 * 16 mm (L * W * H)		

Appearance structure drawing:
(Height: 16MM)

备注：A/B 款仅为 PM2.5 传感器风口方向不一样



Appearance
size
diagram:



UART interface definition:

interface	The name of the	function
1	VDD	The power supply 5 v
2	GND	The power to
3	A	
4	B	

Serial port data stream format:

Baud rate	9600bps
Data bits	eight
Check digit	There is no
Stop bit	1

UART Communication protocol:

Word section	The name of the	instructions
B1	The frame head 1	A fixed value 3 ch
B2	The frame head 2	A fixed value 02 h
B3	data	eCO_2High byte
B4	data	eCO_2The low byte
B5	data	eCH_2O high byte
B6	data	eCH_2O low byte
B7	data	TVOC high byte
B8	data	TVOC low byte
B9	data	PM2.5 high byte
B10	data	PM2.5 low byte
B11	data	PM10 high byte
B12	data	PM10 low byte
B13	data	Temperature Integer part
B14	data	Temperature fractional
B15	data	Humidity is an integer
B16	data	Humidity is the same as Humidity.
B17	The checksum	The checksum

Note: The checksum B17 is equal to: B1+B2+····· B16, 8 bits lower.

When the bit7 of the temperature data B13 =1, it represents negative temperature; when the bit7 of B13 =0, it represents positive temperature.For example, when B13=9Bh, then bit7=1, indicating negative temperature, the actual temperature is -27℃;If B13=1Bh, then bit7=0, which means positive temperature, then the actual temperature is 27 degrees Celsius.

The attached:

M701 version 485 communication protocol

Ver2.0

1, an overview of the

The communication protocol describes the M701 input and output commands, information and data in detail for use and development by third parties. 1.2 Physical Interfaces:

The main communication port of the upper computer is connected with the standard serial RS-485 communication port.

The information transmission mode is asynchronous. The data bit is 8 bits, the stop bit is 1 bit, and there is no verification.

The default data transmission rate is 9600b/s

2. Details of M701 communication protocol

1) All loop communication shall follow the master/slave mode.In this way, information and data are passed between a single master station and a slave station (monitoring device).

2) Broadcast mode is not supported.

3) Under no circumstances can communication be started from a slave station.

4) If the master station or any slave station receives a package containing an unknown command, the package will be ignored and the receiving station will not respond.

2.2 returned data frame structure description each data frame consists of the following :(RTU mode) address

Function code

Number of data

Data 1

-

Data n

CRC 16-bit check

3. Transmission format

(1) Format of command packets

Host sends read data command:

address	Function code	Where the data starts Addressing a	Where the data starts Addressing low	Return the data The	Return the data The	CRC 16

		high		number of high	number of low	check
xx	03	00	02	0 0	0 7	Low in the first

(Currently, only all data can be read. Seven data values can be read starting from address 0002.) There are only seven data, corresponding to seven addresses, whose address and data high levels are not processed in the module. The default value is 0. The value is 0X00

Internal message

The starting address	The number of data	instructions
It can only be as follows:		
0x0002	1	CO_2The concentration of

0x0004	1	Formaldehyde concentration
0x0006	1	TVOC concentration
0x0008	1	PM2.5 concentrations
0x000A	1	PM10 concentration
0x000C	1	Temperature value
0x000E	1	Humidity value

The start address must be the address listed above. If the number of data is exceeded, no response is given. For example, if the starting address is 0X000C, the number of data can only be 1 or 2. If three data are returned, no response is given. Similarly, if the address 0X000E is used, the number of returned data can only be 1. If the number of returned data is 0, no response is received.

Slave sensor return value:

Slave IP address	Function code	Number of data	Data N	CRC
XX	03	xx	xx	xxxx

The byte length indicates only the data length.

N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13
CO2 high	CO2 low	formaldehyde high	formaldehyde low	TVOC high	TVOC low	PM2.5 high	PM2.5 low	PM10 high	PM10 low	The temperature high	The temperature low	humidity high	humidity low

Host sends read address command:

address	Function code	Where the data starts Addressing a high	Where the data starts Addressing low	The number of data high	The number of data low	CRC 16 check
00	02	00	00	00	02	The low value of Xxxx is in front

Slave return address:

address	Function code	bytes	Address high	Address the low	CRC 16 check
00	02	02	00	xx	The low value of Xxxx is in front

Set slave address command :(set slave address to 2)

address	Function code	Data starts at the high level of the address	Data starting address low value		address	CRC 16-bit check
00	06	0 0	0 0	0 0	02	The low value of Xxxx is in front

4. Host data sampling frequency:

When reading data from the t/h sensor, the upper computer reads data at an interval of at least 500ms (1s is recommended).

5. Function code

03: Reads data

02: Reads the address

06: Setting the address (1 to 255)

6. Command examples:

Serial port Settings: asynchronous communication, data bit 8, no parity check, stop bit 1 data transmission rate the default value is 9600b/s

01 03 00 02 00 07 CRCL, CRCH

M701 returns:

0 x01, 0 x01, 0 x0e, CO_2H,CO_2L H L, formaldehyde, formaldehyde, TVOCH TVOCL, PM2.5 H, PM2.5, L PM10H, PM10L, temperature of H, L, temperature, humidity, H, L, humidity CRCL, CRCH

Case 1:

TX: 01 03 00 02 00 07 CRCL, CRCH

RX: 01 03 0E 01 E2 00 05 00 24 00 2D 00 38 01 31 02 86 CRCL, CRCH CO_2

Concentration = CO_2H x 256 + CO_2L PPM (BYTE3×256 + BYTE4)
formaldehyde concentration = formaldehyde H ×256 + formaldehyde L µg
(BYTE5 ×256 + BYTE6) TVOC concentration = TVOCH ×256 + TVOCL µg
(BYTE7) ×256 + BYTE8) PM2.5 concentration = PM2.5H x 256 + PM2.5L µg
(BYTE9 x 256 + BYTE10) PM10 concentration = PM10H x 256 + PM10L µg
(BYTE11 x 256 + Temperature = ((BYTE13)×256 + (BYTE14)) / 10
Humidity = ((BYTE15) x 256 + (BYTE16)) /10

The above CO_2= 5 = 482, formaldehyde, TVOC = 36, PM2.5 = 45, PM10 = 56, = 30.5 temperature and humidity = 64.6

Example 2:

TX:01 03 000C 00 02 CRCL, CRCH (10:43:52:001) (Reads two data starting from address 01 000C) 0x000C, corresponding to temperature and humidity.

RX:01 03 02 01 27 02 45 03 57 (10:43:52:159)

The returned temperature 0X0127 corresponds to base 295, indicating that the temperature is 29.5 ° C. The returned humidity 0X0245 corresponds to base 581, indicating that the humidity is 58.1%RH

Note:

Returns each data format from the machine

Temperature is a 16-bit signed number and other data is a 16-bit unsigned number.The temperature and humidity are converted to base 10 and divided by 10, which is the test value.

The temperature 0X0127 corresponds to base 295, indicating 29.5 ° C

The return temperature of 0xFFF4 pairs is 12 in base 10, indicating the temperature －1.2℃

Such as:

Returned humidity hexadecimal unsigned data: 0X0311, corresponding to decimal 785, indicating that the humidity is 78.5%RH Return temperature

Hexadecimal signed data: 0X00FF, corresponding to decimal 255, indicating that the temperature is 25.5 ° C Return temperature Hexadecimal signed data: 0xFFF4 corresponds to base 10 12, indicating temperature -1.2℃

Set
address:

Address CRCL, CRCH transmitter: 00 06 00 00, 00,
address CRCL, CRCH

Read address
value:

00 02 02 00, address, CRCL, CRCH

CRC check
reference:

0x01, 0xC0,

CRC high byte value table: static char auchCRCHi[] = {0x00, 0xC1, 0x81, 0x40,0x80, 0x41,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x00,0x01, 0xC0,
0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x80, 0x41,0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x00, 0xC1, 0x81,0x00, 0xC1,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x81, 0x40, 0x01, 0xC0,0x81, 0x40,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x01, 0xC0, 0x80, 0x41, 0x00,0x01, 0xC0,
0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,0x00, 0xC1,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,0x80, 0x41,
0x80, 0x41, 0x01, 0xC0, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,0x00, 0xC1,
0xC0, 0x80, 0x41, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,0x81, 0x40,
0x00, 0xC1, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80,0x00, 0xC1,
0x41, 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, 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};

CRC low byte value table: static 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, 0xF5, 0xF7, 0x37,
0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0x3B, 0xFA, 0x3A,
0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0xEE, 0x2A, 0xEA,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0xE6, 0x27, 0xE7,
0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x63, 0xA3, 0x61, 0xA1,
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, 0x97, 0x56,
0x57, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x9A, 0x5E,
0x5A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x4A, 0x8B,
0x8A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C, 0x44, 0x84, 0x85, 0x47, 0x45,
0x87, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40};
```

CRC function
calculation method:

1.Preset a 16-bit register to hexadecimal FFFF (that is, all 1);Call this register a CRC register; 2.The first 8-bit binary data (i.e. the first byte of a communication frame) is different or from the lower 8 bits of a 16-bit CRC register, and the result is placed in the CRC register;

3.Move the contents of the CRC register one right (toward the low) and fill the highest bit with a 0, and check the move out bit after the right move; 4.If move out is 0: repeat step 3 (move right again one bit); If the offbit is 1, the CRC register is xOR with multientry A001 (1010 0000 0000 0001);

5.Repeat steps 3 and 4 until the right shift is 8 times so that the entire 8-bit data has been processed;

6.Repeat Step 2 to Step 5 to process the next byte of the communication information frame.

7.After all bytes of the communication information frame are calculated according to the above steps, the high and low bytes of the 16-bit CRC register obtained are exchanged;

8.The final CRC register content is: CRC code.

CRC function routines: //*pushMsg is the array pointer variable to be checked, usDataLen is the number of data variables to be checked

```
void CRC16(char *pushMsg,unsigned short usDataLen) {


char uchCRCHi=0xFF;              // High CRC byte initialization
char uchCRCLo=0xFF;              // Low CRC bytes initialize
unsigned int uIndex;             // index in CRC loop
while(usDataLen--)
uIndex=uchCRCHi^*pushMsg++;/ / CRC
calculation
uchCRCHi=uchCRCLo^auchCRCHi[uIndex];
  uchCRCLo=auchCRCLo[uIndex];
      }
  *pushMsg++=uchCRCHi;           // Check data is higher
  than that
    *pushMsg=uchCRCLo;           }}
```

Matters needing attention

To power on a device for the first time, preheat the device for at least five minutes.

Unconsciously, do not apply this module to systems that involve personal security. Unconsciously do not expose the module to high levels of organic gas for long periods of time.

Anyway, do not install a module in a
strong-air convection environment.

Wired the metal shell plugs into the sensor's internal power supply, taking care not to short-circuit other external circuits or the chassis shell.

The optimal installation method is to put the air inlet and outlet in a plane that is close to the user's inner wall to connect with the outside world, and there is no shelter within 2cm around the air outlet.There should be air isolation between the air inlet and the air outlet to avoid direct flow of air back to the air inlet from the air outlet inside the equipment.

The hole sizes of the air inlet and outlet of the device should not be smaller than the hole sizes of the air inlet of the sensor.

When using something like a purifier, users avoid putting sensors directly into the purifier's own air duct by designing a self-supporting structural space where the sensor is placed, insulating it from the purifier's own air duct.

The sensor should be installed 15 to 20CM higher than the ground. Otherwise, the fan may be polluted by large dust particles such as ground dust, floating flocs or even flocs. You are advised to use the device to prefilter the fan.

Users automatically do not disassemble the sensor, including the metal shield, to prevent irreversible damage.

The sensor data ensures consistency between manufacturers, without using third-party instrumentation or data as a comparison standard.If the user wants the final measurement result to be consistent with a third-party testing device, the user can perform data fitting and calibration according to the actual collected results.

The sensor lends itself to a common indoor environment, where the user's device can lose data consistency due to excessive dust, oil and water accumulation:

A) Annual dust concentration greater than 300 μ g/m3 for more than 50% of the time, or greater than 500 μ g/m3 for more than 20% of the time; B) Lampblack environment;C) High water mist environment;D) outside.