AOSONG

Digital temperature and humidity sensor

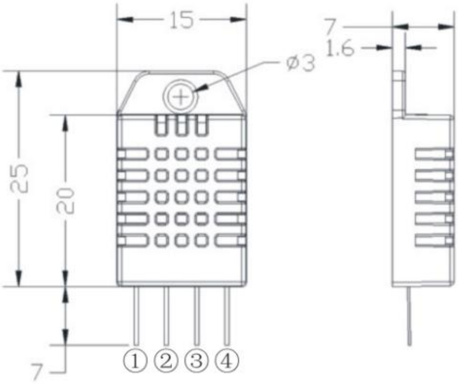
AM2312 Product Manual

www.aosong.com

1、Product Overview

AM2312 capacitive humidity sensor is a digital temperature and humidity have been calibrated digital signal output with temperature and humidity sensor complex. Temperature and humidity using a dedicated capture technology to ensure product with high reliability and excellent long-term stability. Sensor includes a capacitive sense element and a high-precision integrated wet-temperature components, and connected with a high-performance microprocessor. Therefore, the product has excellent quality, fast response, anti-interference ability, high cost and so on. AM2312 communication standard I^{2}C interface. I^{2}C communication standard communication timing, the user can communicate directly hung on the I^{2}C bus, no additional wiring, simple to use. Direct communication data output by the temperature compensation of the humidity, temperature, and Check (CRC) and other digital information, users do not need a second calculation of the digital output, and no need for temperature compensation of the humidity, temperature and humidity can get accurate information. The communication easy to use, wide range of applications. 4 lead product, easy connection, special packages can be provided according to user needs.

Physical map



Dimensions (unit: mm)

2、Applications

HVAC air conditioners, dehumidifiers, testing and inspection equipment, consumer goods, automotive, automation, data loggers, weather stations, home appliances, humidifiers, medical, and other related humidity measurement and control.

3、Features

Completely interchangeable, low cost, long-term stability, relative humidity and temperature measurement, long distance signal transmission, digital signal output, precise calibration, low power consumption, standard I^{2}C bus digital interface.

4、Interface definition

4.1 AM2312 pin assignment

Table 1: AM2312 pin assignment

Pin	Color	Name	Description
1	Red	VDD	Power (3.3V-5.5V)
2	Yellow	SDA	Serial data, bidirectional
3	Black	GND	Ground
4	White	SCL	Serial Clock, input

4.2 Power supply pins（VDD GND）

AM2312 supply voltage range of 3.3 - 5.5V, recommended supply voltage is 5V.

4.3 Data line（SDA SCL）

SCL I^2C communication signal line when the clock line, SCL used for communication between the microprocessor and the AM2312 synchronous. SDA pin is tri−state structure for reading, writing sensor data. Specific communication timing, see a detailed description of communication protocols.

5、Sensor Performance

5.1 Relative Humidity

Table 2: AM2312 relative humidity performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		%RH
Range		0		99.9	%RH
Accuracy [1]	25℃		±3		%RH
Repeatability			±0.3		%RH
Exchange		Completely interchangeable			
Response [2]	1/e(63%)		<5		S
Sluggish			<0.3		%RH
Drift [3]	Typical		<0.5		%RH/yr

5.2 Temperature

Table 3: AM2312 relative temperature performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		℃
			16		bit
Accuracy			±0.3	±1	℃
Range		−40		80	℃
Repeat			±0.2		℃
Exchange		Completely interchangeable			
Response	1/e(63%)		<10		S
Drift			±0.3		℃/yr

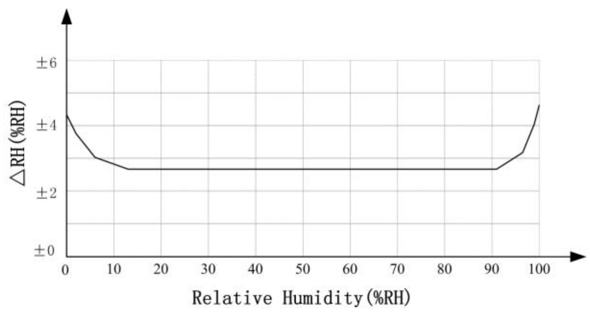


Figure 2: 25 ℃ relative humidity when the AM2312 maximum error

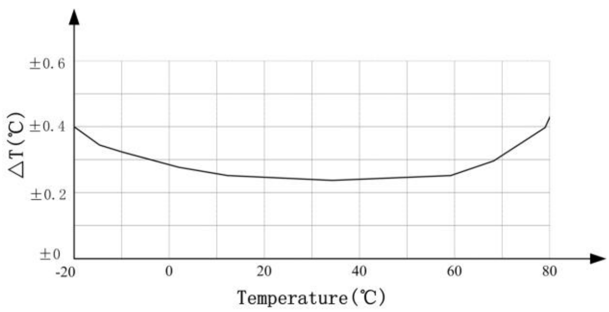


Figure 3: The maximum temperature error of temperature sensor

6、Electrical Characteristics

Electrical characteristics, such as energy consumption, high, low, input, output voltage, depends on power. Table 4 details the AM2312 electrical characteristics, if not stated otherwise supply voltage of 5V. For best results with the sensors, design strictly in accordance with Table 4 and Table 6 of the conditions of the design.

Table 4: AM2312 DC Characteristics。

Parameter	Condition	min	typ	max	Unit
Supply voltage		3.3	5	5.5	V
Power [4]	Dormancy	10	15		μA
	Measuring		500		μA
	Average		300		μA
Low-level output	I_{OL} [5]	0		300	mV
High output	Rp<25 kΩ	90%		100%	VDD
Low input voltage	Decline	0		30%	VDD
Input High	Rise	70%		100%	VDD
Rpu[6]	VDD = 5V VIN = VSS	30	45	60	kΩ
Output current	On		8		mA
	Off	10	20		μA
Sampling period		2			S

[1] The accuracy of the factory test, the sensor at 25 ℃ and 5V, the accuracy of indicators under the conditions tested, it does not include hysteresis and non-linear, and only for non-condensing environment.
[2] at 25 ℃ and 1m / s air flow conditions, to reach 63% of first-order response time required.
[3] in volatile organic compounds, the values may be higher. See manual application to store information.
[4] This value is VDD = 5.0V, when the temperature is 25 ℃, 2S / time, average conditions.
[5] low output current.
[6] that pull-up resistor.

7、I^2C Communication protocol

7.1 I^2C communication protocol introduced

7.1.1 I^2C About the bus

AM2312 and micro-processor control interface form of the I^2C serial bus, in this brief introduction about the I^2C bus protocol standard. Due to space limitations, the agreement can not list the entire contents of a deeper problem, please refer to the relevant information (refer to the Philips site inspection).

7.1.2 I^2C Bus Overview

Philips (Philips) at 20 years ago invented a simple two-way two-wire serial communication bus, the bus is known as the Inter-I^2C bus. I^2C bus has now become the industry standard solution for embedded applications, is widely used in variety of microcontroller-based professional, consumer and telecommunications products, as a control, diagnostics and power management bus. More in line with standard I^2C bus devices can communicate with an I^2C bus, without the need to address decoder.

I^2C bus only composed by the two signal lines, a serial data line SDA, the other root is the serial clock line SCL. I^2C bus devices generally have their SDA and SCL pins are open-drain (or open collector) output structure. Therefore, the actual use, SDA and SCL signal lines must be pull-up resistor (Rp, Pull-Up Resistor). Pull-up resistor on the general value of 3 ~ 10 kΩ . Therefore, when the bus is free, the two signal lines remain high, almost no current consumption; electrical compatibility, and supports a variety of different voltage logic interface; different between the two can be directly connected to the bus, not require additional conversion circuitry to support a variety of ways a master multi-slave communication is the most common means of communication. It also supports dual-host communications, multi-host communications, and broadcast mode, and so on.

I^2C typical configuration is shown in Figure 4。

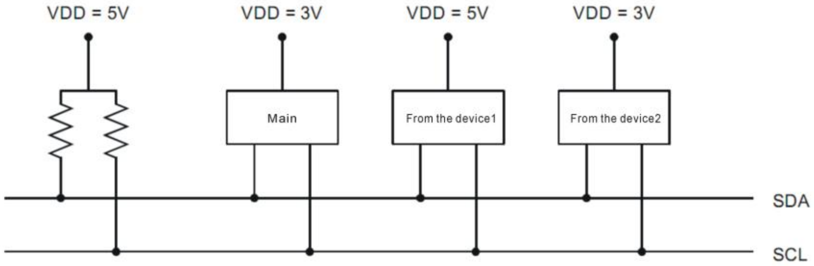


Figure 4：I^2C typical configuration

7.1.3 I^2C Bus protocol specification

◎ **I^2C Bus Definition of Terms**

I^2C−bus serial data (SDA) and serial clock (SCL) line connected to the bus, allowing the device to transmit information between each device has a unique address identification, and can be used as a transmitter or receiver (the device features decision), the device performs data transmission can also be seen as a master or a slave, the host is initialized bus allows data transfer and generates the clock signal transmission device. At this point, any device addressed is considered a slave. Details of the I^2C bus definition of terms in Table 5.

◎ **I^2C Bus transfer rate**

I^2C Bus communication speed controlled by the host, to fast to slow. However, the maximum rate is limited, I^2C bus data transfer rate up to the standard mode, fast 100Kb /s.

◎ **I^2C Bus bit transmission**

I^2C Bus bit transmission is through the data line SDA and SCL line to complete the two lines together. In the high period of SCL clock line, data line SDA low logic level, said current transmission "0"; in the high period of SCL clock line, data line SDA high logic level, said current transmission "1." Logic "0" (low) and "1" (high) level, is related by the VDD voltage level (for details see Table 4 AM2312 DC Characteristics table). In addition, each data bit transferred on to generate a clock pulse.

Table 5：I^2C Bus definitions of terms

Term	Description
Transmitter	Devices to send data to the bus
Receiver	Device receiving data from the bus
Host	Initialization and termination of the clock signal sent sent produce devices
From machine	Addressed by the host device
Multi−host	At the same time try to control more than one host, but does not destroy the message bus.
Arbitration	It's a multiple hosts simultaneously try to control the bus but only one is allowed to control the bus and make the message is not destroyed in the process.
Synchronous	Two or more devices synchronized clock signal in the process

◎ **The validity of data**

Data line SDA of the data must be in the HIGH period of the clock remain stable. SDA data line high or low state only in the clock line SCL low time was allowed to change. But in the beginning and end of the I^2C bus when the exception (for details see Start and Stop conditions). Some other data may require the serial bus clock signal edge (rising or falling edge) is valid, but the level I^2C bus is valid. Specific timing diagram shown in Figure 5.

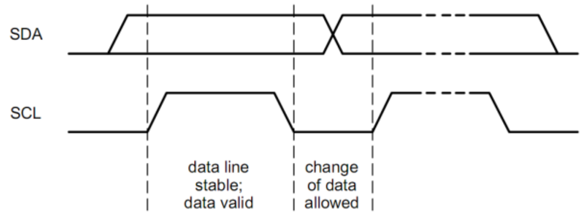


Figure 5： I^2CBus bit transmission

◎ **Start and stop conditions**

Initial conditions： During the time when SCL is high, SDA from high to low transition generated when the initial conditions. Produced in the initial conditions after the bus is considered busy. Initial conditions that are often denoted by S.

Stop condition： During the time when SCL is high, SDA from low to high transition when a Stop condition. Generate a stop condition after the bus is idle. Stop condition denoted by P.

Start and stop conditions diagram shown in Figure 6。

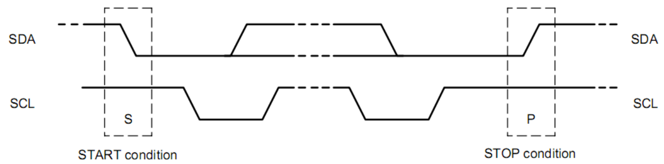


Figure 6： Schematic diagram of the start and stop conditions

◎ **Byte transfer format**

I^2C Bus to send and receive data bytes. Transferred to the SDA line must be 8−bit per byte. The number of bytes per transfer is unrestricted. First, the data transmission is the highest bit (MSB No. 7), the last transmission is the lowest bit (LSB, bit 0). In addition, each byte must be followed by an acknowledge bit (ACK). I^2C transmit data in Figure 7.

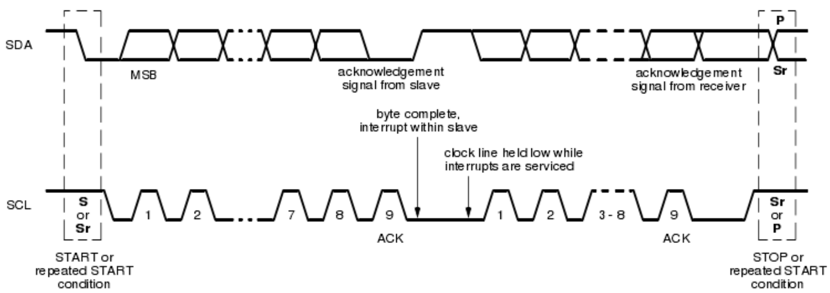


Figure 7： I^2C Bus data transfer

◎ **I^{2}C Bus response**

I^{2}C−bus transfers data in the process, each transmission of a byte, must be answered with a status bit. The case of the receiver can receive data acknowledge bit to inform the transmitter. Acknowledge bit, still the master generates the clock pulse, and an acknowledge bit, the data state is follow the "who who receives a" principle, which always generates the acknowledge bit by the receiver, in response to the clock pulse period the receiver must pull the SDA line low, making it the clock pulse is stable LOW during the HIGH period (see Figure 8), of course, setup and hold times must be considered (for details see table 6). Send data from master to slave, the slave generates the acknowledge bit; host to receive data from the slave, the acknowledge bit by the master.

I^{2}C bus standard: 0 acknowledge bit to the receiver acknowledge (ACK), often abbreviated as A; of 1 indicates non−response (NACK), often abbreviated to NA. After the transmitter sends the LSB should release the SDA line (pulled SDA), to wait for the receiver generating an acknowledge bit.

If the receiver is receiving last byte of data, or can not receive more data, it should produce non−ACK to notify the sender. If you find the receiver transmitter generates a non−response state, you should terminate the transmission.

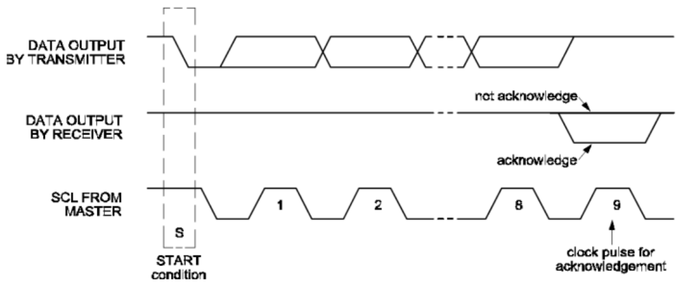


Figure 8： I^{2}CBus response

◎ **Slave Address**

No additional I^{2}C bus address decoder and chip select signals. Multiple devices with I^{2}C bus interface can be connected to the same I^{2}C bus, and address them through the device to distinguish between. The process of addressing the I2C bus is usually the initial conditions in the first byte after the decision to choose which one from the host machine, that is 7−bit addressing address (the other is 10−bit addressing address is different, the sensors 7−bit addressing address). The first byte of the bit definitions shown in Figure 9, the first byte of the first seven bits of the slave address, the lowest bit (LSB) is No. 8. It determines the direction of the message, the first byte of the least significant bit (LSB) is "0": Indicates that the host will write the information to be selected from the machine; "1" indicates that the master will read information from the machine.

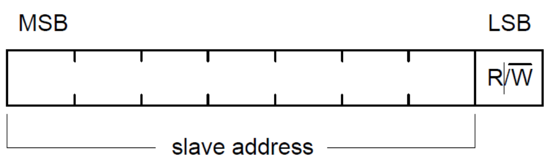


Figure 9： After the initial conditions of the first byte

Sent an address, the system of each device in the initial conditions, the first seven and more if its own address as the device will think it is the host address, as a slave receiver or slave transmitter by the R / W bits. Host is the master Parts, it does not need the device device address, and other devices are all from the machine, have the device address. Must ensure that all on the same I^{2}C bus slave addresses are uniquely determined, can be repeated, or the I^{2}C bus will not work.

◎ **Schematic diagram of the basic data transfer format**

Figure 10 and Figure 11 are given the I²C data transmit and receive the basic format. It should be noted, Figure 11 and Figure 12 is different, in Figure 11, the host sends to the last byte from the data, the response from the machine may or may not answer, but in any case the host can generate stop condition. If the host sends data to the slave (even including slave address) when the detected non−response from the machine, you should promptly stop the transmission.

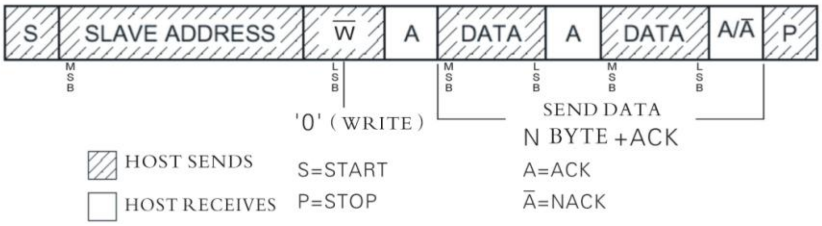


Figure 10: I²C The bus master sends data to the basic format from

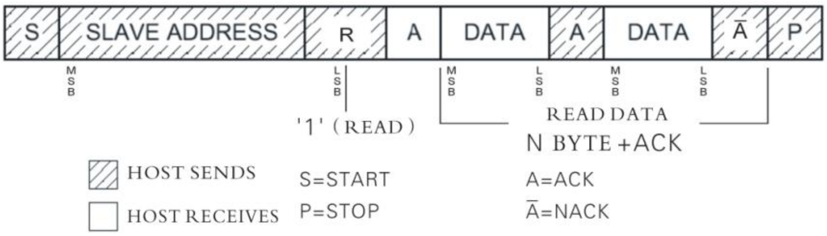


Figure 11: I²CThe bus master receives data from the slave's basic format

7.2 AM2312 Sensor I²C communication protocol

AM2312 I²C bus serial interface, in full accordance with the standard I2C protocol addressing, can be directly linked to the I²C bus. AM2312 sensor I2C address (SLAVE ADDRESS) is 0xB8, the I²C bus protocol based on the standard, based on ModBus protocol, developed a unique communication protocol, reducing the transmission error rate. Microcontroller read AM2312 sensor, the sensor I²C_ModBus in strict accordance with AM2312 communication protocol and timing design.

7.2.1 I²C Interface Description

AM2312 digital temperature and humidity sensor as a slave, and the host (user microprocessor) way communication between the I²C bus standard mode. For the accurate measurement of humidity, temperature and humidity to reduce the impact, AM2312 sensors in the non−working period, automatically be converted to sleep, to reduce the work consumption, to reduce the sensor self−heating of the humidity of the surrounding environment. AM2312 uses passive mode, that is, the host through the instruction wake−up sensor, and then sends the appropriate commands, read the corresponding values of temperature and humidity; communication after the acquisition of temperature and humidity sensor is triggered once; so if the long did not read the sensor, please read the two consecutive second sensor (minimum interval of two to read 2S), the second is the latest measured value; collected after the end of the sensor automatically be converted to sleep. Next host to be read sensor, the need to re−awaken the sensor. Must be noted that host communication from start to finish, for a maximum of 3S. If communication is not completed within 3S, sensors automatically end communication, automatically be converted to sleep, read again the host, such as sensors, need to re−send the wake−up command.

7.2.2 I²C Interface Features

This section describes the AM2312 Sensor I²C interface characteristics, if you want to get the best communications with the sensor results, designed strictly in accordance with Figure 12 and Table 6 of condition design.。

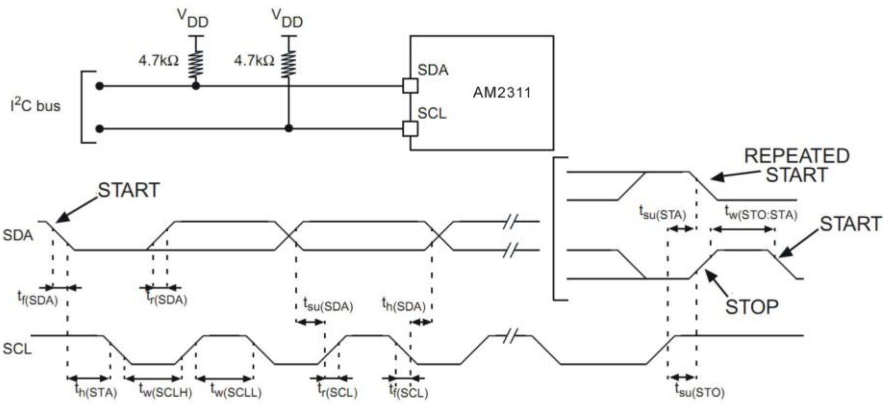


Figure 12： AM2312 Typical application circuit and the I2C bus timing diagram

Table 6： AM2312 Sensor I²C interface characteristics

Symbol	Parameters	Standard I²C mode		Unit
		min	max	
SCLClock frequency			100	kHz
tw(SCLL)	SCLClock low time	4.7		μs
tw(SCLH)	SCLClock high time	4.0		
tsu(SDA)	SDA Setup time	250		ns
th(SDA)	SDA Data hold time	0[1]		
tr(SDA) tr(SCL)	SDA & SCL Rise time		1000	
tf(SDA) tf(SCL)	SDA & SCL Fall Time		300	
th(STA)	Start condition hold time	4.0		μs
tsu(STA)	Repeated START condition setup time	4.7		
tsu(STO)	Stop condition setup time	4.0		μs
tw(STO:STA)	Stop to Start condition time (bus free)	4.7		μs
Cb	Capacitive load for each bus		400	pF

7.2.3 Communication protocol

AM2312 Sensor I²C communication protocol is a standard I²C bus protocol based on the reference ModBus protocol, the sensor according to AM2312 own characteristics, a combination of I²C_ModBus agreement. The following format:：

◎ Communication data (information frame) format

Data format:	I^2CAdd+R/W	Function code	Data area	CRC Check [3]
Data length:	1Byte [2]	1Byte	NByte	16-bit CRC (cycle redundancy code)

[1] If the interface does not allow to extend the time low, only need to comply with the longest hold time Start condition.

[2] A byte consists of 8-bit binary number (both 8 bit).

[3] CRC checksum algorithm, for details see: CRC code calculation method; detailed below.

◎ Communication and information transfer process

When the communication command from the sending device (host) to the sensor, I^2C address of the command line with the sensor, the sensor was to receive, and in accordance with the relevant requirements of the function code and read the information; and the implementation of the results (data) back to the host. The information returned includes the function code, data and after the implementation of the CRC code (user time to read the CRC, stop conditions can be sent directly).

◎ Communication I^2C slave address

AM2312 I^2C addresses are the same for each sensor, and is 0xB8. Therefore, in the same bus can only be linked to a AM2312 sensor, the sensor only after receiving the start signal and the same with its I^2C address only to respond to the host.

◎ I^2C communication function code

Communication of information for each function code is the first byte of the frame transmission. I^2C_ModBus communication rules, define the function code is 1 to 127. As a host request, through the function code tells the slave what action should be implemented. Response as a slave, the slave returns the function code and the functions from the host to send the same code, it indicates that the response from the host machine and have been associated operations. I^2C_ModBus part of the function codes described in Table 7.

Table 7: I^2C_ModBus Part of the function code

Function code	Definition	Operations (binary)
0x03	Read register data	Read data from one or more registers
0x10	Write multiple registers	Multiple sets of binary data is written to multiple registers

◎ I^2C communication data area

Data area including the need to return to what information from the sensor or perform any action. This information can be data (such as: temperature, humidity, the sensor device information, user-written data, etc.), the reference address. For example, the host told the sensor 03 through the function code value return register (read register contains the starting address and length of register read), the returned data includes the length of the data register and data register contents.

I^2C_Modbus sensor uses a custom communication protocol, host, use of communication commands (function code 03), can read the data register any of its data register table shown in Table 8. Sensor temperature and humidity data register stores the value and the corresponding sensor signal equipment and other related information; each data register is a single byte (8 bits) of binary data; read sensor, up to 10 registers of data, more than reading take the length of the sensor will return the corresponding error code. Error code information, see Table 1.

Table 8：AM2312 Data register table

Register information	Add	Register information	Add	Register information	Add	Register information	Add
High RH	0x00	Model high	0x08	Users register a high	0x10	Retention	0x18
Low RH	0x01	Model low	0x09	Users register a low	0x11	Retention	0x19
High temp.	0x02	Version number	0x0A	Users register 2 high	0x12	Retention	0x1A
Low temp.	0x03	ID(24−31) Bit	0x0B	Users register 2 low	0x13	Retention	0x1B
Retention	0x04	ID(16−23) Bit	0x0C	Retention	0x14	Retention	0x1C
Retention	0x05	ID(8 − 15) Bit	0x0D	Retention	0x15	Retention	0x1D
Retention	0x06	ID(0 − 7) Bit	0x0E	Retention	0x16	Retention	0x1E
Retention	0x07	Status Register	0x0F	Retention	0x17	Retention	0x1F

◎ **Temperature output format**

Temperature resolution is 16Bit, the maximum temperature position (Bit15) equal to 1 indicates a negative temperature, the temperature highest (Bit15) is equal to 0 for positive temperature; temperature in addition to highest (Bit14 ~ Bit0) that string out of the temperature sensor. String out of the temperature sensor is 10 times the actual temperature；

◎ **Status Register**

Status register, Bit7−Bit0 bit, temporarily reserved

Status register bits	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Features	Retention							

◎ **I^2C_ModBus Function Code Description**

一、**Function code "03"**：Read multiple sensors register

Host sends a read frame format：

START+ (I^2CAddress+W)+ Function code（0x03）+ Start address + Register number +STOP

The host to read back the data：

START+ (I^2C Add+R)+ Continuously read sensor data returned +STOP

Sensor response frame format：

Function code（0x03）+ Register number + Data +CRC[1]

for example： The host continues to read sensor data: 0x00 start address register of the four sensor data.

Sensor data register address and data：

Register add	Register data	Data on	Register add	Register data	Data on
0x00	0x01	High RH	0x02	0x00	High temp.
0x01	0xF4	Low RH	0x03	0xFA	Low temp.

Host sends the message format：

Host sends	Bytes	message	Remarks
Sensor add	1	0xB8	Sensor I^2C address（0xB8）+W（0）
Function code	1	0x03	Read the register
Start add	1	0x00	Register start add 0x00
Register No.	1	0x04	Read the register number

Sensor response to the return message format：

Response from machine	Bytes	Send	Remarks
Function code	1	0x03	Read the register
Returns No. of bytes	1	0x04	Back 4 of 4 bytes register
Register 1	1	0x01	Add 0x00 of the content (RH high byte)
Register 2	1	0xF4	Add 0x01 of the content (RH low byte)
Register 3	1	0x00	Add the content of 0x02 (temp. high byte)
寄存器 4	1	0xFA	Add 0x03 of the content (temp. low byte)
CRC 码	2	31A5	Back to the CRC, the low byte first ;

[1]： Details of the CRC calculations see the back of the CRC introduced, the sensor returns all the data are CRC check, the user can choose to read or not read。

Numerical：

Read back from the sensor temperature and humidity values, as long as the value into a decimal number 10 is divided by the value corresponding to the temperature and humidity, the temperature of the corresponding unit ℃, humidity units of % RH. For example, read back the data above:

Humidity: 01F4 = 1×256+15×16+4 = 500 => Humidity = 500÷10=50.0%RH;

Temp.: 00FA= 15×16+10 = 250 => Temp. = 250÷10 = 25.0℃

➘ Note： CRC verification code CRC is calculated by the code to arrive, and then pass the CRC and the sensor compared; the same, the data is received correctly, or that the data has errors.

二、 **Function code "10":** write multiple registers to sensor

The host can use this function code multiple data stored in the register to the sensor. AM2312 sensors register a single one-byte or 8 bits. Sensors allow you to save a maximum of 10 data registers. Therefore, the host single sensor to save up to 10 registers. More than 10, the sensor will return the corresponding error code.

Host sends write frame format：

START+(I^2C Add+W)+ Function code (0x10)+ Start address + Register number + Save the data +CRC+STOP

Confirm the host to read instructions：

START+(I^2C Add+R)+ Read the sensor data returned +STOP

Sensor response frame format：

Function code（0x10）+ Start Add + Register number +CRC

For example： Should save the address of the host 01, 02 10, 11 of the sensor to register.

Host sends the message format:

Host sends	Bytes	Send	Remarks
Sensor Add	1	0xB8	Sensor I^2C Add （0xB8）+W（0）
Function code	1	0x10	Write multiple registers
Start Add	1	0x10	To write the start address register
Byte length	1	0x02	Save the data word length (2 Byte)
Data 1	1	0x01	Save the data (address: 10)
Data 2	1	0x02	Save the data (address: 11)
CRC Code	2	C092	Host calculated CRC code, low byte first (I^2C addresses not included in the CRC calculation)

Sensor response to the return message format：

Response from machine	Bytes	Send message	Remarks
Function Code	1	0x10	Write multiple registers
Start address	1	0x10	Save the start address
Save the data length	1	0x02	Sensors save data length
CRC Code	2	FC04	Sensors return the CRC calculation, the low byte first ;

◎ CRC Check

Host or check code can be used to discriminate the sensor to receive information is correct. As the electronic noise or some other interference, the information during transmission error sometimes occurs, the error check code (CRC) can verify that the host or sensor data transmission process in the communication of information is wrong, wrong data can be discarded (either send or receive), which increases the system's security and efficiency.

I^2C_ModBus protocol of CRC (cycle redundancy code) consists of 2 bytes, or 16−bit binary number. CRC code from the sending device (host), is placed in the rear send a message frame, I^2C address is not included in the CRC calculation. CRC are either sent or received by the low byte first, high byte format after sending.

Receive information (sensors) and then recalculate the information received by the CRC, the CRC is calculated by comparing the receiver to match, if they do not match, then the error. Users need special

Do note that the sensor reading instruction without adding the CRC; write sensor, the CRC must increase; and all the return data are CRC.

◎ CRC code is calculated

1. Preset a 16-bit registers as hexadecimal FFFF (that is, all 1); call this register is the CRC register;
2. The first 8-bit binary data (communication of information both the first byte of the frame) and 16-bit CRC register, or the lower 8 bits are different, the results put in the CRC register;
3. The CRC register right one (towards low) with 0 fill the highest place, and check out right after the bit;
4. If out of the bit is 0: Repeat Step 3 (again shifted to the right one); if out of place as 1: CRC register with the polynomial A001 (1010 0000 0000 0001) XOR;
5. Repeat steps 3 and 4, until the right 8 times, so that all of the 8-bit data were processed;
6. Repeat steps 2 through 5, the communication frame to the next byte of information processing;
7. All bytes of the communication of information frames calculated according to the above steps completed, the resulting 16-bit CRC register is the high and low bytes are exchanged;
8. Finally, get the contents of the CRC register is: CRC code.

◎ CRC code calculation code in C language

Description: This program calculates the length of * ptr within the first len bytes of the CRC.

```
unsigned short crc16(unsigned char *ptr, unsigned char len)
{
    unsigned short crc=0xFFFF;
    unsigned char i;
    while(len--)
    {
        crc ^=*ptr++;
        for(i=0;i<8;i++)
        {
            if(crc & 0x01)
            {
                crc>>=1;
                crc^=0xA001;
            }else
            {
                crc>>=1;
            }
        }
    }
    return crc;
}
```

7.2.4 I²C Communication Timing

AM2312 sensor I²C communication, although communication is based on standard I²C timing, but necessary according to our protocol and communication timing requirements, in order to accurately read the sensor. Please strictly in accordance with protocol design and timing for reading.

◎ **I^2C Read the complete sequence example**

Figure 13 shows a complete example of the sensor to read and write and read and write special time requirements, in strict accordance with specific time requirements to read and write, or they will not read the sensor or data appear incorrect and so on. Timing diagram of several special needs attention, detailed figure of the time requirements; host communication from start to finish, for a maximum of 3S.

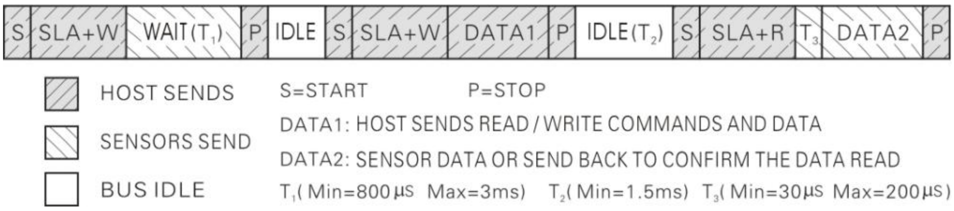


Figure 13： I^2C Sensor reading and writing a complete sample chart

◎ **I^2C Read and write timing decomposition**

Read or write the sensor, the following three steps must be, otherwise it will not communicate or can not read the correct data：

Step one： Wake−up sensor

In order to reduce the humidity sensor self−heating caused the error, the sensor in the non−working state, in a dormant state, so the sensor must wake up to read the sensor, in order to send read and write commands, otherwise the sensor will not respond. It should be noted that, in the wake sensor, sending the I2C address, the sensor will not respond to ACK, but the host must send ACK back the clock to confirm that the ninth SCL clock signal. Wake up the sensor's operating instructions as follows:

Sends a start signal is applied to host the start address, wait for a period of time (to wait for at least 800μ s, the largest 3ms; if the host is a hardware I2C, you do not need to wait, to wait for hardware I2C automatically), then sends a stop signal.

Namely: the initial signal +0 xB8 + wait (800us−3ms) + stop signal timing diagram shown in Figure 14.

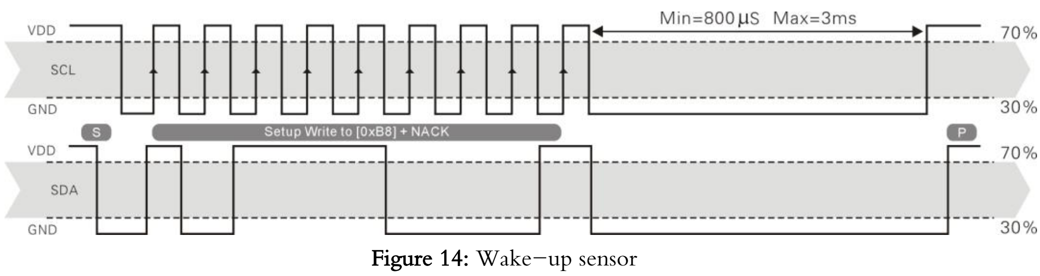


Figure 14: Wake−up sensor

Step Two read commands to send or send written instructions

Wake AM2312 sensors, can be read in full accordance with standard I^2C timing, the maximum speed supported 100Kb / s. Read temperature and humidity sample, shown in Figure 15.

Host to send commands to: START +0 xB8 (SLA) +0 x03 (function code) +0 x00 (start address) +0 x04 (register length) + STOP

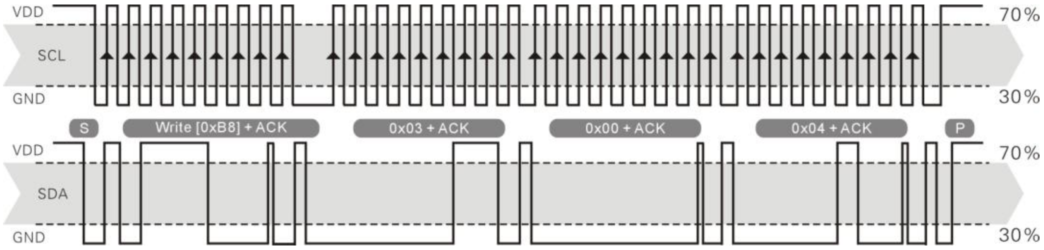


Figure 15： Sends a read command example temperature and humidity

Step three data read back or confirmation signal

Send read / write command, the host must wait at least 1.5ms, and then send a read sequence, to read back the data example shown in Figure 16; to note is that you read the data, finished his I2C address is required wait at least 30μ s before sending over the next serial clock, the read data, or communication error will occur.

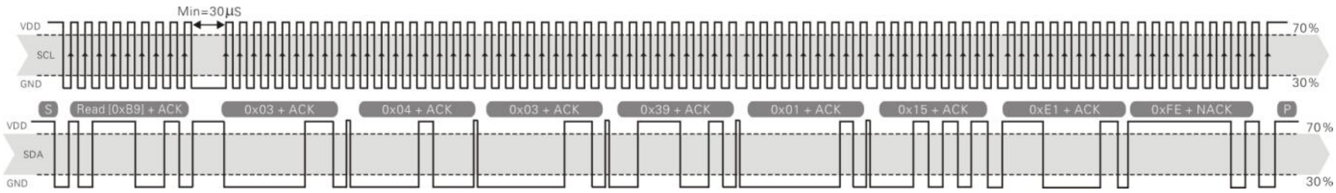


Figure 16: Example values of temperature and humidity reading

Data is read back to the host：

0x03(Function code)+0x04(Data length)+0x03(High humidity)+0x39(Low humidity)+

0x01(High temperature)+0x15(Low temperature)+0xE1(CRC Low byte checksum)+ 0xFE(CRC Checksum high byte)；

So： 0339H = 3×256+3×16+9 = 825 => Humidity = 825÷10=82.5%RH;

0115H = 1×256+1×16+5 = 277 => Temperature = 277÷10=27.7℃

Through the above three steps to complete all the registers of the sensor reading and writing operations (users can write registers, only five, namely, the status register, four user registers at the same time, the status register, can be written separately, otherwise an error); users in the design, follow these three steps must be fully read and write.

Sensors send the data, trigger a temperature and humidity measurement; measurement is completed, record temperature and humidity values, then a communication completed, the sensor automatically into hibernation; so long as the sensor does not read, read the second consecutive sensors, back to the second reading of the temperature and humidity for the latest value (minimum interval of continuous reading 2S).

7.2.5 Peripherals read the flow chart

AM2312 schematic diagram I2C sensor reading is shown in Figure 17, while our company also provides sample code to read C51, customers need to download, please visit our website (www.aosong.com) associated download, this manual does not provide code instructions.

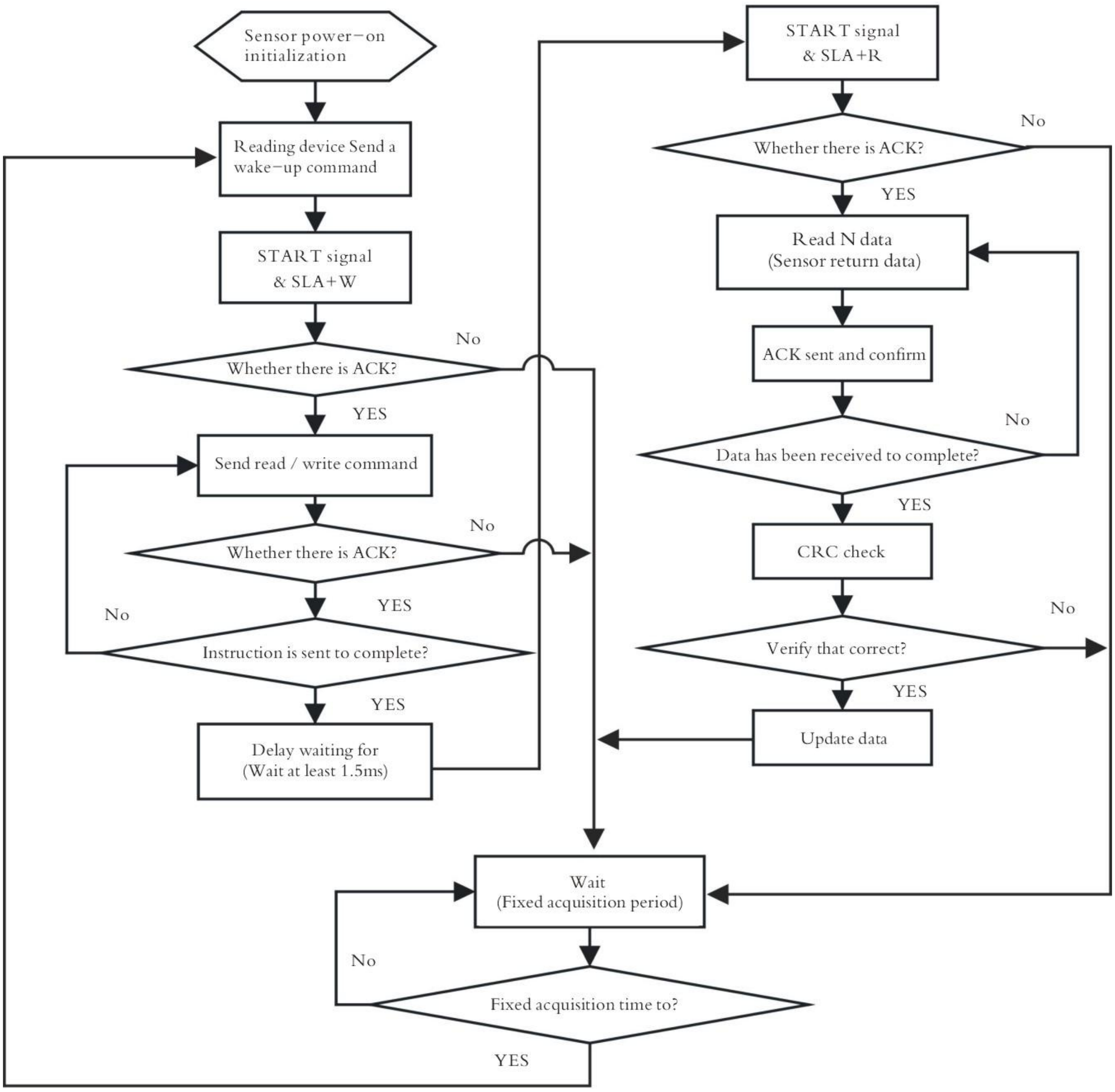


Figure 17: Sensor with I2C read flow chart

Table 1: I²C_MODBUS communication protocol summary table

Read the bus Description: I2C address is 0xB8; access a maximum of 10 registers;

Read the bus communication time for a maximum of a 3 S。

Each returned sensor data plus the CRC, the user can choose not to read the CRC。

Sensor reading frame format: Host frame format： (SLA+W)+ Function code（0x03）+ Start Add + Register number

From the machine frame format： Function code（0x03）+ Data length + Return data +CRC

Write frame format sensor: Host frame format： (SLA+W)+ Function code（0x10）+ Start Add + Register number + Save the data +CRC

From the machine frame format： Function code（0x03）+ Start Add + Register number +CRC

AM2312 sensor registers list:

Register information	Add	Register information	Add	Register information	Add	Register information	Add
High RH	0x00	Model high	0x08	Users register 1 high	0x10	Retention	0x18
Low RH	0x01	Model low	0x09	Users register 1 low	0x11	Retention	0x19
High temp.	0x02	Version number	0x0A	Users register 2 high	0x12	Retention	0x1A
Low temp.	0x03	ID(24−31) Bit	0x0B	Users register 2 low	0x13	Retention	0x1B
Retention	0x04	ID(16−23) Bit	0x0C	Retention	0x14	Retention	0x1C
Retention	0x05	ID(8 − 15) Bit	0x0D	Retention	0x15	Retention	0x1D
Retention	0x06	ID(0 − 7) Bit	0x0E	Retention	0x16	Retention	0x1E
Retention	0x07	Status Register	0x0F	Retention	0x17	Retention	0x1F

Status Register Definition: Bit7−Bit0 bit reserved；

Temperature format: temperature highest (Bit15) equal to 1 indicates a negative temperature, the temperature highest (Bit15) is equal to 0 for positive temperature; temperature in addition to the highest bit (Bit14 ~ Bit0) that string out of the temperature sensor. String out of the temperature sensor is 10 times the actual humidity value;

Write sensor： Register is available for users to write (0x0F ~ 0x13); others prohibit write register and status register can only be written separately。

Reading and writing sample：

Function	code	Start Add	Frame data content
Read temperature and humidity	0x03	0x00	Send：(SLA+W)+0x03+0x00+0x04
			Return：0x03+0x04+ High RH + Low RH + High temp. + Low temp. +CRC
Read the temp.	0x03	0x02	Send：(SLA+W)+0x03+0x02+0x02
			Return：0x03+0x02+ High temp. + Low temp. +CRC
Humidity reading	0x03	0x00	Send：(SLA+W)+0x03+0x00+0x02
			Return：0x03+0x02+ Low RH+ Low RH+ CRC
Read device information	0x03	0x08	Send：(SLA+W)+0x03+0x08+0x07
			Return：0x03+0x07+ Model (16)+ Version (8)+ ID(32)+CRC
Write Status Register	0x10	0x0F	Send：(SLA+W)+0x10+0x0F+0x01+0x01+0xF4（Low）+0xB7（High） Note: Function code + register start address register number + Save + content + CRC
			Return：0x10+0x0F+0x01+0xB4（Low byte）+0x35（High byte） Note：Function code + register start address + register number + CRC
Write user registers 1	0x10	0x10	Send：(SLA+W)+0x10+0x10+0x02+0x01+0x02+0xC0+0x92
			Return：0x10+0x10+0x02+0xFC+0x04

Note: SLA = I2C address 0xB8. Table for the parity bit CRC, CRC for the 16-bit, low byte first, high byte in the post.
Return error code: 0x80: does not support the function code 0x81: reading an illegal address 0x82: write data beyond the scope of 0x83: CRC checksum error 0x84: disable writes.

8、the license agreement

Without the copyright holder's prior written permission in any form or by any means, electronic or mechanical (including photocopying), any part of this manual may be reproduced, nor may its contents be communicated to third parties. The contents are subject to change without notice.

The company has a software and third-party ownership, the user only signed a contract or license the software before use.

9、warnings, and personal injury

Do not use this product as safety or emergency stop devices, as well as the product failure could cause personal injury to any other application, unless there are special purpose or authorized use. In the installation, handling, use or maintenance of the product, please consult the product data sheets and application notes. Failure to comply with this proposal, likely to cause death and serious injury. The Company will not assume the resulting personal injury or death of any compensation, and thus exempt from the company's managers and employees and affiliated agents, distributors, etc. that may arise from any claims, including: a variety of costs, compensation costs, legal fees and so on.

10、Quality Assurance

The company's direct purchase of their products to provide a period of 12 months quality assurance (from date of shipment). The company published the product data sheet specifications shall prevail. If the warranty period, the quality of the product proved defective indeed, the company will provide free repair or replacement. User must meet the following conditions:

① This product is found defective within 14 days of written notice to the Company;
② The product shall be paid by mail back to the company;
③ The product should be in the warranty period.

The Company applied only to those who meet the technical requirements of the occasion of the product resulting from defective products responsible. Those companies for their products are used in special applications without any guarantee, warranty or written statement. The company or its products into the product reliability of the circuit does not make any promises.