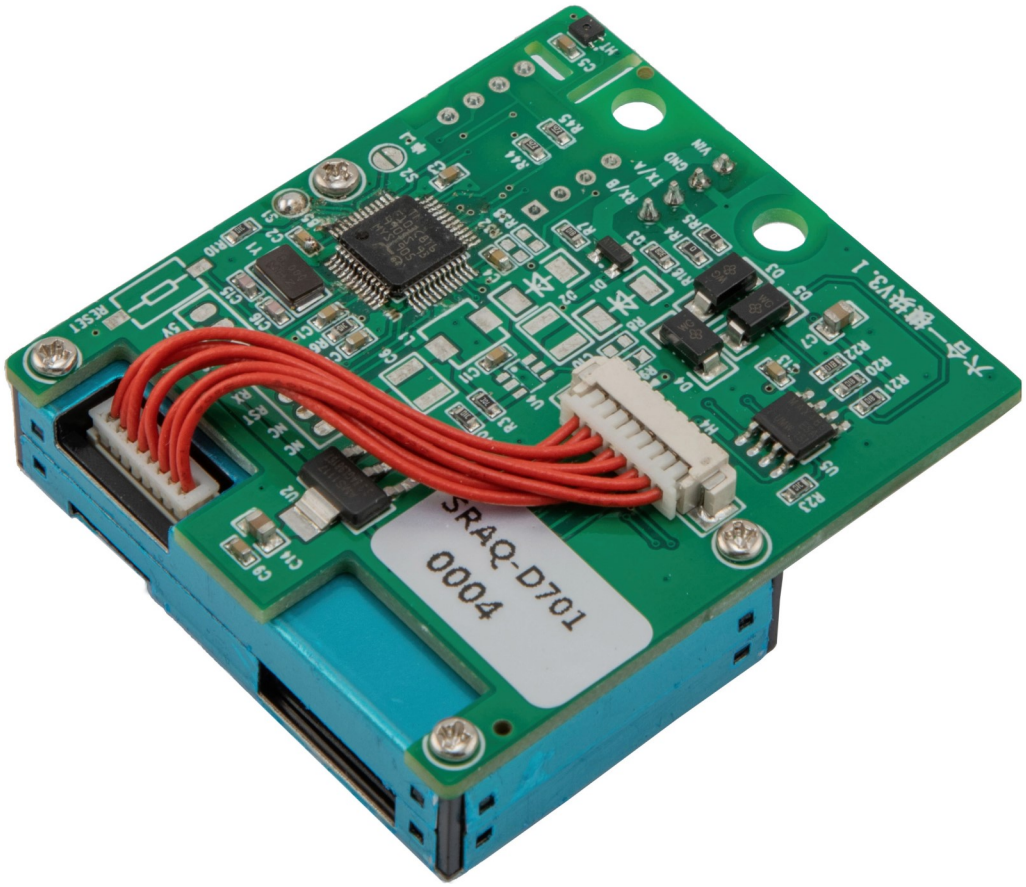
MULTIFUNCTION AIR QUALITY SENSOR MODULE (MODBUS)

User Guide for SRAQ-D701



Product Description

The SRAQ-D701 is a comprehensive air quality sensor module which combines CO_2, laser dust, temperature and humidity, TVOC and formaldehyde. Developed for integration into air purifiers, air quality monitors and custom enclosures. The SRAQ-D701 provides real time concentration values via Modbus-RTU output. It offer good stability, and is well suited for applications where TVOC and particulate matter concentrations are a concern.

Features

- Detects multiple pollutants CO_2, dust etc.
- Modbus output
- High reliability and good stability
- High accuracy & fast response
- Compact design for easy integration

Applications

- Office and commercial buildings
- Factory floors
- Laboratories
- HAVC industry
- General air quality monitoring
- Smart home systems

Thank you for choosing L-com product. To ensure safe, accurate performance and product longevity, please take a moment to familiarize yourself with this manual before powering the device. Please keep it handy for future reference. In case of any questions regarding the installation or use of product, please call us at 800.341.5266.

Reach out to us at customerservice@l-com.com and visit our website at www.l-com.com

M_L-com_SRAQ-D701-11/2022_R1.0

Technical Parameters

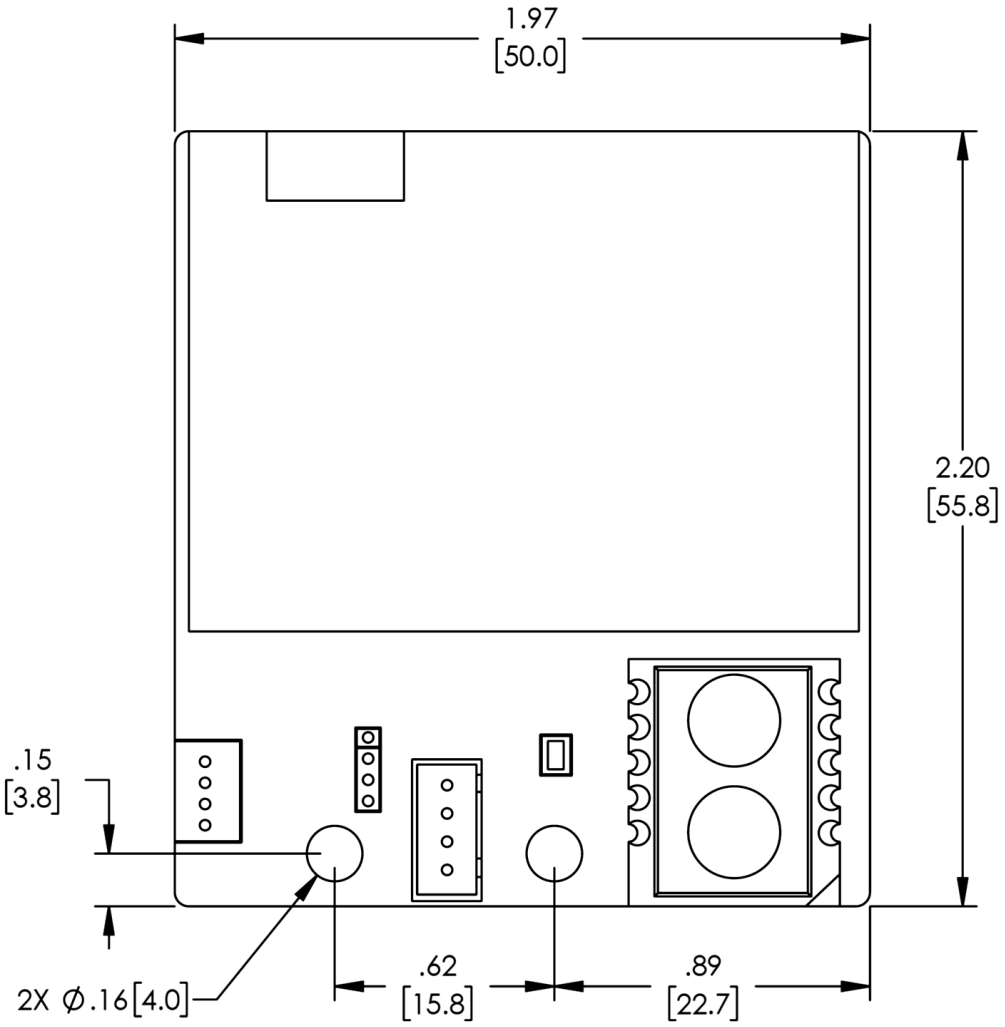
Detection Parameters and Resolution: The CO_2 value and formaldehyde value measured by the module are equivalent value of TVOC, please pay attention when using or purchasing.

Pollutant	Measurement resolution	Measuring range	Accuracy
eCO_2	1 ppm	400~5000 ppm	± 100 ppm
PM2.5	0.3 ug/m^3	0~1000 ug/m^3	± 10%
PM10	0.3 ug/m^3	0~1000 ug/m^3	± 10%
Temperature	0.01 °C	20~80 °C	± 0.5 °C
Humidity	0.04%	5~100% RH	± 3% RH
TVOC	1 ug/m^3	1230 ug/m^3	± 90 ug/m^3
CH_2O	1 ug/m^3	500 ug/m^3	± 40 ug/m^3

1. Working voltage: 5 ± 0.2 VDC
2. Working environment: Working temperature 0~50 °C, Working humidity: 15% RH – 90% RH (no condensation)
3. Working environment: Temperature range -20~60 °C, Humidity: 5% RH – 95% RH (no condensation)
4. Communication method: Modbus RTU – RS485
5. Preheating time: 10-15 minutes, the longer the power-on time, the more stable the data will be.

Mounting Diagrams

Mounting type: Board mount. Hole diameter 2 mm, pitch 11.5 mm.



Wire Definition

Port connector type is XH2.54mm pitch 4-pin terminal connector, 90° vertical mounted to PCB.

Sr. No.	Function	Notes	Color
1	VCC	12 – 24 VDC, power supply positive	Brown
2	GND	GND, power supply negative	Black
3	RS485_A+	RS485_A+	Green
4	RS485_B-	RS485_B-	Blue

Communication Protocol

The module hardware supports serial TTL communication interface and RS485 communication interface. When testing, users need to choose the corresponding communication method development according to the module interface type they have purchased.

RS485 Interface Communication Interface Related Protocol:

Serial port parameters: (9600/8/N/1) ModBus-RTU 485

The factory default address of the module is 0x01

Baud rate: 9600, Check bit: no, Stop position: 1 bit

Return data time: <300ms

The reading speed cannot exceed 350ms

Read Address Command

Receive	Fixed	Function Code	Fixed	Fixed	Fixed	Fixed	CRC_L	CRC_H
	FE	17	00	00	00	01	CRC Check	
Answer	Fixed	Function Code	Number of Bytes	Firmware Version	Current Address	CRC_L	CRC_H	-
	FE	17	02	xx	yy	CRC Check		-

For example: Send command: FE 17 00 00 00 01 A0 06 Return data: FE 17 02 11 01 64 30

Indicates that the device address is 01 and the version number is V1.1

Modify Address Command

The factory default address of the module is 0x01, and the address range can be changed to: 0~254.

Receive	Current Address	Function Code	Fixed	Fixed	Reserved	Default Address	CRC_L	CRC_H
	yy	06	00	00	00	zz	CRC check	
Answer	Address Before Modification	Function Code	Number of Bytes	Keep	Modified Address	CRC_L	CRC_H	-
	yy	06	02	00	Zz	CRC check		-

For example: When the current address is 01, the preset address is: 02

Send command: 01 06 00 00 00 02 08 0B Return data: 01 06 02 00 02 39 49

Read Data Command

Receive	Current Address	Function Code	Register Start Address		Number of Sensors Required to Read		CRC_L	CRC_H
	yy	03	00	MM	00	NN	CRC check	
Answer	Current Address	Function Code	Data Length	Sensor Data	CRC_L	CRC_H	-	
	yy	03	NN*2	xx xx ……xx xx	CRC Check		-	

Note: The returned xx xx ……xx xx sensor data can be changed according to the register address and data length.

00 MM represents the register address of the sensor, and 00 NN is the data length.

Register address（00 MM）						
00 00	00 01	00 02	00 03	00 04	00 05	00 06
eCO2	TVOC	CH2O	PM2.5	Humidity	Temperature	PM10

Description

1. The minimum number of sensors 00 NN is 00 01, and the maximum is 00 07. When the MM value is 00, the maximum NN value can be 07. At this time, the value of all sensors can be read, and it can also be 01. When it is 01, only CO_2 data can be read alone. And so on.
2. The address at the front of the register can read the data of the following sensor when the data length is increased, but the address at the back of the register cannot read the data of the sensor before this address. See details below:

00 MM 00 NN	Environmental data that can be read
00 00 00 01	Indicates that CO_2 data is read from the starting address 00 00
00 00 00 02	Means to read CO_2, TVOC data from the starting address 00 00
00 00 00 03	Means to read CO_2, TVOC, CH_2O data from the starting address 00 00
00 00 00 04	Means to read CO_2, TVOC, CH_2O, PM2.5 data from the starting address 00 00
00 00 00 05	Means to read CO_2, TVOC, CH_2O, PM2.5, H data from the starting address 00 00
00 00 00 06	Means to read CO_2, TVOC, CH_2O, PM2.5, H, T data from the starting address 00 00
00 00 00 07	Means to read CO_2, TVOC, CH_2O, PM2.5, H, T, PM10 data from the starting address 00 00
00 01 00 01	Indicates that the TVOC data is read from the starting address 00 01
00 01 00 02	Indicates that TVOC and CH_2O data are read from the starting address 00 01
00 01 00 03	Means to read TVOC, CH_2O, PM2.5 data from the starting address 00 01
00 01 00 04	Means to read TVOC, CH_2O, PM2.5, H data from the starting address 00 01
00 01 00 05	Means to read TVOC, CH_2O, PM2.5, H, T data from the starting address 00 01
00 01 00 06	Means to read TVOC, CH_2O, PM2.5, H, T, PM10 data from the starting address 00 01
00 02 00 01	Indicates that CH_2O data is read from the starting address 00 02
00 02 00 02	Indicates that CH_2O and PM2.5 data are read from the starting address 00 02
00 02 00 03	Means to read CH_2O, PM2.5, H data from the starting address 00 02
00 02 00 04	Means to read CH_2O, PM2.5, H, T data from the starting address 00 02
00 02 00 05	Means to read CH_2O, PM2.5, H, T, PM10 data from the starting address 00 02
00 03 00 01	Indicates that PM2.5 data is read from the starting address 00 03
00 03 00 02	Means to read PM2.5, H data from the starting address 00 03
00 03 00 03	Means to read PM2.5, H, T data from the starting address 00 03
00 03 00 04	Means to read PM2.5, H, T, PM10 data from the starting address 00 03
00 04 00 01	Indicates that the H data is read from the starting address 00 04
00 04 00 02	Means to read H, T data from the starting address 00 04
00 04 00 03	Means to read H, T, PM10 data from the starting address 00 04
00 05 00 01	Indicates that the T data is read from the starting address 00 05
00 05 00 02	Means to read T, PM10 data from the starting address 00 05
00 06 00 01	Indicates that PM10 data is read from the start address 00 06

How to Read a Frame of Data

For example, if the address of the module is 0x01, the user needs to send 01 03 00 00 00 07 04 08 to read out seven kinds of sensor data.
The returned data format is as follows:

Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]	Byte[5]	Byte[6]	Byte[7]	Byte[8]	-
Header	Function Code	Data Length	CO_2 Data		TVOC Data		CH_2O Data		-
0x01	0x03	0x0E	CO_2_H	CO_2_L	TVOC_H	TVOC_L	CH_2O_H	CH_2O_L	-
Byte[9]	Byte[10]	Byte[11]	Byte[12]	Byte[13]	Byte[14]	Byte[15]	Byte[16]	Byte[17]	Byte[18]
PM2.5 Data		Humidity Data		Temperature Data		PM10 Data		CRC16 Check	
PM2.5_H	PM2.5_L	Hum_H	Hum_L	Tem_H	Tem_L	PM10_H	PM10_L	CRC16_L	CRC16_H

Calculation Method of Each Sensor Data

CO_2 (ppm) = CO_2_H * 256 + CO_2_L

TVOC(ug/m3) = (TVOC_H * 256 + TVOC_L) / 10

CH_2O(ug/m3) = (CH_2O_H * 256 + CH_2O_L) / 10

PM2.5 (ug/m3) = PM2.5_H * 256 + PM2.5_L

PM10 (ug/m3) = PM10_H * 256 + PM10_L

Positive temperature (°C) = (Temperature_H * 256 + Temperature_L) / 10

For example: T = (0 x 0110) / 10 = 272 / 10 = 27.2 ℃

Negative Temperature (°C)

Method 1: Directly convert to a signed int type, for example: T = 0 x ffbf / 10 = -65 / 10 = -6.5 ℃

Method 2: If greater than 0 x 7fff (32767)

Then it is a negative number T = (0 x ffbf - 65536) / 10 = (65471 - 65536) / 10 = -65 / 10 = -6.5 ℃

Humidity (%RH) = (Humidity_H * 256 + Humidity_L) / 10

For example: T = (0 x 02AD) / 10 = 685 / 10 = 68.5 %RH

Note: If user need to use mg/m^3 as the unit, please convert it. The conversion formula is: 1mg/m^3 =1000 ug/m^3

The length of the CRC check byte is 17 (that is, the bytes of Byte[0]~Byte[16]), select A001 or 8005 in reverse order.

Matters Needing Attention

1. Avoid contact with organic solvents (including silica gel and other adhesives), paints, pharmaceuticals, oils and high-concentration gases.
2. The module must not be subjected to excessive shock or vibration.
3. Do not apply this module to systems involving personal safety.
4. Do not install the module in a strong air convection environment.

CRC Check Calculation Method

Function function: CRC check function, generate CRC

Parameter description: arr_buff: array set to be verified

len: The length of the data to be verified

Return parameter: CRC is an unsigned int type, the high byte is the high byte in front, and the low byte is in the back

```
unsigned int CRC_Compute ( unsigned char *arr_buff, unsigned char len)
{
unsigned int crc=0xFFFF;
        unsigned char i, j;
        for ( j=0; j <len;j++)
        {
                crc=crc ^*arr_buff++;
                for ( i=0; i<8; i++)
                {
                        if( ( crc&0x0001) >0)
                        {
                                crc=crc>>1;
                                crc=crc^ 0xa001;
                        }
                        else
                                crc=crc>>1;
                }
        }
        return ( crc);
}
```