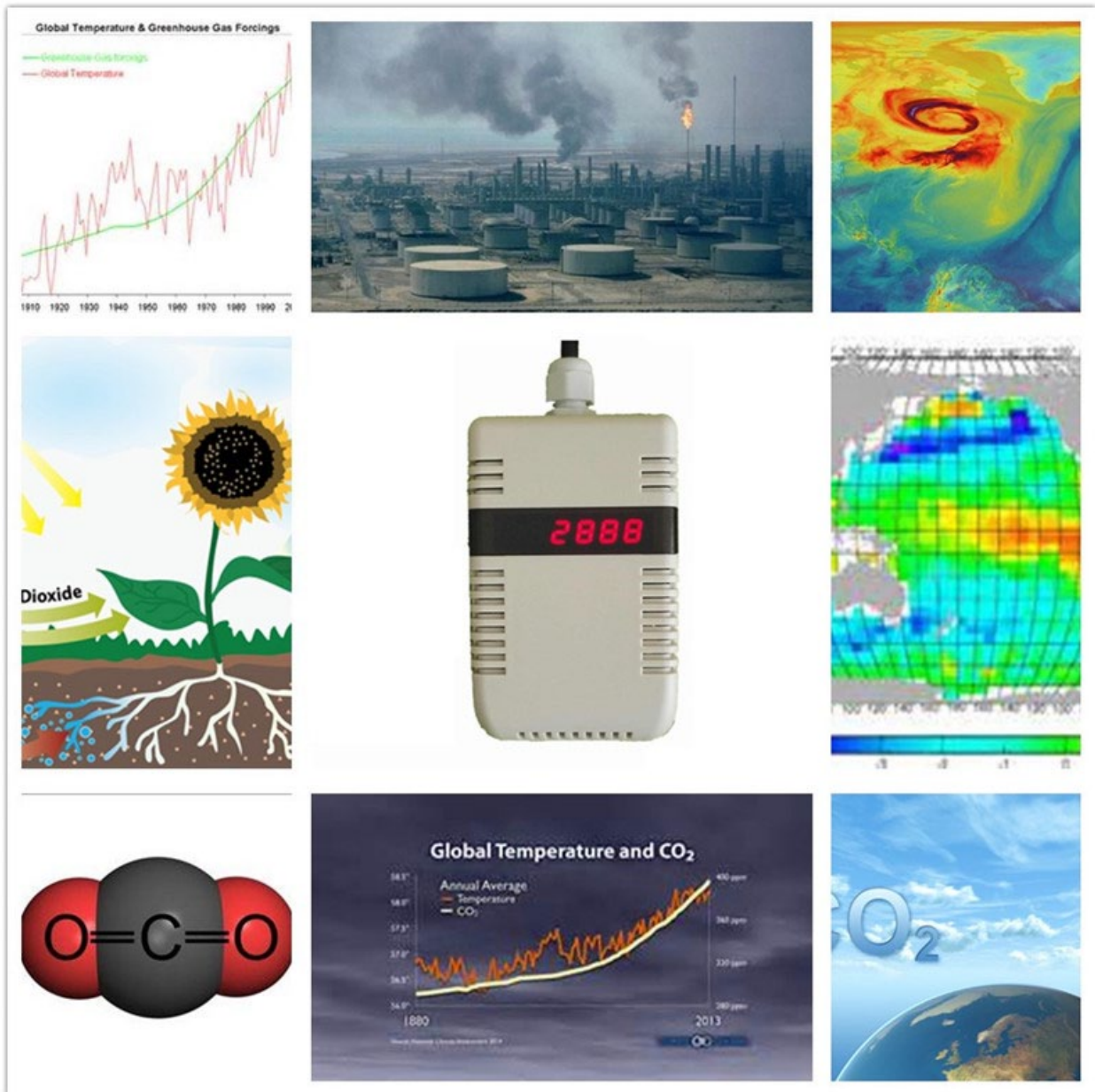


# CD10

## CO2 Carbon Dioxide Sensor Transmitter

### User Manual



# Index

1	Customer Support.....	3
2	Introduction .....	4
3	Wiring diagrams .....	5
4	Dimension and Ordering Infomation .....	7
	<b>4.1 Dimension</b> .....	<b>7</b>
	<b>4.2 Ordering Infomation</b> .....	<b>7</b>
5	Safty ,Care and Installation .....	9
	<b>5.1 Installation</b> .....	<b>9</b>
6	Output Signal Conversion .....	10
7	RS485 Modbus Protocol .....	11
	<b>7.1 Modbus Protocol</b> .....	<b>11</b>
	<b>7.2 Modbus Register</b> .....	<b>11</b>
	<b>7.3 Modbus Register Detail Descripton</b> .....	<b>12</b>
	<b>7.4 Modbus Function Code</b> .....	<b>14</b>
	7.4.1 Function Code 3 Protocol Example .....	14
	7.4.2 Function Code 4 Protocol Example .....	16
	7.4.3 Function Code 6 Protocol Example .....	17
	7.4.4 Function Code 16 Protocol Example .....	18
8	Software Configuration Utility.....	19
	<b>8.1 Hardwar Setup</b> .....	<b>19</b>
	<b>8.2 Universal Modbus Comm Utility</b> .....	<b>19</b>
	<b>8.3 SensorOneSet Configuration Utility</b> .....	<b>20</b>
	Appendix .....	21
	<b>Copyright and Trademark</b> .....	<b>21</b>
	<b>Version Control</b> .....	<b>21</b>

# 1 Customer Support

Thank you very much for your order. Our success comes from the continuous faith in the excellence of our products and services, something we are committed to and would never sacrifice. Our customer service, especially in the after sales phase, guarantees the satisfaction of our clients. In line with this strategy, we appreciate that you can share with us your feedback at any time for our improvement, be it positive or negative, so if we can serve you better in anyway, please do inform us.

## **Website**

<http://www.infwin.com>

## **E-Mail**

[infwin@163.com](mailto:infwin@163.com)

## **Telephone**

+86-411-66831953, +86-4000-511-521

## **Fax**

+86-411-82388125

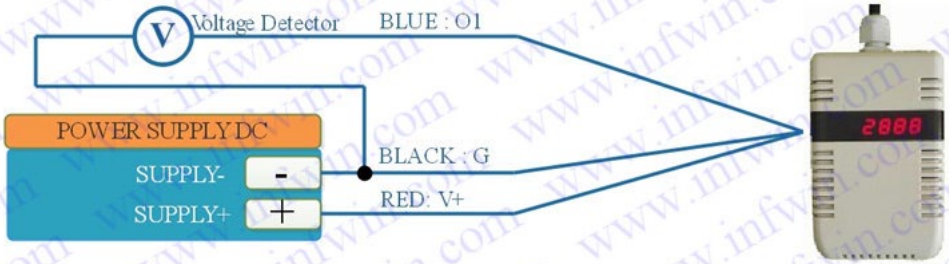
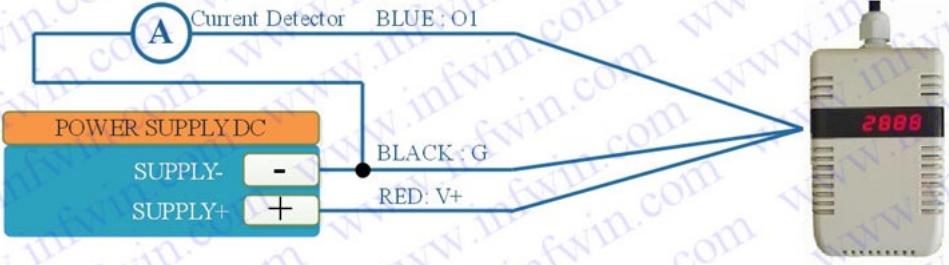
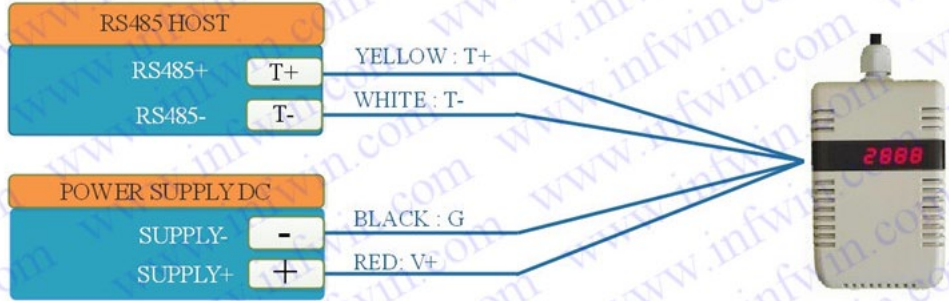
## 2 Introduction

CD10 co2 transmitter integrated with dual channel non-dispersive infrared (NDIR) CO2 sensor. Dual channels consist of one CO2 channel that measures gas concentration and one reference channel that measures the sensor signal intensity. The sensor is applicable for science research, air conditioning, greenhouse, smart agriculture etc.

- Integrated with dual channel non-dispersive infrared (NDIR) CO2 sensor
- 10-years long life sensing with temperature calibration
- Output Interface with RS485, Voltage, Current
- Immunity to background gas e.g. CH4、H2S、SO2、N2、O2 etc.
- High accuracy with excellent stability
- Reverse power protection and Built-in TVS/ESD protection

<b>Specifications</b>			
<b>Output Interface</b>	Analog Voltage 0-2V (Output resistance ~0ohm)	Analog Current 4-20mA (Load Resistor<500ohm)	RS485 Modbus-RTU
<b>Power Supply</b>	12-30V/DC	12-30V/DC	12-30V/DC
<b>Power Consumption</b>	60mA@24V DC	80mA@24V DC (with 20mA output signal)	60mA@24V DC
<b>CO2 Measurement</b>	Range 0-1000ppm, Accuracy $\pm(50\text{ppm}+3\% \text{ Reading value})$ , Resolution 1ppm Range 0-2000ppm, Accuracy $\pm(50\text{ppm}+3\% \text{ Reading value})$ , Resolution 1ppm Range 0-5000ppm, Accuracy $\pm(50\text{ppm}+3\% \text{ Reading value})$ , Resolution 1ppm Range 0-10000ppm, Accuracy $\pm(10\% \text{ Reading value})$ , Resolution 1ppm Range 0-30000ppm, Accuracy $\pm(10\% \text{ Reading value})$ , Resolution 1ppm Range 0-50000ppm, Accuracy $\pm(10\% \text{ Reading value})$ , Resolution 1ppm		
<b>Temperature Measurement</b>	Range: -40~80°C, Resolution:0.1°C, Accuracy:±0.5°C		
<b>Sensing method</b>	Dual channel non-dispersive infrared (NDIR) CO2 sensor		
<b>Warmup time</b>	Max. 180 seconds		
<b>IP Ratings</b>	IP54		
<b>Operating Temperature</b>	Temperature:-20-60 °C Humidity:0-95% RH (No condensation)		
<b>Installation</b>	Wall mounting		
<b>Cable Length</b>	2 meters, or Customize		
<b>Dimension</b>	130*70*42mm		

### 3 Wiring diagrams

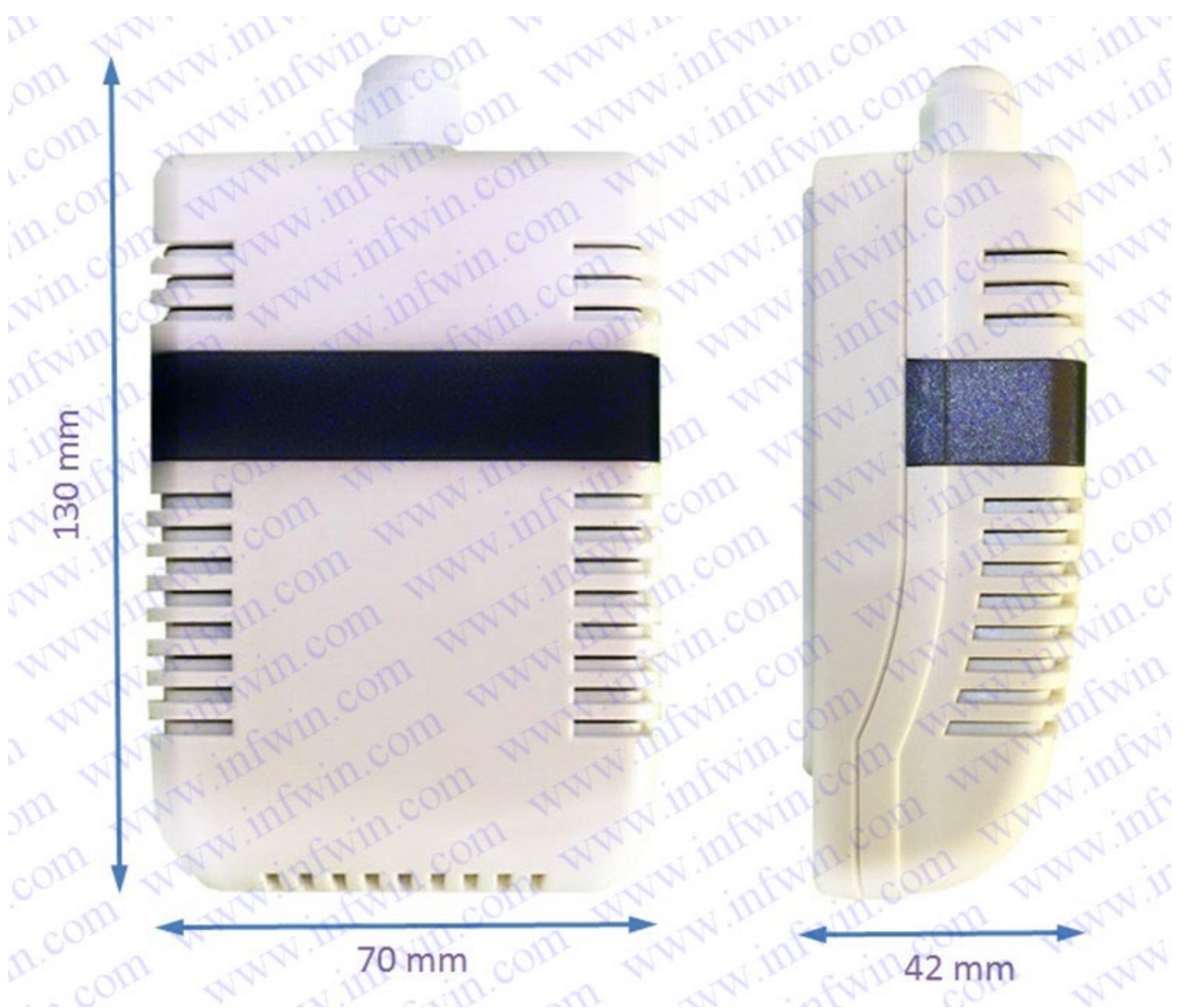
Type	Wiring diagram
<b>Analog Voltage Output</b>	<p>Red (V+): Power Supply +                      Black (G): Power Supply -                      Blue (O1): Analog Output</p> <p style="text-align: center;"><b>Wiring Diagram for Analog Voltage Output 0-2V</b></p> 
<b>Analog Current Output</b>	<p>Red (V+): Power Supply +                      Black (G): Power Supply -                      Blue (O1): Analog Output</p> <p style="text-align: center;"><b>Wiring Diagram for Analog Current Output 4-20mA</b></p> 
<b>RS485 Modbus</b>	<p>Red (V+): Power Supply +                      Black (G): Power Supply -                      Yellow (T+): RS485+/A/T+                      White (T-): RS485-/B/T-</p> <p style="text-align: center;"><b>Wiring Diagram for RS485 Modbus</b></p> 

ALL RS485 communication parameters (Mosbus Slave Address, baudrate, parity, databits, stopbits) are set in internal register and can be saved when power down, the factory setting is ADDRESS=1, BAUDRATE=9600bps, PARITY=NONE, DATABITS=8bits, STOPBITS=1bit;

Sometimes you may FORGET the communication settings, In this case, you can open the shield module and set the jumper to SET position and re-power up the transmitter, then the sensor start-up with a fixed communication settings (we call it setting mode) ADDRESS=0, BAUDRATE=9600bps, PARITY=NONE, DATABITS=8bits, STOPBITS=1bit; Communicate with the sensor using this parameters and then set your desired communication parameters, then set the jumper to run position, then re-power up the sensor, and the sensor will communicate with your settings.

## 4 Dimension and Ordering Information

### 4.1 Dimension



### 4.2 Ordering Information

Parameters	Code	Comments
Code 1: Product Series	CD10	CD10 CO2 carbon dioxide sensor transmitter
Code 2: Measuring Parameters	A	CO2
Code 3: CO2 Range	A B C	0-1000ppm 0-2000ppm 0-5000ppm

	D	0-10000ppm
	E	0-30000ppm
	F	0-50000ppm
Code 4:Display	A	With led display
	B	No led display
Code 5: Power Supply	A	12-30V DC
Code 6: Output Interface	A	Analog Voltage 0-2V
	B	Analog Current 4-20mA
	C	RS485,Modbus-RTU
	D	RS485,Modbus-RTU & Analog Voltage 0-2V
	E	RS485,Modbus-RTU & Analog Current 4-20mA
	F	Customize
Code 7: Cable Length	002	2 meters
	XXX	Customize, XXX is required cable length(Unit: meter)

Ordering Code Example:

CD10 CO2 carbon dioxide sensor transmitter, Range 0-5000ppm, With led display, Power Supply 12-30V DC,Output Interface RS485 Modbus-RTU, Cable Length 5 meters. Ordering Code is : CD10 - A C A A C 005



## 5 Safty ,Care and Installation

### 5.1 Installation



## 6 Output Signal Conversion

Output Interface	Parameters Range	Conversion Formula
<b>Analog Voltage Output 0-2V</b>	RANGE:0-1000ppm RANGE:0-2000ppm RANGE:0-5000ppm RANGE:0-10000ppm RANGE:0-30000ppm RANGE:0-50000ppm	$CO_2 = RANGE * VOLTAGE / 2.00$ . When RANGE=2000 ppm and VOLTAGE=1.0V, then $CO_2 = 2000 * 1.00 / 2.00 = 1000$ ppm
<b>Analog Current Output 4-20mA</b>	RANGE:0-1000ppm RANGE:0-2000ppm RANGE:0-5000ppm RANGE:0-10000ppm RANGE:0-30000ppm RANGE:0-50000ppm	$CO_2 = RANGE * (CURRENT - 4) / 16$ . When RANGE=2000 ppm and CURRENT=12mA, then $CO_2 = 2000 * (12 - 4) / 16 = 1000$ ppm
<b>RS485 Modbus-RTU</b>	RANGE:All	$CO_2 = (REGISTER VALUE)$ . When REGISTER VALUE=1000, then $CO_2 = 1000$ ppm.
<b>Customize</b>	Contact support for customized sensor interface	

NOTE: The unit of VOLTAGE is (V), The unit of CURRENT is (mA).

NOTE: CO2 is CO2 ppm value.

# 7 RS485 Modbus Protocol

## 7.1 Modbus Protocol

Modbus Protocol is widely used to establish master-slave communication between intelligent devices or sensors. A MODBUS message sent from a master to a slave contains the address of the slave, the function code (e.g. 'read register' or 'write register'), the data, and a check sum (LRC or CRC).

The sensor is RS485 interface with Modbus protocol. The default serial communication settings is slave address 1, modbus rtu, 9600bps, 8 databits and 1 stop bit. All communication settings can be changed with modbus command, and take effective after re-power up the sensor.

Following modbus function code are supported by sensor.

Modbus Function Code 0x03 : used for reading holding register.

Modbus Function Code 0x04 : used for reading input register.

Modbus Function Code 0x06 : used for writing single holding register.

Modbus Function Code 0x10: used for writing multiple holding register.

## 7.2 Modbus Register

Parameters	Register Addr. (HEX/DEC)	Data Type	Modbus Function Code(DEC)	Range and Comments	Default Value
CO2 ppm value	0x0000 /0	UINT16 RO	3/4	0-50000 for 0-50000ppm	N/A
RESERVED	0x0001 /1	UINT16 RO	3/4	0	0
RESERVED	0x0002 /2	UINT16 RO	3/4	0	0
RESERVED	0x0003 /3	UINT16 RO	3/4	0	0
RESERVED	0x0004 /4	UINT16 RO	3/4	0	0
RESERVED	0x0004 /5	UINT16 RO	3/4	0	0

SLAVEADDRESS	0x0200 /512	UINT16 R/W	3/6/16	0-255	1
BAUDRATE	0x0201 /513	UINT16 R/W	3/6/16	0-6 0:1200bps 1:2400bps 2:4800bps 3:9600bps 4:19200bps 5:38400bps	3:9600bps
PROTOCOL	0x0202 /514	UINT16 R/W	3/6/16	0-1 0:Modbus RTU 1:Modbus ASCii	0:Modbus RTU
PARITY	0x0203 /515	UINT16 R/W	3/6/16	0-2 0:None 1:Even 2:Odd	0:None Parity
DATABITS	0x0204 /516	UINT16 R/W	3/6/16	1 1:8 databits	1:8 databits
STOPBITS	0x0205 /517	UINT16 R/W	3/6/16	0-1 0:1 stopbit 1:2 stopbits	0:1 stopbit
RESPONSEDELAY	0x0206 /518	UINT16 R/W	3/6/16	0-255 for 0-2550 milliseconds	0
ACTIVEOUTPUTIN TERVAL	0x0207 /519	UINT16 R/W	3/6/16	0-255 for 0-255 seconds.	0

NOTE: UINT16:16 bit unsigned integer, INT16:16bit signed integer

NOTE: R: Register is Readonly, R/W: Register is Read/Write

NOTE: HEX is Hexadecimal (data with 0x/0X prefix), DEC is Decimal

### 7.3 Modbus Register Detail Descripton

<b>CO2</b>		
Data Range	0-50000 For 0-50000ppm	Default: N/A
Power Down Save	N/A	

Note:Temperature value (Binary complement).

Example:When REGISTER = 0x0702 (HEX format), then VALUE=(0x07\*256+0x02) =1794ppm

<b>SLAVEADDRESS --- Modbus Slave Address</b>		
Data Range	0-255	Default: 1
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

<b>BAUDRATE --- Serial Comm Baudrate</b>		
Data Range	<b>0-5</b> <b>0:</b> 1200bps <b>1:</b> 2400bps <b>2:</b> 4800bps <b>3:</b> 9600bps <b>4:</b> 19200bps <b>5:</b> 38400bps	Default: 3
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

<b>PROTOCOL --- Serial Comm Protocol</b>		
Data Range	0 0:Modbus RTU	Default: 0
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

<b>PARITY --- Serial Comm Parity</b>		
Data Range	0-2 0:NONE 1:EVEN 2:ODD	Default: 0
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

<b>DATABITS --- Serial Comm Databits</b>		
Data Range	1 1:8 databits	Default: 1
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

<b>STOPBITS --- Serial Comm Stopbits</b>		
Data Range	0-1	Default: 0

	0:1 stopbit 1:2 stopbits	
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

<b>RESPONSEDELAY --- Serial Comm Response Delay</b>		
Data Range	0-255 for 0-2550 milliseconds, 0 for disabled	Default: 0
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

Note: Sensor will delay a period before response to master request command.

Example: When set to 5 and receive a request from master device, then sensor will delay 5\*10ms=50ms, then response to master.

<b>ACTIVEOUTPUTINTERVAL --- Serial Comm Active Output Interval time</b>		
Data Range	0-255 for 0-255 seconds, 0 for disabled	Default: 0
Power Down Save	YES	

Note: Please re-power on the sensor to take effective after set.

Note: Sensor will output the data actively without any master request command.

Note: Only ONE sensor should be on RS485 network, or there will be data collision and corrupt the data on line.

Note: Refer to SETTING mode to exit the Active Output Mode.

Example: When set to 5 then sensor will output the data every 5 seconds without any master request command.

## 7.4 Modbus Function Code

For description below, data started with 0X/0x means that it's in HEX format.

### 7.4.1 Function Code 3 Protocol Example

**Master Request: AA 03 RRRR NNNN CCCC**

AA	1 byte	Slave Address,0-255
0x03	1 byte	Function Code 3
RRRR	2 byte	Starting Register Addr

NNNN	2 byte	Quantity of Register to read
CCCC	2 byte	CRC CHECKSUM

**Slave Response:AA 03 MM VV0 VV1 VV2 VV3... CCCC**

AA	1 byte	Slave Address,0-255
0x03	1 byte	Function Code 3
MM	1 byte	Register Data Byte Count
VV0,VV1	2 byte	Register Value (High8bits first)
VV2,VV3	2 byte	Register Value (High8bits first)
...	...	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

**Example:Read register 0x0200-0x0201,that is slave address and baudrate.**

**Master Request:01 03 0200 0002 C5B3**

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x03
Starting Register Addr.	2 byte	0x0200
Quantity of Register to read	2 byte	0x0002
Checksum	2 byte	0xC5B3

**Slave Response:01 03 04 00 01 00 03 EB F2**

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x03
Register Data Byte Count	1 byte	0x04
Register Value: Address	2 byte	0x00(HIGH 8 Bits)
		0x01(LOW8 Bits)
Register Value: Baudrate	2 byte	0x00(HIGH 8 Bits)
		0x03(LOW8 Bits)
Checksum	2 byte	0xEBF2

## 7.4.2 Function Code 4 Protocol Example

### Master Request: AA 04 RRRR NNNN CCCC

AA	1 byte	Slave Address,0-255
0x04	1 byte	Function Code 4
RRRR	2 byte	Starting Register Addr
NNNN	2 byte	Quantity of Register to read
CCCC	2 byte	CRC CHECKSUM

### Slave Response: AA 04 MM VV0 VV1 VV2 VV3... CCCC

AA	1 byte	Slave Address,0-255
0x04	1 byte	Function Code 4
MM	1 byte	Register Data Byte Count
VV0,VV1	2 byte	Register Value (High8bits first)
VV2,VV3	2 byte	Register Value (High8bits first)
...	...	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

### Example:Read register 0x0000,that is CO2 value:

#### Master Request: 01 04 0000 0001 31CA

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x04
Starting Register Addr.	2 byte	0x0000
Quantity of Register to read	2 byte	0x0001
Checksum	2 byte	0x31CA

#### Slave Response: 01 04 02 0010 B8FC

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x04
Register Data Byte Count	1 byte	0x02
Register Value:	2 byte	0x05 (High8bits first)
CO2 Value		0xA3



Checksum	2 byte	0xFA19
----------	--------	--------

PYR=0x05\*256+0xA3=1443ppm.

### 7.4.3 Function Code 6 Protocol Example

#### Master Request:AA 06 RRRR VVVV CCCC

AA	1 byte	Slave Address,0-255
0x06	1 byte	Function Code 6
RRRR	2 byte	Register Addr (High8bits first)
VVVV	2 byte	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

#### Slave Response:AA 06 RRRR VVVV CCCC

AA	1 byte	Slave Address,0-255
0x06	1 byte	Function Code 6
RRRR	2 byte	Register Addr (High8bits first)
VVVV	2 byte	Register Value (High8bits first)
CCCC	2 byte	CRC CHECKSUM

**Example:Write Register 0x0200,that is change modbus slave address to 2.**

#### Master Request: 01 06 0200 0002 09B3

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x06
Register Addr.	2 byte	0x0200
Register Value	2 byte	0x0002
Checksum	2 byte	0x09B3

#### Slave Response: 01 06 0200 0002 09B3

Slave Addr.	1 byte	0x01
Function Code	1 byte	0x06
Register Addr.	2 byte	0x0200
Register Value	2 byte	0x0002
Checksum	2 byte	0x09B3

## 7.4.4 Function Code 16 Protocol Example

**Master Request:AA 10 RRRR NNNN MM VVVV1 VVVV2 ...CCCC**

AA	1 byte	Slave Address,0-255
0x10	1 byte	Function Code 0x10
RRRR	2 byte	Starting Register Addr
NNNN	2 byte	Quantity of Register to write
MM	1 byte	Register Data Byte Count
VVVV1	2 byte	Register Value(High8bits first)
VVVV2	2 byte	Register Value(High8bits first)
...	...	Register Value(High8bits first)
CCCC	2 byte	CRC CHECKSUM

**Slave Response:AA 10 RRRR NNNN CCCC**

AA	1 byte	Slave Address,0-255
0x10	1 byte	Function Code 0x10
RRRR	2 byte	Starting Register Addr
NNNN	2 byte	Quantity of Register to write
CCCC	2 byte	CRC CHECKSUM

**Example:Write Register 0x0200-0x0201,that is set slave address to 1,and baudrate to 19200bp.**

**Master Request:01 10 0200 0002 04 0001 0004 BACC**

0x01	1 byte	Slave Addr.
0x10(HEX)	1 byte	Function Code 0x10
0x0200	2 byte	Starting Register Addr
0x0002	2 byte	Quantity of Register to write
0x04	1 byte	Register Data Byte Count
0x0001	2 byte	Register Value: Slave Address 1
0x0004	2 byte	Register Value: Baudrate 19200bps
0xBACC	2 byte	CRC CHECKSUM

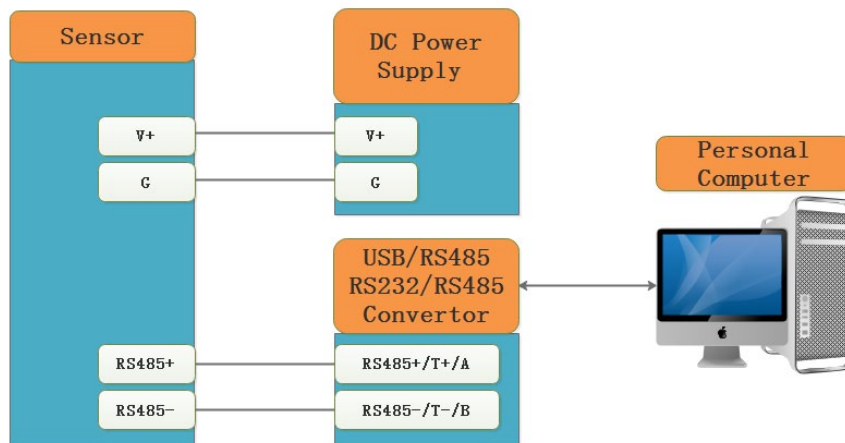
**Salve Response:01 10 0200 0002 4070**

0x01	1 byte	Slave Addr.
0x10(HEX)	1 byte	Function Code 0x10

0x0200	2 byte	Starting Register Addr(High8bits first)
0x0002	2 byte	Quantity of Register to write(High8bits first)
0x4070	2 byte	CRC CHECKSUM

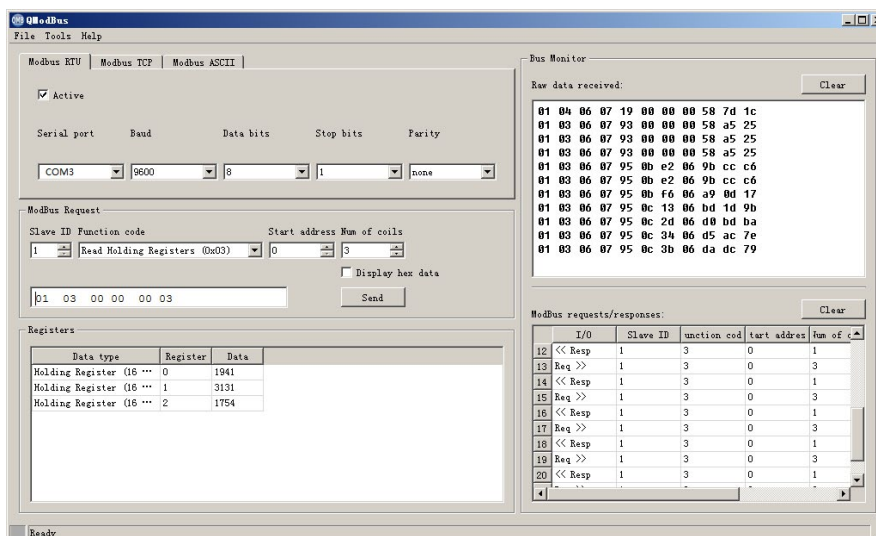
## 8 Software Configuration Utility

### 8.1 Hardwar Setup



### 8.2 Universal Modbus Comm Utility

You can use software listed below to try reading/writing the register of sensor, <https://github.com/ed-chemnitz/qmodbus/releases>



## 8.3 SensorOneSet Configuration Utility

SensorOneSet is a configuration utility to read/set sensor config for all of our serial communication sensor products. Please contact us if you need the English version.

# Appendix

## Copyright and Trademark

This document is copyrighted, 2019, by Dalian Endeavour Technology Co., Ltd. All rights are reserved. Dalian Endeavour Technology Co., Ltd. reserves the right to make improvements to the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission. Information provided in this manual is intended to be accurate and reliable. However, Dalian Endeavour Technology Co., Ltd. assumes no responsibility for its use, nor for any infringements upon the rights of third parties, which may result from its use.

INFWIN® is the trademark of Dalian Endeavour Technology Co., Ltd.

## Version Control

<b>Date</b>	<b>Version</b>	<b>Comment</b>	<b>Updated by</b>
2015-04-23	V1.0	Initial Creation	fg49597