SPECIFICATION

Product Name: Single Beam NDIR CO_2 Sensor Module

Item No.: CM1106

Version: V0.2

Date: February 27, 2019

Writer	Audit	Approved
Mei Yang		

Revision

No.	Version	Content	Reviser	Date
1	V0.2	Add batch board source code	Mei yang	2019/2/27

Single Beam NDIR CO$_2$ Sensor Module

CM1106

Applications

- HVAC industry
- IAQ monitor
- Air purifier
- Automotive
- IoT devices
- Plant growth

Description

CM1106 is a single beam NDIR CO$_2$ sensor module, based on non-dispersive infrared (NDIR) technology, which can detect CO$_2$ concentration of indoor air. With high accuracy, high stability, small size, it is widely used for ventilation system, air purifier, air conditioner, HVAC transmitter, IAQ monitor, etc.

Features

- NDIR technology with independent intellectual property
- High accuracy, long term stability, long life (>10years)
- Temperature calibration within whole measurement range
- Auto-calibration, no need of maintenance
- Signal output PWM / UART / I²C
- Small size and compact structure, easy to install

Working Principle

The main components of an NDIR CO$_2$ sensor are an infrared source, a sample chamber, a filter and an infrared detector. The infrared light is directed by the infrared source passing through the gas chamber towards the detector.
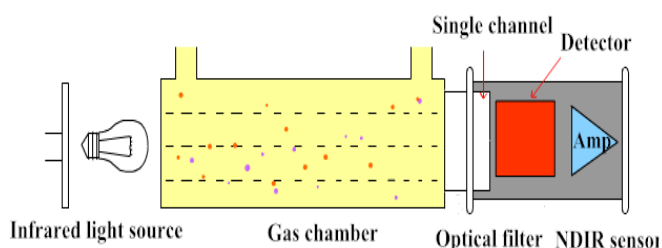

CO$_2$ molecules inside the gas chamber will only absorb a specific wavelength of the light. The filter allows only the specific wavelength corresponded to pass through it. The detector measures the intensity of infrared light that is related to the intensity of CO$_2$ and can be described through the Lambert-Beer's Law. The change in sensor signal reflects the change in gas concentration.

Specifications

Single Beam NDIR CO$_2$ Sensor Specification	
Target gas	Carbon dioxide (CO2)
Operating principle	Non-dispersive infrared (NDIR)
Measurement range	0 ~ 5,000ppm
Working temperature	-10°C ~ 50°C
Working humidity	0 ~ 95%RH (Non-condensing)
Storage temperature	-30°C ~ 70°C
Storage humidity	0 ~ 95%RH (non-condensing)
Accuracy	± (50ppm + 5% of reading) @25 ±2℃, 50 ±10%RH
Sampling frequency	1 sec
Time to first reading	≤30 sec
Power supply	DC 3.5V ~ 5.5V
Ripple wave	<50mV
Working current	<45mA
Dimensions	33x19.7x8.9mm (pin is not included)
Weight	5g
Signal output	UART: TTL PWM: linear output I²C
PWM output	Output high level minimum duration: 2ms (0ppm)
	Output high level maximum duration: 1,004ms (5,000ppm)
Alarm output (Reserved)	According to customer's actual usage environment and requirements
Life span	≥10 years

Dimensions and Connector

1. Dimensions (Unit mm, tolerance ±0.2 mm)
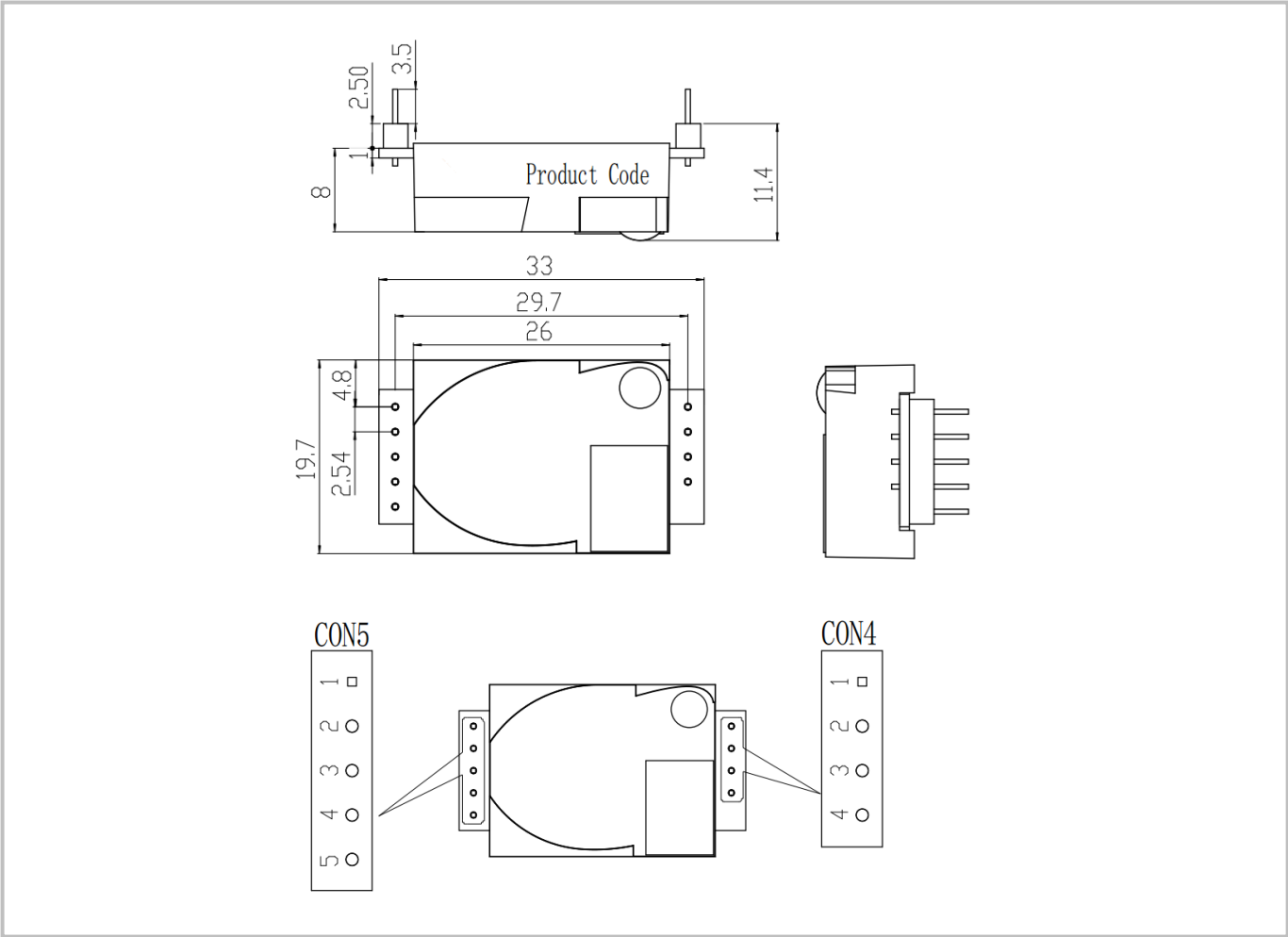
2. I/O Connector Pinout

CON5			CON4		
Pin	Name	Description	Pin	Name	Description
1	+3.3V	Power supply output (+3.3V/100mA)	1	+5V	Power supply input (3.5V ~ 5.5V)
2	RX/SDA	UART-RX (Receiving)/I²C data, compatible with 3.3V and 5V communication	2	GND	Power supply input (GND)
3	TX/SCL	UART-TX (Sending)/I²C clock, Compatible with 3.3V communication	3	A	Alarming
4	R/T	Output mode exchange: High level or floating is UART communication mode, low level is I²C communication mode	4	PWM	PWM output
5	CA	Manual calibration			

Typical Application Circuit

Application scene: UART TTL serial port output
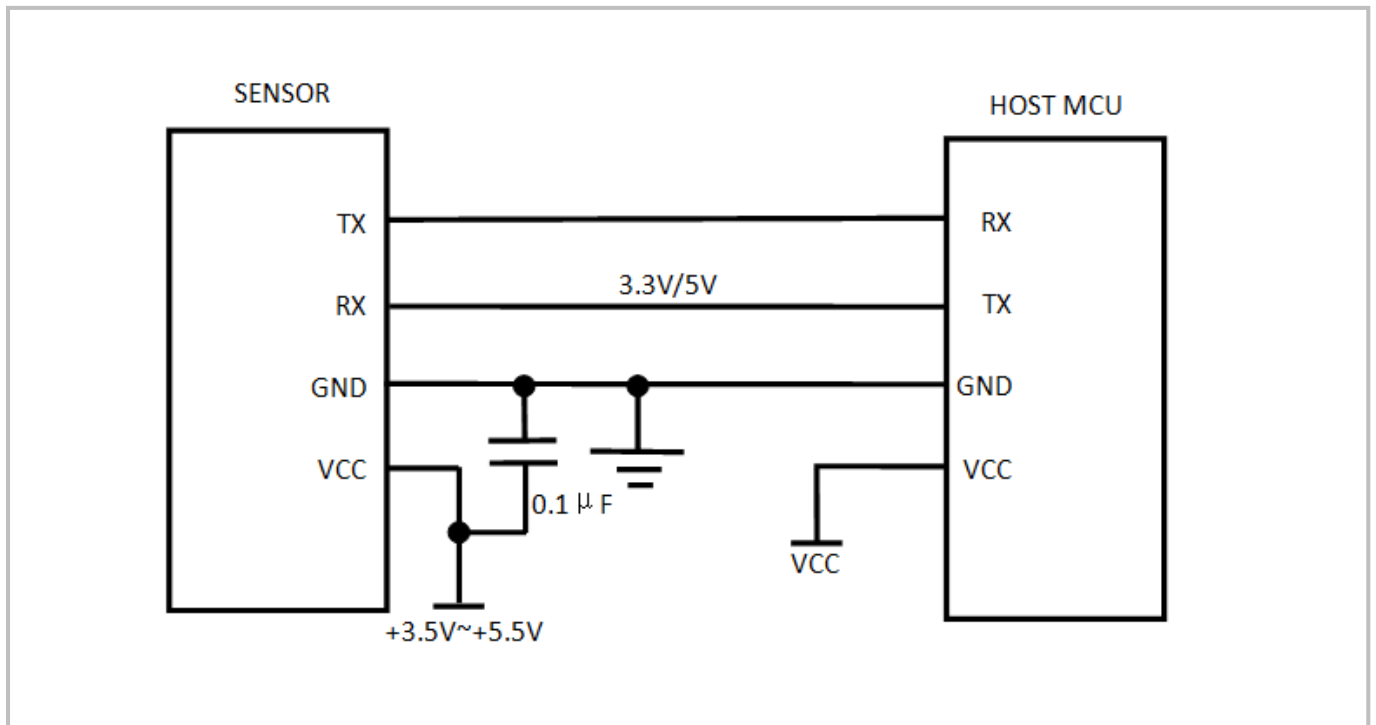
Description of Calibration

Auto Calibration:

Rough installing, non-correct soldering and transportation might result in a reducing of sensor reading accuracy and zero drift, sensor will correct the drift by the built-in self-correcting logic. Powering on the sensor for 15 days continuously, it will record the lowest CO_2 concentration measurement value during these 15 days. Sensor will do auto calibration after 15 days and will regard the outdoor fresh air CO_2 concentration (400ppm) as baseline. In order to ensure the reading accuracy after auto calibration, please make sure the working environment of sensor can reach the outdoor fresh air level, that is to say, the CO_2 concentration of sensor can reduce to the outdoor air level (400ppm) during the 15 days.

Note: Please contact with Cubic for more detailed auto calibration strategy.

Manual Calibration:

Rough installing, non-correct soldering and transportation might result in a reducing of sensor reading accuracy and zero drift. If need to recover accuracy quickly after installing, you can do manual calibration. To put the sensor in the environment where the CO_2 concentration level can reach 400ppm, and to ensure the CO_2 concentration in this environment is stable before calibration. The CA pin of sensor should be well connected at least 2 seconds when doing the manual calibration. Sensor will activate the calibration program after 6 seconds. In addition, sensor also can do manual calibration by sending command, please refer to the communication protocol for more details.
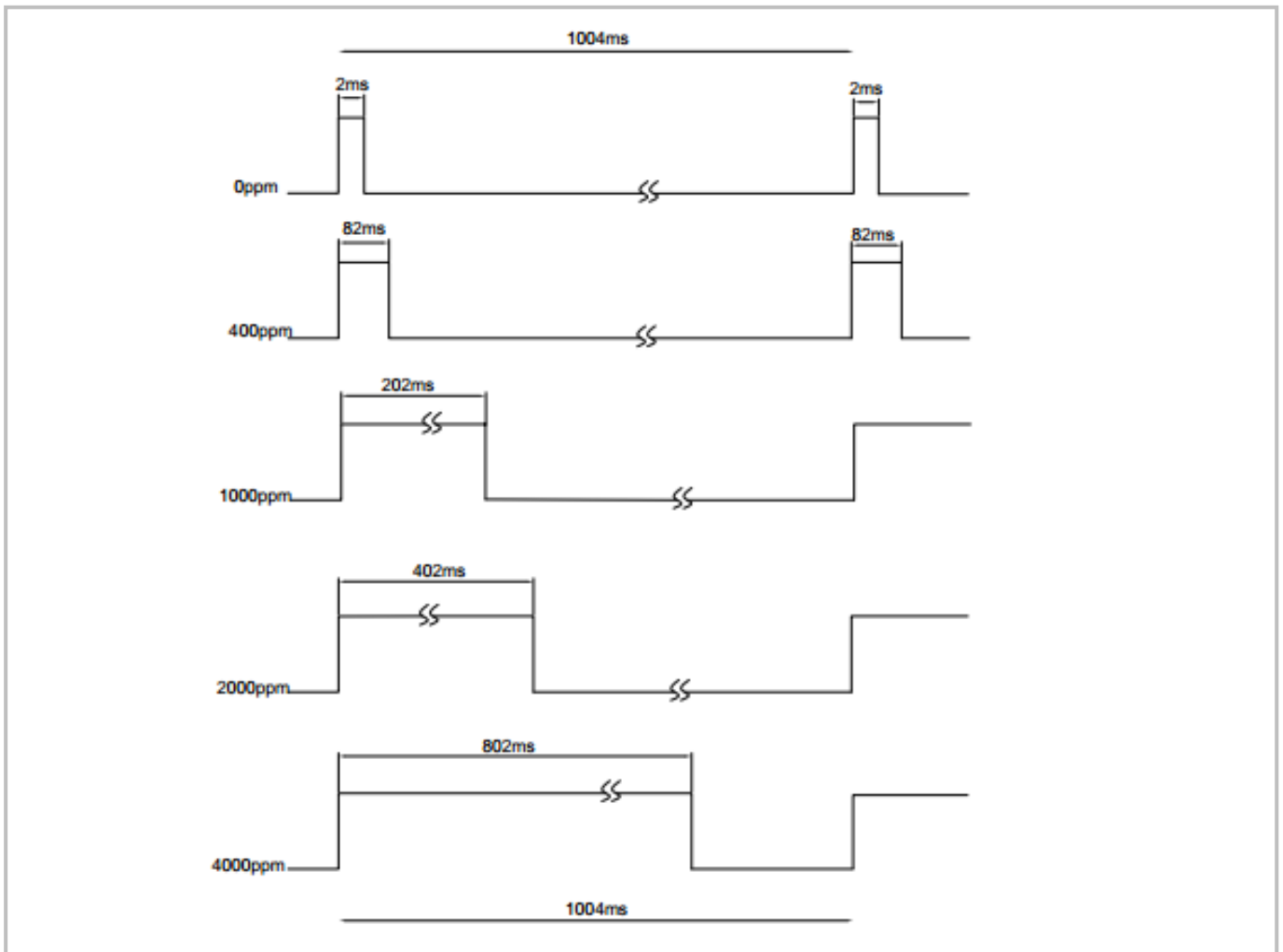
PWM and Alarm Output

PWM cycle: 1,004ms

Positive pulse width: 0 ~ 5,000: (PPM / 5) + 2ms

CO_2 concentration detection value: 500ppm range: (PWM Positive pulse width -2) * 5

PWM output schema:

Alarm Output

If the CO_2 concentration rises up to more than 1,000ppm, the alarming will be triggered and output high level.

When the CO_2 concentration goes down to below 800ppm, the alarming will stop and output low level.

Note

Connect the pin of PWM to the oscilloscope.

Add a pull-up resistor around 5K-10K between the pin of PWM and power supply.

Product Installation

1. In order to ensure airflow diffusion into the sensor inner, make sure the minimum distance between the area of waterproof filter and the other components is 1.5 mm, otherwise, quick response time of the sensor will be effected.

Reference as below:

1.5mm

1.5mm

Waterproof Filter
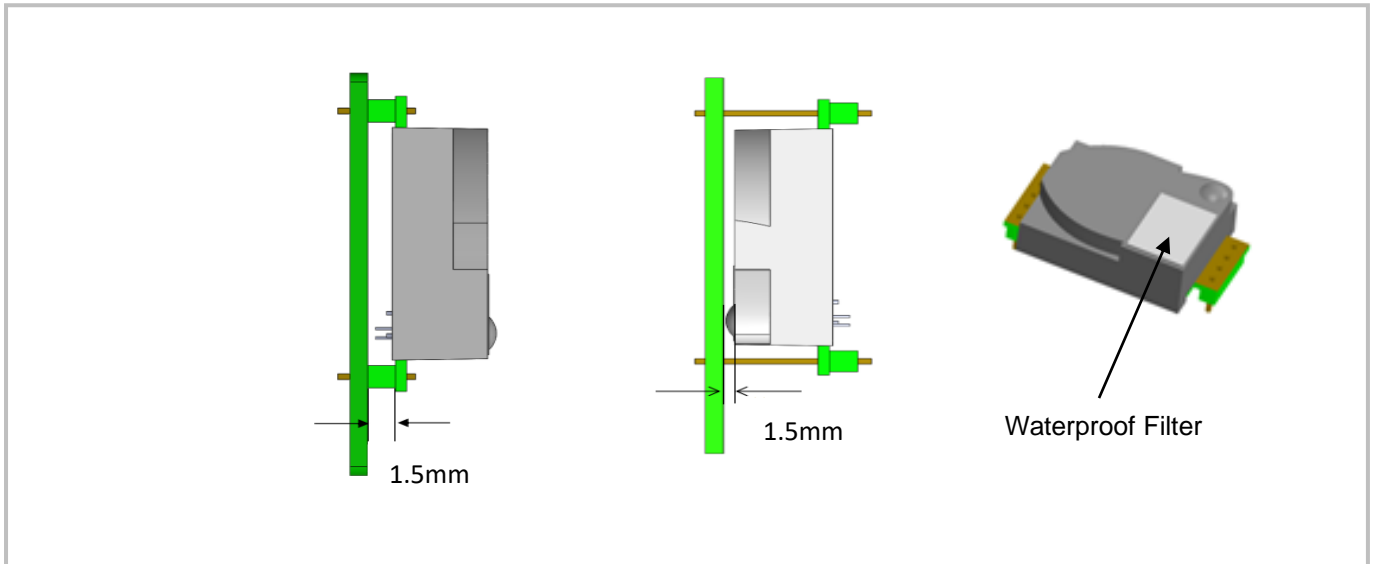
2. To avoid the influence of stress on sensor, please soldering by hand as much as possible when mounting the sensor to the PCB.

Reference as below:

Batch Board Source Code

1. Example

```c
// send command
void Send_Cmd_To_Sensor( unsigned char Cmd, unsigned char Channel )
{
    uint8_t Cnt = 0;
    Switch_Channel( Channel );// Switch channel
    Send_Data[0] = 0x11;
    switch( Cmd )
    {
            //【read real time value】
            case Result_info:{
                Send_Data[1] = 0x01;
                Send_Data[2] = 0x01;
                Send_Data[3] = 0xED;
                        }break;
                    default : break;
    }
    // check sum
    Send_Data[ Send_Data[ 1 ] + 2 ] = CS_count( &Send_Data[0], Send_Data[1] + 2 );
        for( Cnt = 0; Cnt < Send_Data[1] + 3; Cnt++ )
        {
                Putchar_To_Sensor( Send_Data[ Cnt ] );
        }
}
//sensor data analyze
void Serson_Analysis_Command( uint8_t Date )
{
    UART3_buff[ UART3_rev ] = Date;
    UART3_rev++;
    if( UART3_buff[0] == 0x16 ){ UART3_rev_OK[0] = 1; }// Check the frame head
    else{ UART3_rev = 0; }
    if( UART3_rev_OK[0] && UART3_rev > 3   )
{
```

```c
                if( UART3_buff[1] + 3 == UART3_rev ){
                    memset( UART3_rev_OK, sizeof( UART3_rev_OK ), 0 );
                    UART3_rev = 0;
                    UART3_rev_flge = 1;//response received successfully
                }
        }
                if( UART3_rev == sizeof( UART3_buff ) ){// Communication confusion

                memset( UART3_rev_OK, sizeof( UART3_rev_OK ), 0 );
                memset( UART3_buff, sizeof( UART3_buff ), 0 );
                UART3_rev = 0;
        }
}
// Process the data returned by the sensor
void REV_Sensor( void )
{
    uint8_t   checksum = 0;
    uint16_t Length = 0;
    uint8_t   temprx[ 30 ] = { 0 };
    Length = UART3_buff[ 1 ] + 3;
    memcpy( temprx, UART3_buff, Length );
    memset( UART3_buff, 0, Length );// Empty buffer
    checksum = Add_Check( temprx, Length - 1 );//check sum
    if( ( temprx[0] == 0X16 ) && ( checksum == 1 ) )
    {
        switch( temprx[ 2 ] )
        {
                case 0x01:{//real time data
                    CO2Concentration = temprx[3] << 8 | temprx[4];//concentration
                }break;
            default:
             break;
        }
    }
}
```

2. Computational results

CO2 concentration in PPM

序号	浓度
1	2276
2	2291
3	2253
4	2238

UART Communication Protocol

1. General Statement

1).The data in this protocol is all hexadecimal data. Example: "46" for decimal [70].

2). Baud rate: 9,600, Data Bits: 8, Stop Bits: 1, Parity: No, Flow control: No.

3). [xx] is for single-byte data (unsigned, 0 ~ 255); for double data, high byte is in front of low byte.

2. Format of Serial Communication Protocol

Sending format of test software:

Start Symbol	Length	Command	Data1	…	Data n.	Check Sum
HEAD	LEN	CMD	DATA1	…	DATAn	CS
11H	XXH	XXH	XXH	…	XXH	XXH

Detail description on protocol format:

Protocol Format	Description
Start Symbol	Sending by test software is fixed as [11H], module response is fixed as [16H]
Length	Length of frame bytes= data length +1 (including CMD+DATA)
Command	Command
Data	Data of writing or reading, length is not fixed
Check Sum	Cumulative sum of data= 256 - (HEAD+LEN+CMD+DATA)

3. Command Table of Serial Protocol

Item No.	Function Name	Command
1	Read measured result of CO_2	0x01
2	Open/ Close ABC and set ABC parameter	0x10
3	Calibrate concentration value of CO_2	0x03
4	Read the serial number of the sensor	0x1F
5	Read software version	0x1E

4. Detail Description of RS232 Protocol

4.1 Read Measured Result of CO₂

Send: 11 01 01 ED

Response: 16 05 01 DF1 ~ DF4 [CS]

Function: Read measured result of CO2 (Unit: ppm)

Note:

CO_2 measured result= DF1 * 256 + DF2

DF3 DF4 is reserved

Example:

Response: 16 05 01 02 58 00 00 8B

Explanation:

Hex is converted to decimal: 02 is 02; 58 is 88. CO_2 concentration= 02 * 256 + 88= 600ppm

4.2 Open / Close ABC and Set ABC Parameter

Send: 11 07 10 DF1 DF2 DF3 DF4 DF5 DF6 [CS]

Response: 16 01 10 D9

Explanation:

DF1: reserved, default 100 (0x64)

DF2: open/close auto calibration (0: open; 2: close)

DF3: calibration cycle (1-30 days optional, default is 15days)

DF4: High base value (2 bytes)

DF5: Low base value (2 bytes)

DF6: reserved, default is 100 (0x64)

Note: The auto calibration function is open with 15 days calibration cycle by default.

The default value of DF4 and DF5 is 400, that is DF4: 01; DF5:90

4.2.1 Close ABC

In our sensor, the ABC function is default open status. If you want to close it, you should set the DF2= 2.

Example:

Send: 11 07 10 64 02 07 01 90 64 76

Response: 16 01 10 D9

4.2.2 Open ABC and Set Calibration Cycle

When ABC function is closed and you want to re-open ABC function, you should set the DF2=0.

Example:

you could send below command to open ABC function and set the calibration cycle 7 days.

Send: 11 07 10 64 00 07 01 90 64 78

Response: 16 01 10 D9

4.2.3 Change the Calibration Cycle

The calibration cycle is 15 days by default. For example, if you want to change the calibration cycle to 10 days, you should set the DF3= 10.

Example:

Send: 11 07 10 64 00 0A 01 90 64 75

Response: 16 01 10 D9

4.3 Calibration of CO_2 Concentration

Send: 11 03 03 DF1 DF2 [CS]

Response: 16 01 03 E6

Function: Calibration of CO_2 concentration

Note:

1. Calibration target value= DF1 * 256 + DF2 Unit: PPM, range (400 ~ 1,500ppm)

2. Before calibration, please make sure CO_2 concentration in current ambient is calibration target value. Keeping this CO_2 concentration for two 2 minutes, then began calibration.

Example:

When you need to calibrate CO_2 concentration of the sensor to 600ppm, send command:

Send: 11 03 03 02 58 8F

Hex is converted to decimal: 02 is 02; 58 is 88

CO_2 concentration= 02 * 256 + 88= 600ppm

4.4 Read the Serial Number of the Sensor

Send: 11 01 1F CF

Response: 16 0B 1F (SN1) (SN2) (SN3) (SN4) (SN5) [CS]

Function: Read the serial number of the sensor

Note: Read the serial number of the sensor. SNn: 0~9999, 5 integer form 20-digit number

4.5 Read Software Version

Send: 11 01 1E D0

Response: 16 0C 1E DF1 ~ DF11 [CS]

Function: Read software version

Note:

DF1 ~ DF10: stand for ASCII code of software version, DF11 is reserved

Example:

When the sensor version is CM V0.0.20, response data as follows:

Hexadecimal converted to ASCII code:

Note: when 20 converted to ASCII code, it equals to blank space.

16 0C 1E 43 4D 20 56 30 2E 30 2E 32 30

CM V0.0.20

# I²C Communication Protocol

1. Timing Diagram Introduction

1.1 Common Description

a. This protocol is based on standard I²C timing sequence, the clock frequency is 10kHz ~ 400kHz.

b. Use big-endian format, the most significant bit to be sent first.

1.2 I²C Sequence Diagram Introduction

Item	Parameter			Unit
	Min	Type	Max	
fSCL (SCL clock frequency)	10		400	KHz
tHD.STA (hold time of the starting bit)		0.6		us
tSU.STA (setup time of the starting bit)		0.6		us
tHD.DAT (hold time of the data)		0		ns
tSU.DAT (setup time of the data)		250		ns
tSU.STO (setup time of the stop bit)		4		us

Note: SCL clock frequency is generated by the master device with the range 10khz ~ 400khz.
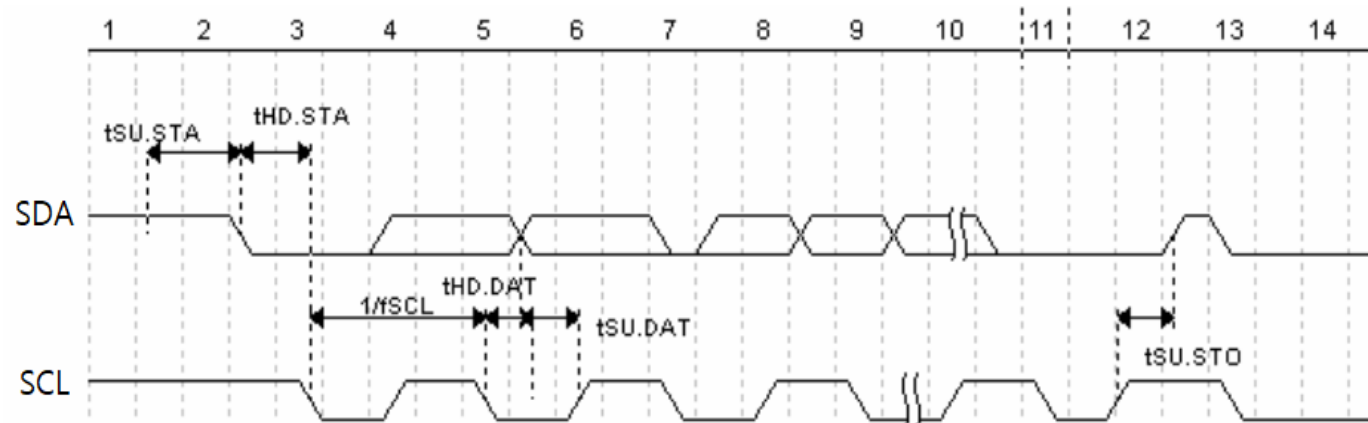
Picture1: I²C clock introduction

1.3 Basic Data Transmission Formats

S	SA	W	A	D	A	D	…	D	A/~A	P
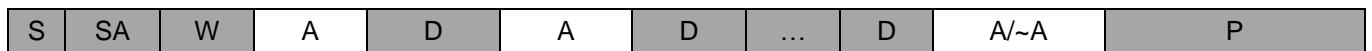
Picture 2: The general data format sends from the master device to the slave

S	SA	R	A	D	A	D	…	D	A/~A	P

Picture 3: The general data format received from the slave device to the master device

The meaning of the symbol in picture 1.2 and picture 1.3:

S: start condition

SA: slave address

W: write bit

R: read bit
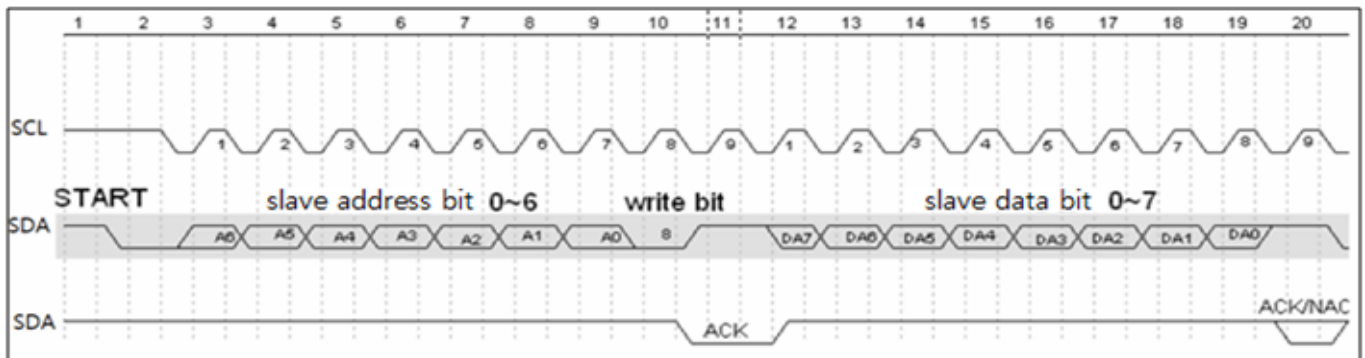
A: acknowledge bit

~A: not acknowledge bit

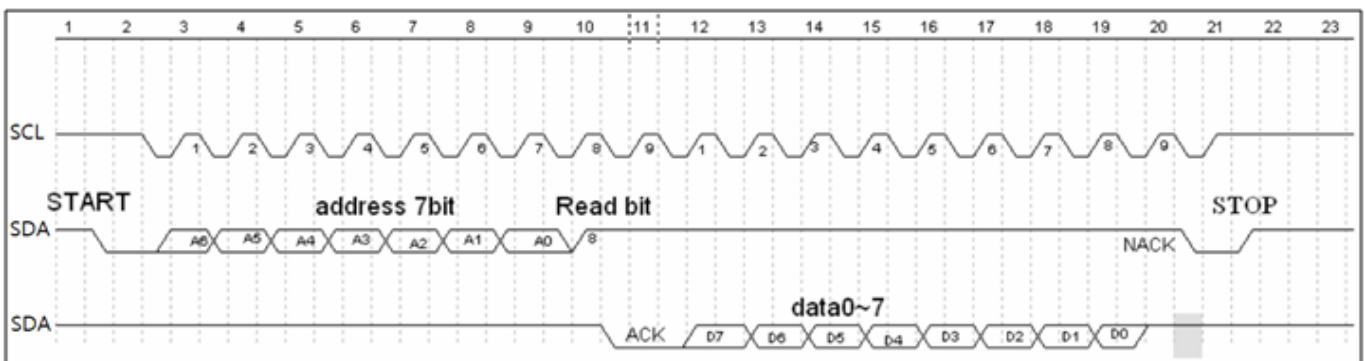D: data, each data is 8bit

P: stop condition

Shadow: The signal generated from the master device

No Shadow: The signal generated from the slave device

1.4 Timing Diagram

Picture 4: The address byte sent from the master device

Picture 5: The master device read a byte from the slave device

1.5 Notes

The performance of the MCU which is used in the sensor is not very high. If you use I/O port to simulate I²C master device, it is suggested to reserve a period before and after ACK signal (such as 100us), after sending every byte (8 bit) to leave enough time for the SCM to process the data. Within requirements of speed, it is recommended to lower the reading speed as much as possible.

2. Measuring Function

Format of command

Format of sending:[CMD][DF0]……[DFn]

[CMD]	Command number, for distinguishing different command.
[DF0]…… [DFn]	The command with parameter item and optional items.
Format of response：	[CMD][DF0]…… [DFn] ［CS］
[CMD]:	Command number
[DF0]…… [DFn]:	Effective data
[CS]:	Data check bit = -([CMD]+ [DF0]+……[DFn]) Only use the lowest bit

2.1 Statement of Measuring Command

The slave address is 0x31, the data command of the slave device is as below:

No.	Function name	CMD	Function description
1	Measure result	0x01	Read measuring result
2	Auto zero setting specification setting	0x10	Set auto zero setting specification
3	Calibration	0x03	Calibration of CO_2
4	Read the serial number of the sensor	0x1F	Read the serial number of the sensor
5	Check software version	0x1E	Read software version

2.2 Measuring Result

The master device should send command of measuring result.

Send: 0x01

Response: [0x01][DF0][DF1][DF2][CS]

Note:

1. Sensor starts measuring result status once receiving the command 0x01. After this, all the data which I²C read will be such status format data, until the sensor receives new command or re-powering on.

2. Data format, master device receives DF_0 first，and then receives CS at last.

Remark	Status Bite	Decimal Effective Reading Value Range	Relative Value	Multiple
CO_2 measuring result	[DF0] [DF1]	0 ~ 5,000 ppm	0 ~ 5,000 ppm	1
Status bit	[DF2]			1

CO_2 measuring result: $DF0 * 256 + DF1$, Fixed output is 550ppm during preheating period

Status bit: 0: Preheating; 1: Normal operation; 2: Operating trouble; 3: Out of FS, 5: Non-calibrated

Example:

The master device reads some data: Read 3 bit.

0x01 0x03 0x20 0x01 0xDB

CO_2 measuring result= (0x03 0x20) $_{hexadecimal}$= (800) $_{decimal}$= 800 ppm

Status bit: 0x01 means working normally

[CS]= -(0x01+0x03+0x20+0x01), Only keep the lowest bite.

2.3 Auto Zero Setting Specification Setting

Send: 0x10 [DF0] [DF1] [DF2] [DF3] [DF4] [DF5]
Response: [0x10] [DF0] [DF1] [DF2] [DF3] [DF4] [DF5] [CS]

Format description:

1. Sensor will be auto calibration specification setting status after receiving command 0x10. After this, all the data which I^2C read are the data in this status format, until sensor receives new command or repowering on.

2. Data format, the master will receive [DF0] firstly, and receive [CS] at last.

The result is calculated by high bit in front

Remark	Data Bite	Decimal Effective Reading Value Range	Relative Value	Multiple
Wrong code accelerate value	[DF0]	By default: 100	100	1
Zero setting switch	[DF1]	0 or 2	0: Open, 2: Close	1
Calibration period	[DF2]	1 ~ 15	1 ~ 15	1
Calibration concentration value	[DF3] [DF4]	400 ~ 1,499	Suggest 400 ~ 1,499	1
Reserved byte	[DF5]	By default: 100	100	1

2.4 Calibration

The master device should send command of calibration.

Send: 0x03 [DF0] [DF1]
Response: [0x03] [DF0] [DF1] [CS]

Note:

1. Sensor starts calibration status once receiving command 0x03. After this, all the data which I^2C read will be such status format data, until the sensor receives new command or re-powering on.

2. Data format, master device receives DF0 first, and then receives CS at last. The result is calculated by high bit in front: [DF0] * 256 + [DF1].

Remark	Data Bite	Decimal Effective Reading Value Range	Relative Value	Multiple
Adjust value	[DF0] [DF1]	400 ~ 1,500	400 ~ 1,500ppm	1

2.5 Read the Serial Number of the Sensor

Send: 0x1F

Response: [0x1F] [DF0] [DF1] [DF2] [DF3] [DF4] [DF5] [DF6] [DF7] [DF8] [DF9] [CS]

Note:

1. Sensor starts device code output status once receiving the command 0x1F. After this, all the data which I²C read will be such status format data, until the sensor receives new command or re-powering on.

2. Data format, the master device receives [DF0] first, and then receives [CS] at last. The result is calculated by high bit in front.

Remark	Data Bit	Decimal Effective Reading Value Range	Relative Value	Multiple
Integer type 1	[DF0] [DF1]	0 ~ 9,999	0 ~ 9,999	1
Integer type 2	[DF2] [DF3]	0 ~ 9,999	0 ~ 9,999	1
Integer type 3	[DF4] [DF5]	0 ~ 9,999	0 ~ 9,999	1
Integer type 4	[DF6] [DF7]	0 ~ 9,999	0 ~ 9,999	1
Integer type 5	[DF8] [DF9]	0 ~ 9,999	0 ~ 9,999	1

3. Five integers form a 20-bit code.

2.6 Read Software Version

Send: 0x1E

Response: [0x1E] [DF0] [DF1] [DF2] [DF3] [DF4] [DF5] [DF6] [DF7] [DF8] [DF9] [CS]

Note:

1. Sensor starts software version output status once receiving the command 0x1E. After this, all the data which I²C read will be such status format data, until the sensor receives new command or re-powering on.

2. Data format, the master device receives DF_0 first, and then receives CS at last. [DF_0] …… [DF_9] is ASCII.

3. Communication Diagram

Diagram: The master device read two bytes continuously from the slave device.

The slave machine address: 0x31= 0110001 (the machine address is 7 bit) + read/write bit (1bit)

The slave data address: 0x01= 00000001

Step 1: The master device sends the address of the slave device+ write bit: 0110001+0 → 01100010 (0x62); at this time, the master device is in sending status.

Step 2: The master device sends the slave data address: 0x01
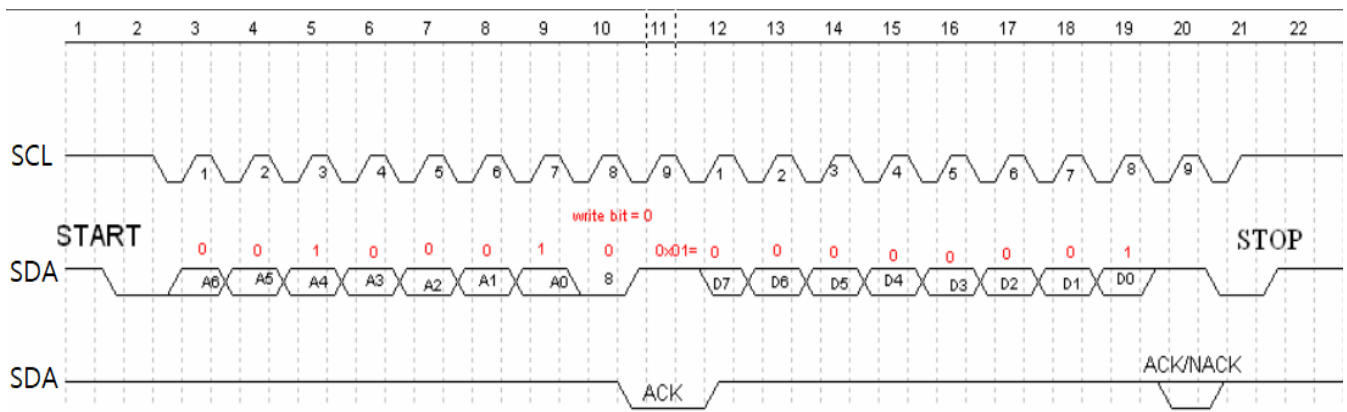
Picture 6: The timing diagram send from the master device

Step 3: The master device send the slave machine address+ read bit: 0110001+1 → 01100011 (0x63); at this time, the master device is in receiving status.
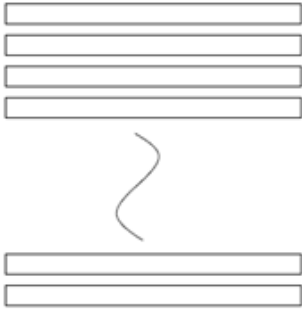
Step 4: The master device sends the answer bit after receiving a one-bit data and the slave continuously sends the next data. If the master device sends the no-answer bit after receiving a one-bit data, then the communication will stop.
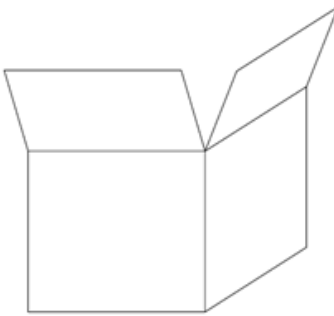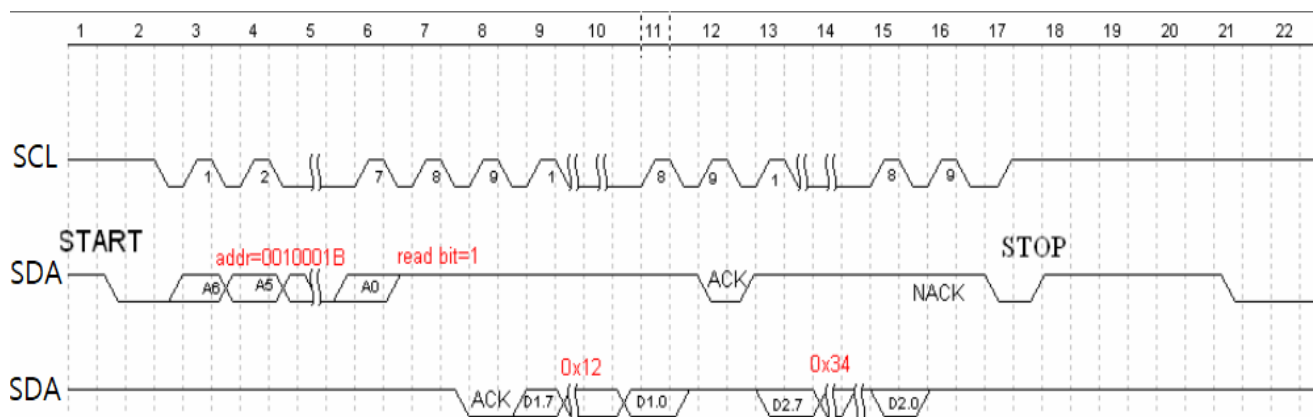

Picture 7: The master device receives the data from the slave device

Packing Information

Tray Size

Tray Q'ty 18 layers

Carton

Sensor per Tray	Tray Q'ty	Sensor per Carton	Carton Dimensions	Packing Material
60 pcs	18 layers	1080 pcs	395 * 310 * 480 mm	Red anti-static EPE

After-Sales Services and Consultancy

Wuhan Cubic Optoelectronics Co., Ltd.

Tel: +86 (0)27 81628827 / Fax: +86 (0)27 81628821 / +82 010-5349-8385 (South Korea Contact)

Add: Fenghuang No.3 Road, Fenghuang Industrial Park, Eastlake Hi-tech Development Zone, Wuhan 430205, China

Web: www.gassensor.com.cn / E-mail: info@gassensor.com.cn / malbonair@naver.com (South Korea Contact)