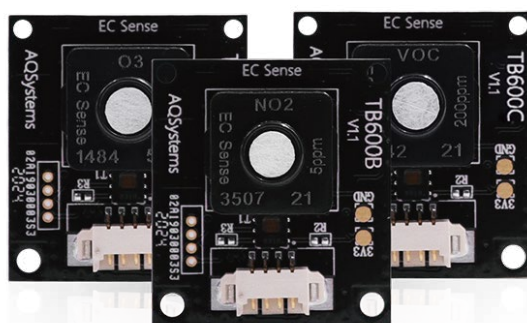




TB600B&C 智能气体传感器模组

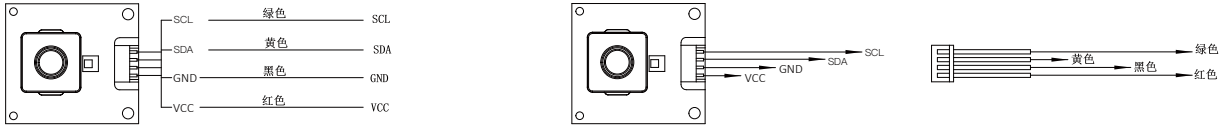
—— I²C用户通信协议 ——

1、I²C的介绍

I²C是一种串行总线，用于同步通信和半双路传输。它有两根信号线，一根是双向的数据线SDA，另一根是时钟线SCL，可以发送和接收数据。

I²C串行总线通讯具有接口少、通讯效率高等优点。所有接到I²C总线设备上的串行数据SDA都接到总线的SDA上，各设备的时钟线SCL接到总线的SCL上。

IIC 连接



2、通讯模式

TB600B&C传感器模组采用I²C接口通信

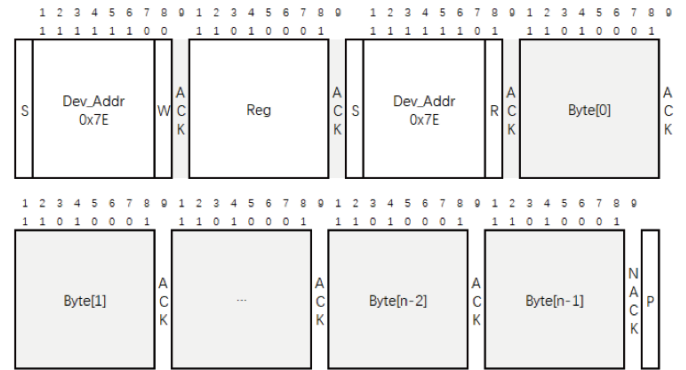
通讯参数如下：

传感器模组的接口兼容信号电压：3.3V

传感器模组支持I²C标准模式，SCL最大时钟频率为：20kHz

I²C 7位默认地址：0x7E（始终存在）。用户可以设置一个新地址，设置完成后0x7E依旧可用

数据读写时间间隔要求：>=1s



3、单位和传感器类型

附表 1 传感器浓度单位代码表：

浓度单位	ppm	ppb	%
值(Hex)	0x02	0x04	0x08

附表 2 传感器类型代码表：

传感器类型	HCHO	VOC	CO	Cl ₂	H ₂	H ₂ S	HCl	HCN	HF	NH ₃	NO ₂
值(Hex)	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F	0x20	0x21
传感器类型	O ₂	O ₃	SO ₂	HBr	Br ₂	F ₂	PH ₃	AsH ₃	SiH ₄	GeH ₄	B ₂ H ₆
值(Hex)	0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C
传感器类型	BF ₃	WF ₆	SiF ₄	XeF ₂	TiF ₄	SMELL	IAQ	AQI	NMHC	SO _x	NO _x
值(Hex)	0x2D	0x2E	0x2F	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
传感器类型	NO	C ₄ H ₈	C ₃ H ₈ O ₂	CH ₄ S	C ₈ H ₈	C ₄ H ₁₀	C ₂ H ₆	C ₆ H ₁₄	C ₂ H ₄ O	C ₃ H ₉ N	C ₂ H ₇ N
值(Hex)	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F	0x40	0x41	0x42

传感器类型	C ₂ H ₆ O	CS ₂	C ₂ H ₆ S	C ₂ H ₆ S ₂	C ₂ H ₄	CH ₃ OH	C ₆ H ₆	C ₈ H ₁₀	C ₇ H ₈	CH ₃ COOH
值(Hex)	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C

传感器类型	ClO ₂	H ₂ O ₂	N ₂ H ₄	C ₂ H ₈ N ₂	C ₂ HCl ₃	CHCl ₃	C ₂ H ₃ Cl ₃	H ₂ Se		
值(Hex)	0x4D	0x4E	0x4F	0x50	0x51	0x52	0x53	0x54		

4、数据寄存器以及数据解析

4.1 获取传感器实时数据

Real-time data (Read Only)

Reg	0xD1									
Reg data len	7									
byte index	0	1	2	3	4	5	6			
Definition	Concentration		Temperature		Humidity		add-8 sum of byte[0] to byte[5]			

Byte[0:1]: float concentration = (float)((uint16_t)(Byte[0]<<8)|Byte[1])

小数位数通过寄存器-0xD2获取，例如获取到的小数位数为3，则浓度值需要除以1000

concentration = concentration/1000.0f

Byte[2:3]: float temperature = (float)((int16_t)((Byte[2]<<8)|Byte[3]))

温度包含固定的2位小数，单位为摄氏度

Temperature = temperature/100.0f

Byte[4:5]: float rh = (float)((uint16_t)((Byte[4]<<8)|Byte[5]))

湿度包含固定的2位小数，单位为%rh

rh = rh/100.0f

Byte[6]: 校验和, Byte[6] = (Byte[0]+ Byte[1]+ Byte[2]+ Byte[3]+ Byte[4]+Byte[5])&0xFF

4.2 获取传感器参数

Parameter (Read Only)

Reg	0xD2									
Reg data len	6									
byte index	0		1	2	3	4	5			
Definition	Concentration decimal num		Sensor type	Unit type	Range		add-8 sum of byte[0] to byte[4]			

Byte[0]: concentration decimal num, 浓度值的小数位数, 在获取浓度时使用

Byte[1]: sensor type, 传感器类型, 请参照 [附表1]

Byte[2]: unit type, 浓度单位, 请参照 [附表2]

Byte[3:4]: uint16_t range = (uint16_t)((Byte[4]<<8)|Byte[5]), 传感器量程

Byte[5]: 校验和, Byte[5] = (Byte[0]+ Byte[1]+ Byte[2]+ Byte[3]+ Byte[4])&0xFF

• 4.3 设置修改I²C地址

Customer define addr (Read/Write), 修改地址后默认地址(0x7E)依然可用

Reg	0xD0	
Reg data len	2	
byte index	0	1
Definition	I ² C addr	add-8 sum of byte[0] to byte[0]

Byte[0]: I²C addr, 用户自定义I²C地址

Byte[1]: Byte[1] = Byte[0]

• 4.4 读取软件版本号

Software version (Read Only)

Reg	0xD3						
Reg data len	7						
byte index	0	1	2	3	4	5	6
Definition	sw_version						add-8 sum of byte[0] to byte[5]

Byte[0:5]: sw_version, 软件版本号(BCD)

例如读取的原始数据: 0x20, 0x23, 0x11, 0x28, 0x18, 0x37

软件版本号为: 202311281837

Byte[6]: 校验和, $\text{Byte}[6] = (\text{Byte}[0] + \text{Byte}[1] + \text{Byte}[2] + \text{Byte}[3] + \text{Byte}[4] + \text{Byte}[5]) \& 0xFF$

• 4.5 读取传感器序列号

Sensor serial (Read Only)

Reg	0xD4					
Reg data len	6					
byte index	0	1	2	3	4	5
Definition	sensor_serial					add-8 sum of byte[0] to byte[4]

Byte[0:4]: sensor_serial, 传感器序列号(BCD)

例如读取的原始数据: 0x01 0x23 0x45 0x67 0x89

传感器序列号为: 0123456789

Byte[5]: 校验和, $\text{Byte}[5] = (\text{Byte}[0] + \text{Byte}[1] + \text{Byte}[2] + \text{Byte}[3] + \text{Byte}[4]) \& 0xFF$

• 4.6 用户标定

Calibrate concentration (Write Only)

Reg	0xD5				
Reg data len	5				
byte index	0	1	2	3	4
Definition	concentration[0]	concentration[1]	concentration[2]	concentration[3]	add-8 sum of byte[0] to byte[3]

注：concentration (浓度值)为32位float类型 (IEEE754 Single precision 32-bit)，写入浓度值必须大于等于0 (concentration>=0)

例如写入的值为：11.2，对应的十六进制数值为：0x41333333

可参考：https://www.binaryconvert.com/convert_float.html

具体写入数据为：

Byte[0]: 0x41

Byte[1]: 0x33

Byte[2]: 0x33

Byte[3]: 0x33

Byte[4]: (Byte[0]+Byte[1]+Byte[2]+Byte[3])&0xFF=(0x41+0x33+0x33+0x33)&0xFF=0xDA

• 4.7 恢复出厂标定

Restore factory calibration (Write Only)

Reg	0xD6	
Reg data len	2	
byte index	0	1
Definition	reset flag (0x01)	add-8 sum of byte[0] to byte[0]

如要恢复出厂设置，必须写入0x01，写入其他数值无效

恢复出厂标定写入数据：

Byte[0]: 0x01

Byte[1]: 0x01

• 4.8 控制运行灯

Control run led (Read/Write)

Reg	0xD7	
Reg data len	2	
byte index	0	1
Definition	switch run led (0x00 or 0x01)	add-8 sum of byte[0] to byte[0]

运行灯状态：0x00为关闭状态，0x01为打开状态

打开运行灯写入数据：

Byte[0]: 0x01

Byte[1]: 0x01

关闭运行灯写入数据：

Byte[0]: 0x00

Byte[1]: 0x00

• 4.9 休眠模式

Switch sleep mode (Read/Write)

Reg	0xD8	
Reg data len	2	
byte index	0	1
Definition	Entry/Exit Sleep mode (0x00 or 0x01)	add-8 sum of byte[0] to byte[0]

模组工作状态：0x00为正常运行模式，0x01为休眠状态

当从休眠模式唤醒后，需要等待2s才能继续通信

进入休眠写入数据：

Byte[0]: 0x01

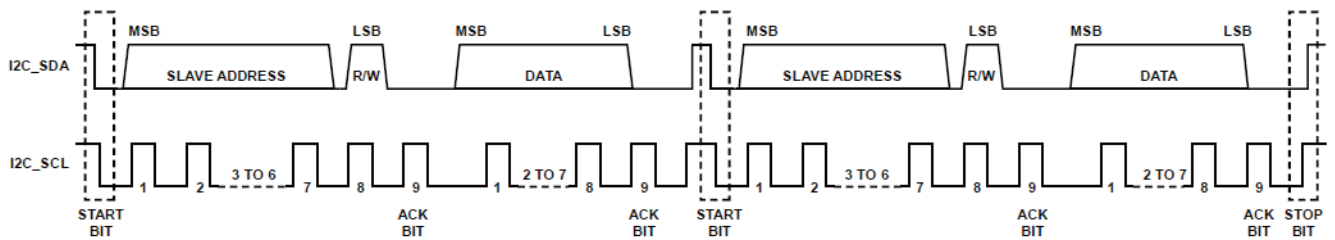
Byte[1]: 0x01

退出休眠写入数据：

Byte[0]: 0x00

Byte[1]: 0x00

» 5、I²C时序



读写接口例程：

```
/******
```

功能：向一个寄存器中写入一个字节数据

dev_add: I2C 7位地址左移一位

register_addr: 写入寄存器地址

data: 写入数据

```
*****/
```

```
void send_data(uint8_t dev_add, uint8_t register_addr, uint8_t data)
```

```
{
    I2C_Start();
    I2C_SendByte(dev_add);
    if(I2C_CheckAck())
    {
        I2C_Stop();
        return;
    }
    I2C_SendByte(register_addr);
    if(I2C_CheckAck())
    {
        I2C_Stop();
        return;
    }
    I2C_SendByte(data);
```

```
if(I2C_CheckAck())
{
    I2C_Stop();
    return;
}
I2C_Stop();
}

/*****
功能: 从register_addr寄存器开始读取read_len个字节
dev_add: I2C 7位地址左移一位
register_addr: 写入寄存器地址
data: 读取数据存储地址
read_len: 读取长度
*****/
void read_data(uint8_t dev_add, uint8_t register_addr, uint8_t* data, uint8_t read_len)
{
    uint8 i = 0;
    I2C_Start();
    I2C_SendByte(dev_add);
    if(I2C_CheckAck())
    {
        I2C_Stop();
        return;
    }
    I2C_SendByte(register_addr);
    if(I2C_CheckAck())
    {
        I2C_Stop();
        return;
    }

    I2C_ReStart();

    I2C_SendByte(dev_add|0x01);
    if(I2C_CheckAck())
    {
        I2C_Stop();
        return;
    }

    data[i] = I2C_ReceiveByte();
    i++;
    for(;i<data_len;i++)
    {
        I2C_SendAck();
        data[i] = I2C_ReceiveByte();
    }
    I2C_SendNAck();
    I2C_Stop();
}
```



德国研发生产中心

德国 EC Sense GmbH

Wangener Weg 3 | 82069 Hohenschäftlarn

座机: +49 (0)8178-99992-10

传真: +49 (0)8178-99992-11

邮箱: office@ecsense.com

网址: www.ecsense.com

亚太区·中国应用设计研发中心

宁波爱氮森科技有限公司

浙江·宁波市鄞州区金谷北路 228 号中物科技园 6号楼

邮编: 315100

座机: 0574-88097236, 88096372

邮箱: info@aqsystems.cn

网址: www.ecsense.cn