



DS4-PID 智能气体传感器模组

Modbus-RTU用户通讯协议

UART接口

本文件详细介绍了DS4-PID 智能气体传感器模组的Modbus-RTU通信协议，用户在使用产品时所需要使用的具体例程。

1. 产品概述和配置参数

1.1 产品概述

该产品使用UART进行通信，具体配置请参阅【1.2 UART配置参数】。

1.2 UART配置参数

波特率	数据位	起始位	停止位	校验位	读写数据间隔
9600	8位	1位	1位	无	>=1s

1.3 CRC-16/Modbus 的计算方式

计算公式为： $x_{16}+x_{15}+x_2+1$

C语言计算函数:

```
uint16_t get_modbus_crc(uint8_t *buffer, uint16_t len)
{
    uint16_t wrcr = 0xFFFF;
    uint8_t temp;
    uint16_t i = 0, j = 0;
    for (i = 0; i < len; i++)
    {
        temp = *buffer & 0X00FF;
        buffer++;
        wrcr ^= temp;
        for (j = 0; j < 8; j++)
        {
            if (wrcr & 0X0001)
            {
                wrcr >>= 1;
                wrcr ^= 0XA001;
            }
            else
            {
                wrcr >>= 1;
            }
        }
    }
    return wrcr;
}
```

1.4 数据类型介绍

a. 本产品数据传输采用大端模式。

例如传输的字节序：byte[0]=0x12, byte[1]=0x34, byte[2]=0x56, byte[3]=0x78.

则32位的数据为：0x12345678

b. float类型数据的传输采用32位单精度浮点数(IEEE754 Single precision 32-bit)。

例如传输的字节序为：byte[0]=0x41, byte[1]=0x28, byte[2]=0x00,byte[3]=0x00.

则表示的浮点数为：10.5f

用户如果采用C语言开发，可以参照下面的例程：

//用于转换的联合体

```
typedef union{
    float f_data;
    uint8_t u8_data[4];
}FLOAT_TYPE_U;

float get_float_data(uint8_t* byte)
{
    FLOAT_TYPE_U float_data = {0};
    //大端模式下使用下面的语句
    float_data.u8_data[0] = byte[0];
    float_data.u8_data[1] = byte[1];
    float_data.u8_data[2] = byte[2];
    float_data.u8_data[3] = byte[3];
    //小端模式下使用下面的语句
    float_data.u8_data[0] = byte[3];
    float_data.u8_data[1] = byte[2];
    float_data.u8_data[2] = byte[1];
    float_data.u8_data[3] = byte[0];
    return float_data.f_data;
}
```

2. 气体浓度单位表

值 (Hex)	0x02	0x04	0x08	0x10
浓度单位	ppm	ppb	%vol.	%LEL

3. 传感器类型表

传感器类型	HCHO	VOC	CO	Cl ₂	H ₂	H ₂ S	HCl	HCN	HF	NH ₃	NO ₂	O ₂	O ₃	SO ₂
值 (Hex)	0x17	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F	0x20	0x21	0x22	0x23	0x24

传感器类型	HBr	Br ₂	F ₂	PH ₃	AsH ₃	SiH ₄	GeH ₄	B ₂ H ₆	BF ₃	WF ₆	SiF ₄	XeF ₂	TiF ₄	SMELL
值 (Hex)	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F	0x30	0x31	0x32

传感器类型	IAQ	AQI	NMHC	SO _x	NO _x	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
值 (Hex)	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F	0x40

传感器类型	C ₃ H ₉ N	C ₂ H ₇ N	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	ClO ₂
值 (Hex)	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D

传感器类型	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	LEL	CO ₂	PID_VOCS	C ₂ H ₂
值 (Hex)	0x4E	0x4F	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58

4. 修改Modbus地址

建议修改范围: 01 (0x01)<=addr<=127 (0x7F), 默认为01 (0x01)。

发送:

指令头										地址
0x53	0x45	0x54	0x41	0x44	0x44	0x52	0x45	0x53	0x53	addr

返回:

指令头										地址
0x53	0x45	0x54	0x41	0x44	0x44	0x52	0x45	0x53	0x53	addr

5. 保持寄存器表, 以及读写示例

此协议只使用了Modbus-RTU中的保持寄存器, 读多个保持寄存器的操作码为0x03, 写单个保持寄存器操作码为0x06, 写多个保持寄存器的操作码为0x10

5.1 测量数据寄存器

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF000	Byte[0]	气体浓度值 (byte[0]<<24) (byte[1]<<16) (byte[2]<<8) byte[3] 然后按照32位float数据类型转换	float	R
	Byte[1]			
0xF001	Byte[2]			
	Byte[3]			
0xF002	Byte[4]	芯片内部温度(摄氏度) (byte[4]<<24) (byte[5]<<16) (byte[6]<<8) byte[7] 然后按照32位float数据类型转换	float	R
	Byte[5]			
0xF003	Byte[6]			
	Byte[7]			
0xF004	Byte[8]	保留 固定为0		R
	Byte[9]			
0xF005	Byte[11]			
	Byte[12]			
0xF006	Byte[13]	传感器状态 0: 正常 1: 建议更换传感器 2: 传感器失效	uint16	R/W
	Byte[14]			

读取示例:

发送: 0x01 0x03 0xF0 0x00 0x00 0x07 0x37 0x08

接收: 0x01 0x03 0x0E 0x00 0x00 0x00 0x00 0x41 0x86 0x8E 0xD9 0x00 0x00 0x00 0x00 0x00 0x37 0x43

发送解析:

0x01-----Modbus地址
0x03-----读保持寄存器
0xF0 0x00-----起始寄存器地址
0x00 0x07-----读取寄存器长度 - 7
0x37 0x08-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
0x03-----读保持寄存器
0x0E-----读取的字节长度 - 14
0x00 0x00 0x00 0x00-----浓度值 - 0x00000000 - 0.0f (IEEE754 single precision 32-bit)
0x41 0x86 0x8E 0xD9-----温度值 - 0x41868ED9 - 16.8197 (IEEE754 single precision 32-bit)
0x00 0x00 0x00 0x00-----保留 - 0x00000000
0x00 0x00-----传感器状态 - 正常工作
0x37 0x43-----Modbus-RTU CRC

5.2 传感器参数寄存器

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF100	Byte[0]	传感器类型, 请参照 【3.传感器类型表】	float	R
	Byte[1]			
0xF101	Byte[2]	浓度单位, 请参照 【2.气体浓度单位表】		
	Byte[3]			
0xF102	Byte[4]	传感器量程	float	R
	Byte[5]			
0xF103	Byte[6]	(byte[0]<<24) (byte[1]<<16) (byte[2]<<8) byte[3] 然后按照32位float数据类型转换		
	Byte[7]			

读取示例:

发送: 0x01 0x03 0xF1 0x00 0x00 0x04 0x76 0xF5

接收: 0x01 0x03 0x08 0x00 0x57 0x00 0x02 0x42 0xC8 0x00 0x00 0x5F 0x54

发送解析:

0x01-----Modbus地址
0x03-----读保持寄存器
0xF1 0x00-----起始寄存器地址
0x00 0x04-----读取寄存器长度
0x76 0xF5-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0x08-----读取的字节长度 - 8
 0x00 0x57-----传感器类型 - 通过查表【3.传感器类型表】- PID_VOCS
 0x00 0x02-----浓度单位 - 通过查表【2.气体浓度单位表】- ppm
 0x42 0xC8 0x00 0x00-----传感器量程 - 0x42C80000 - 100.0f
 0x5F 0x54-----Modbus-RTU CRC

5.3 设备序列号寄存器

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF200	Byte[0]	设备序列号 例如: "1234567890123456"	Ascii码	R
	Byte[1]			
0xF201	Byte[2]			
	Byte[3]			
0xF202	Byte[4]			
	Byte[5]			
0xF203	Byte[6]			
	Byte[7]			
0xF204	Byte[8]			
	Byte[9]			
0xF205	Byte[10]			
	Byte[11]			
0xF206	Byte[12]			
	Byte[13]			
0xF207	Byte[14]			
	Byte[15]			

读取示例:

发送: 0x01 0x03 0xF2 0x00 0x00 0x08 0x76 0xB4

接收: 0x01 0x03 0x10 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x30 0x31 0x32 0x33 0x34 0x35 0x36 0xC7 0xFB

发送解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0xF2 0x00-----起始寄存器地址
 0x00 0x08-----读取寄存器长度
 0x76 0xB4-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0x10-----读取的字节长度 – 16
 0x31 0x32 0x33 0x34-----“1234”
 0x35 0x36 0x37 0x38-----“5678”
 0x39 0x30 0x31 0x32-----“9012”
 0x33 0x34 0x35 0x36-----“3456”
 0xC7 0xFB-----Modbus-RTU CRC

5.4 传感器序列号寄存器

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF208	Byte[0]	传感器序列号 例如: “abcdefghijklmnop”	Ascii码	R
	Byte[1]			
0xF209	Byte[2]			
	Byte[3]			
0xF20A	Byte[4]			
	Byte[5]			
0xF20B	Byte[6]			
	Byte[7]			
0xF20C	Byte[8]			
	Byte[9]			
0xF20D	Byte[10]			
	Byte[11]			
0xF20E	Byte[12]			
	Byte[13]			
0xF20F	Byte[14]			
	Byte[15]			

读取示例:

发送: 0x01 0x03 0xF2 0x08 0x00 0x08 0xF7 0x76

接收: 0x01 0x03 0x10 0x61 0x62 0x63 0x64 0x65 0x66 0x67 0x68 0x69 0x6A 0x6B 0x6C 0x6D 0x6E 0x6F 0x70 0xD7 0xDF

发送解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0xF2 0x08-----起始寄存器地址
 0x00 0x08-----读取寄存器长度
 0xF7 0x76-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0x10-----读取的字节长度 – 16
 0x61 0x62 0x63 0x64-----“abcd”
 0x65 0x66 0x67 0x68-----“efgh”
 0x69 0x6A 0x6B 0x6C-----“ijkl”
 0x6D 0x6E 0x6F 0x70-----“mnop”
 0xD7 0xDF-----Modbus-RTU CRC

5.5 软件版本号寄存器

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF210	Byte[0]	软件版本号 例如: “DS4PID20231012_V0002”	Ascii码	R
	Byte[1]			
0xF211	Byte[2]			
	Byte[3]			
0xF212	Byte[4]			
	Byte[5]			
0xF213	Byte[6]			
	Byte[7]			
0xF214	Byte[8]			
	Byte[9]			
0xF215	Byte[10]			
	Byte[11]			
0xF216	Byte[12]			
	Byte[13]			
0xF217	Byte[14]			
	Byte[15]			
0xF218	Byte[16]			
	Byte[17]			
0xF219	Byte[18]			
	Byte[19]			

读取示例:

发送: 0x01 0x03 0xF2 0x10 0x00 0x0A 0xF6 0xB0

接收: 0x01 0x03 0x14 0x44 0x53 0x34 0x50 0x49 0x44 0x32 0x30 0x32 0x33 0x31 0x30 0x31 0x32 0x5F 0x56 0x30 0x30 0x30 0x32 0x39 0x46

发送解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0xF2 0x10-----起始寄存器地址
 0x00 0x0A-----读取寄存器长度
 0xF6 0xB0-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0x14-----读取的字节长度 - 20
 0x44 0x53 0x34 0x50-----“DS4P”
 0x49 0x44 0x32 0x30-----“ID20”
 0x32 0x33 0x31 0x30-----“2310”
 0x31 0x32 0x5F 0x56-----“12_V”
 0x30 0x30 0x30 0x32-----“0002”
 0x39 0x46-----Modbus-RTU CRC

5.6 厂家识别码寄存器

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF21A	Byte[0]	厂商识别码	用户自定义	R/W
	Byte[1]			
0xF21B	Byte[2]			
	Byte[3]			
0xF21C	Byte[4]			
	Byte[5]			
0xF21D	Byte[6]			
	Byte[7]			
0xF21E	Byte[8]			
	Byte[9]			
0xF21F	Byte[10]			
	Byte[11]			
0xF220	Byte[12]			
	Byte[13]			
0xF221	Byte[14]			
	Byte[15]			

写入示例:

发送: 0x01 0x10 0xF2 0x1A 0x00 0x08 0x10 0x45 0x43 0x2D 0x53 0x45 0x4E 0x53 0x45 0x2D 0x41 0x51 0x53 0x2D 0x44 0x45
 0x56 0xEC 0xDC

接收: 0x01 0x10 0xF2 0x1A 0x00 0x08 0xD2 0xB00x32 0x39 0x46

发送解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF2 0x1A-----起始寄存器地址
 0x00 0x08-----写入寄存器长度 - 8
 0x10-----写入字节长度 - 16
 0x45 0x43 0x2D 0x53-----写入数据
 0x45 0x4E 0x53 0x45-----写入数据
 0x2D 0x41 0x51 0x53-----写入数据
 0x2D 0x44 0x45 0x56-----写入数据
 0xEC 0xDC-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF2 0x1A-----起始寄存器地址
 0x00 0x08-----写入寄存器长度 - 8
 0xD2 0xB0-----Modbus-RTU CRC

读取示例:

发送: 0x01 0x03 0xF2 0x1A 0x00 0x08 0x57 0x73

接收: 0x01 0x03 0x10 0x45 0x43 0x2D 0x53 0x45 0x4E 0x53 0x45 0x2D 0x41 0x51 0x53 0x2D 0x44 0x45 0x56 0x8C 0x6D

发送解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0xF2 0x1A-----起始寄存器地址
 0x00 0x08-----读取寄存器长度
 0x57 0x73-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0x10-----读取的字节长度 - 16
 0x45 0x43 0x2D 0x53-----读取的数据
 0x45 0x4E 0x53 0x45-----读取的数据
 0x2D 0x41 0x51 0x53-----读取的数据
 0x2D 0x44 0x45 0x56-----读取的数据
 0x8C 0x6D-----Modbus-RTU CRC

5.7 标定寄存器

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF300	Byte[0]	用户标定浓度值，值必须大于等于0 (≥ 0) 建议使用两点标定法： 1. 先通入纯净空气，通过此寄存器写入0 2. 通入目标气体，通过此寄存器写入目标点浓度	float	W
	Byte[1]			
0xF301	Byte[2]			
	Byte[3]			
0xF302	Byte[4]	恢复出厂标定	uint16_t	W
	Byte[5]	1: 恢复出厂设置 其他值: 无效		

用户标定0点示例:

发送: 0x01 0x10 0xF3 0x00 0x00 0x02 0x04 0x00 0x00 0x00 0x00 0xE3 0x5B

接收: 0x01 0x10 0xF3 0x00 0x00 0x02 0x72 0x8C

发送解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF3 0x00-----起始寄存器地址
 0x00 0x02-----写入寄存器长度 - 2
 0x04-----写入字节长度 - 4
 0x00 0x00 0x00 0x00-----写入的浓度值 - 0x00000000 - 0.0f (IEEE754 single precision 32-bit)
 0xE3 0x5B-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF3 0x00-----起始寄存器地址
 0x00 0x02-----写入寄存器长度 - 2
 0x72 0x83-----Modbus-RTU CRC

用户标定目标点示例:

发送: 0x01 0x10 0xF3 0x00 0x00 0x02 0x04 0x41 0x20 0x00 0x00 0xF6 0xAD

接收: 0x01 0x10 0xF3 0x00 0x00 0x02 0x72 0x8C

发送解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF3 0x00-----起始寄存器地址
 0x00 0x02-----写入寄存器长度 - 2
 0x04-----写入字节长度 - 4
 0x00 0x00 0x00 0x00-----写入的浓度值 - 0x41200000 - 10.0f (IEEE754 single precision 32-bit)
 0xF6 0xAD-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF3 0x00-----起始寄存器地址
 0x00 0x02-----写入寄存器长度 - 2
 0x72 0x8C-----Modbus-RTU CRC

用户恢复出厂标定示例:

发送: 0x01 0x10 0xF3 0x02 0x00 0x01 0x02 0x00 0x01 0xA5 0x7D

接收: 0x01 0x10 0xF3 0x02 0x00 0x01 0x93 0x4D

发送解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF3 0x02-----起始寄存器地址
 0x00 0x01-----写入寄存器长度 - 1
 0x02-----写入字节长度 - 2
 0x00 0x01-----恢复出厂标定
 0xA5 0x7D-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x10-----写入多个保持寄存器
 0xF3 0x02-----起始寄存器地址
 0x00 0x01-----写入寄存器长度 - 1
 0x93 0x4D-----Modbus-RTU CRC

5.8 无抑制数据寄存器

可以选择性的提供给需要的用户使用。

寄存器地址 (Hex)	数据	描述	数据类型 (大端模式)	R/W
0xF400	Byte[0]	气体浓度值, 此浓度值有可能为负值 (byte[0]<<24) (byte[1]<<16) (byte[2]<<8) byte[3] 然后按照32位float数据类型转换	float	R
	Byte[1]			
0xF401	Byte[2]			
	Byte[3]			
0xF402	Byte[4]	芯片内部温度 (摄氏度) (byte[4]<<24) (byte[5]<<16) (byte[6]<<8) byte[7] 然后按照32位float数据类型转换	float	R
	Byte[5]			
0xF403	Byte[6]			
	Byte[7]			
0xF404	Byte[8]	保留 固定为0		R
	Byte[9]			
0xF405	Byte[11]			
	Byte[12]			
0xF406	Byte[13]	传感器状态 0: 正常 1: 建议更换传感器 2: 传感器失效	uint16	R/W
	Byte[14]			

读取示例:

发送: 0x01 0x03 0xF4 0x00 0x00 0x07 0x36 0x38

接收: 0x01 0x03 0x0E 0xBF 0xB0 0x5D 0xB0 0x41 0x97 0x56 0x87 0x00 0x00 0x00 0x00 0x00 0x3F 0xB3

发送解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0xF4 0x00-----起始寄存器地址
 0x00 0x07-----写入寄存器长度 - 7
 0x36 0x38-----Modbus-RTU CRC

接收解析:

0x01-----Modbus地址
 0x03-----读保持寄存器
 0x0E-----读取的字节长度 - 14
 0xBF 0xB0 0x5D 0xB0-----浓度值 - 0x00000000 - -1.377859f (IEEE754 single precision 32-bit)
 0x41 0x97 0x56 0x87-----温度值 - 0x41975687 - 18.917 (IEEE754 single precision 32-bit)
 0x41 0x97 0x56 0x87-----温度值 - 0x41975687 - 18.917 (IEEE754 single precision 32-bit)
 0x00 0x00 0x00 0x00-----保留 - 0x00000000
 0x00 0x00-----传感器状态 - 正常工作
 0x3F 0xB3-----Modbus-RTU CRC



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, www.ecnose.de

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 17 幢 4 层

邮编: 315100

座机: 0574-88097236, 88096372

邮箱: info@aqsystems.cn

网址: www.ecsense.cn, www.aqsystems.cn