

## Revision History

编著/修改人	版 本	日 期	备 注
廖朝高	V1.0	2016.9.7	初始版本
	V1.2.1	2018.8.20	
	V1.4.5	2019.3.15	
	V1.4.6	2019.3.27	增加 CMD_READ_SENSOR_ID  CMD_WRITE_SENSOR_ID
	V1.4.7		
	V1.4.8	2019.07.06	增加 <i>CMD_GET_MuICH_OFFSET = 0x2C,</i>  <i>CMD_SET_MuICH_OFFSET = 0x2D,</i>  <i>CMD_GET_PM25_OFFSET = 0x2E,</i>  <i>CMD_SET_PM25_OFFSET = 0x2F,</i>
	V1.4.9	2019.07.29	修正
	V1.5.0	2019.08.16	增加 <i>usr_path[128]</i>
	V1.5.1	2019.08.20	增加 <i>eWH57_SENSOR,</i>  <i>eWH55_SENSORCH1,</i>  <i>eWH55_SENSORCH2,</i>  <i>eWH55_SENSORCH3,</i>  <i>eWH55_SENSORCH4,</i>  <i>#define ITEM_LEAK_CH1 0x58//for Leak_ch1</i>

			<pre>#define ITEM_LEAK_CH2 0x59//for Leak_ch2  #define ITEM_LEAK_CH3 0x5A//for Leak_ch3  #define ITEM_LEAK_CH4 0x5B//for Leak_ch4  #define ITEM_THUNDERDISTANCE 0x60</pre>
	V1.5.2	2020.04.30	增加 ITEM_RAINEVENT 数据传出
	V1.5.3	2020.05.06	<pre>#define ITEM_TF_USR1 0x63//Temperature(℃) 3Byte  #define ITEM_TF_USR2 0x64//Temperature(℃) 3Byte  #define ITEM_TF_USR3 0x65//Temperature(℃) 3Byte  #define ITEM_TF_USR4 0x66//Temperature(℃) 3Byte  #define ITEM_TF_USR5 0x67//Temperature(℃) 3Byte  #define ITEM_TF_USR6 0x68//Temperature(℃) 3Byte  #define ITEM_TF_USR7 0x69//Temperature(℃) 3Byte  #define ITEM_TF_USR8 0x69//Temperature(℃) 3Byte  #define ITEM_TF_BATT 0x6B//tf temperature batt</pre>
	V1.5.4	2020.05.07	增加命令 CMD_READ_SENSOR_ID_NEW
	1.5.5	2020.06.12	<p>删除 ITEM_TF_BATT, 在读取 Live data 时在 ITEM_TF_USRch(ch = 1... 8)之后加一个字节电量数据。</p> <pre>#define ITEM_TF_USR1 0x63//Temperature(℃) 4Byte  #define ITEM_TF_USR2 0x64//Temperature(℃) 4Byte  #define ITEM_TF_USR3 0x65//Temperature(℃) 4Byte  #define ITEM_TF_USR4 0x66//Temperature(℃) 4Byte</pre>

			<b>#define ITEM_TF_USR5    0x67//Temperature(°C) 4Byte</b>
			<b>#define ITEM_TF_USR6    0x68//Temperature(°C) 4Byte</b>
			<b>#define ITEM_TF_USR7    0x69//Temperature(°C) 4Byte</b>
			<b>#define ITEM_TF_USR8    0x69//Temperature(°C) 4Byte</b>

## 1. 数据交换格式:

Fixed header, **CMD**, SIZE, DATA1, DATA2, ... , DATAn, CHECKSUM

Fixed header: 2 bytes, 包头, 固定为 0xffff

**CMD**: 1 byte, 命令

SIZE: 1 byte, 包的大小, 从 CMD 到 CHECKSUM

DATA: n bytes, 传输的数据, 可变长度

CHECKSUM: 1 byte, 校验和, CHECKSUM=CMD+SIZE+DATA1+DATA2+...+DATAn

## 2. 命令及数据结构定义:

```
typedef enum {
```

//通用命令

CMD\_WRITE\_SSID                      = 0x11, //发送路由器 SSID 和 Password 给 WIFI 模块的命令

CMD\_BROADCAST                        = 0x12, //局域网内广播查找设备命令, **注意: 返回的数据 size 是 2 Byte**

CMD\_READ\_ECOWITT                     = 0x1E, //读取 Ecowitt.net 网站设置命令

CMD\_WRITE\_ECOWITT                    = 0x1F, //改写 Ecowitt.net 网站设置命令

CMD\_READ\_WUNDERGROUND               = 0x20, //读取 Wunderground 网站设置命令

CMD\_WRITE\_WUNDERGROUND              = 0x21, //改写 Wunderground 网站设置命令

CMD_READ_WOW	= 0x22, //读取 WeatherObservationsWebsite 网站设置命令
CMD_WRITE_WOW	= 0x23, //改写 WeatherObservationsWebsite 网站设置命令
CMD_READ_WEATHERCLOUD	= 0x24, //读取 Weathercloud 网站设置命令
CMD_WRITE_WEATHERCLOUD	= 0x25, //改写 Weathercloud 网站设置命令
CMD_READ_SATION_MAC	= 0x26, //读取 MAC 地址
CMD_READ_CUSTOMIZED	= 0x2A, //读取 Customized sever 参数
CMD_WRITE_CUSTOMIZED	= 0x2B, //改写 Customized sever 参数
CMD_WRITE_UPDATE	= 0x43, //固件升级
CMD_READ_FIRMWARE_VERSION	= 0x50, //读取固件版本信息
CMD_READ_USR_PATH	= 0x51,
CMD_WRITE_USR_PATH	= 0x52,

//以下命令仅 GW1000 和 WH2650 支持:

CMD_GW1000_LIVEDATA	= 0x27, //读取当前数据, 注意: 返回的数据 size 是 2 Byte
CMD_GET_SOILHUMIAD	= 0x28, //读取 Soilmoisture Sensor 校准参数
CMD_SET_SOILHUMIAD	= 0x29, //改写 Soilmoisture Sensor 校准参数
CMD_GET_MuICH_OFFSET	= 0x2C, //读取多通道温湿度 OFFSET 参数
CMD_SET_MuICH_OFFSET	= 0x2D, //改写多通道温湿度 OFFSET 参数
CMD_GET_PM25_OFFSET	= 0x2E, //读取多通道 PM2.5OFFSET 校准参数
CMD_SET_PM25_OFFSET	= 0x2F, //改写多通道 PM2.5OFFSET 校准参数
CMD_READ_SSSS	= 0x30, //读取系统信息
CMD_WRITE_SSSS	= 0x31, //改写系统信息
CMD_READ_RAINDATA	= 0x34, //读取雨量数据

CMD_WRITE_RAINDATA	= 0x35, //改写雨量数据
CMD_READ_GAIN	= 0x36, //读取校准系数
CMD_WRITE_GAIN	= 0x37, //改写校准系数
CMD_READ_CALIBRATION	= 0x38, //读取 offset 校准值
CMD_WRITE_CALIBRATION	= 0x39, //改写 offset 校准值
CMD_READ_SENSOR_ID	= 0x3A, //读取 Sensors ID 参数
CMD_WRITE_SENSOR_ID	= 0x3B, //改写 Sensors ID 参数
CMD_READ_SENSOR_ID_NEW	= 0x3C, ///读取 Sensors ID 参数 为扩容也而增加
CMD_WRITE_REBOOT	= 0x40, //重启系统
CMD_WRITE_RESET	= 0x41, // 重启并恢复出厂设置

}CMD\_LT;

//\*\*\*\*\*

#define SOIL\_CH\_MAX 8

#define WH31\_CHANNEL 8

#define PM25\_CH\_MAX 4

#define LEAK\_CH\_MAX 4

typedef enum

{

//eWH24\_SENSOR = 0x00,

eWH65\_SENSOR = 0x00,

//eWH69\_SENSOR,

eWH68\_SENSOR,

eWH80\_SENSOR, //80H (发射间隔 15.5s)

eWH40\_SENSOR,

eWH25\_SENSOR,

eWH26\_SENSOR,

eWH31\_SENSORCH1,

eWH31\_SENSORCH2,

eWH31\_SENSORCH3,

eWH31\_SENSORCH4,

eWH31\_SENSORCH5,

eWH31\_SENSORCH6,

eWH31\_SENSORCH7,

eWH31\_SENSORCH8,

eWH51\_SENSORCH1,

eWH51\_SENSORCH2,

eWH51\_SENSORCH3,

eWH51\_SENSORCH4,

eWH51\_SENSORCH5,

eWH51\_SENSORCH6,

eWH51\_SENSORCH7,

eWH51\_SENSORCH8,

eWH41\_SENSORCH1,

eWH41\_SENSORCH2,

eWH41\_SENSORCH3,

eWH41\_SENSORCH4,

//-----

eWH57\_SENSOR,

eWH55\_SENSORCH1,

eWH55\_SENSORCH2,

eWH55\_SENSORCH3,

eWH55\_SENSORCH4,

eWH34\_SENSORCH1 = 31,

eWH34\_SENSORCH2 = 32,

eWH34\_SENSORCH3 = 33,

eWH34\_SENSORCH4 = 34,

eWH34\_SENSORCH5 = 35,

eWH34\_SENSORCH6 = 36,

eWH34\_SENSORCH7 = 37,

eWH34\_SENSORCH8 = 38,

//添加新 sensor 从此处开始添加，前面顺序不可改变

//-----

eMAX\_SENSOR

}SENSOR\_IDT;

//-----

#define ITEM\_INTEMP                      0x01//Indoor Temperature (°C)                      2

#define ITEM\_OUTTEMP                      0x02//Outdoor Temperature (°C)                      2

#define ITEM_DEWPOINT	0x03//Dew point (°C)	2
#define ITEM_WINDCHILL	0x04//Wind chill (°C)	2
#define ITEM_HEATINDEX	0x05//Heat index (°C)	2
#define ITEM_INHUMI	0x06//Indoor Humidity (%)	1
#define ITEM_OUTHUMI	0x07//Outdoor Humidity (%)	1
#define ITEM_ABSBARO	0x08//Absolutely Barometric (hpa)	2
#define ITEM_RELBARO	0x09//Relative Barometric (hpa)	2
#define ITEM_WINDDIRECTION	0x0A//Wind Direction (360°)	2
#define ITEM_WINDSPEED	0x0B//Wind Speed (m/s)	2
#define ITEM_GUSTSPEED	0x0C//Gust Speed (m/s)	2
#define ITEM_RAINEVENT	0x0D//Rain Event (mm)	2
#define ITEM_RAINRATE	0x0E//Rain Rate (mm/h)	2
#define ITEM_RAINHOUR	0x0F//Rain hour (mm)	2
#define ITEM_RAINDAY	0x10//Rain Day (mm)	2
#define ITEM_RAINWEEK	0x11//Rain Week (mm)	2
#define ITEM_RAINMONTH	0x12//Rain Month (mm)	4
#define ITEM_RAINYEAR	0x13//Rain Year (mm)	4
#define ITEM_RAINTOTALS	0x14//Rain Totals (mm)	4
#define ITEM_LIGHT	0x15//Light (lux)	4
#define ITEM_UV	0x16//UV (uW/m2)	2
#define ITEM_UVI	0x17//UVI (0-15 index)	1
#define ITEM_TIME	0x18//Date and time	6
#define ITEM_DAYLWINDMAX	0X19//Day max wind(m/s)	2



#define ITEM_TEMP1	0x1A//Temperature 1(°C)	2
#define ITEM_TEMP2	0x1B//Temperature 2(°C)	2
#define ITEM_TEMP3	0x1C//Temperature 3(°C)	2
#define ITEM_TEMP4	0x1D//Temperature 4(°C)	2
#define ITEM_TEMP5	0x1E//Temperature 5(°C)	2
#define ITEM_TEMP6	0x1F//Temperature 6(°C)	2
#define ITEM_TEMP7	0x20//Temperature 7(°C)	2
#define ITEM_TEMP8	0x21//Temperature 8(°C)	2
#define ITEM_HUMI1	0x22//Humidity 1, 0-100%	1
#define ITEM_HUMI2	0x23//Humidity 2, 0-100%	1
#define ITEM_HUMI3	0x24//Humidity 3, 0-100%	1
#define ITEM_HUMI4	0x25//Humidity 4, 0-100%	1
#define ITEM_HUMI5	0x26//Humidity 5, 0-100%	1
#define ITEM_HUMI6	0x27//Humidity 6, 0-100%	1
#define ITEM_HUMI7	0x28//Humidity 7, 0-100%	1
#define ITEM_HUMI8	0x29//Humidity 8, 0-100%	1
#define ITEM_PM25_CH1	0x2A//PM2.5 Air Quality Sensor(µg/m3)	2
#define ITEM_SOILTEMP1	0x2B//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE1	0x2C//Soil Moisture(%)	1
#define ITEM_SOILTEMP2	0x2D//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE2	0x2E//Soil Moisture(%)	1
#define ITEM_SOILTEMP3	0x2F//Soil Temperature(°C)	2

#define ITEM_SOILMOISTURE3	0x30//Soil Moisture(%)	1
#define ITEM_SOILTEMP4	0x31//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE4	0x32//Soil Moisture(%)	1
#define ITEM_SOILTEMP5	0x33//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE5	0x34//Soil Moisture(%)	1
#define ITEM_SOILTEMP6	0x35//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE6	0x36//Soil Moisture(%)	1
#define ITEM_SOILTEMP7	0x37//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE7	0x38//Soil Moisture(%)	1
#define ITEM_SOILTEMP8	0x39//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE8	0x3A//Soil Moisture(%)	1
#define ITEM_SOILTEMP9	0x3B//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE9	0x3C//Soil Moisture(%)	1
#define ITEM_SOILTEMP10	0x3D//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE10	0x3E//Soil Moisture(%)	1
#define ITEM_SOILTEMP11	0x3F//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE11	0x40//Soil Moisture(%)	1
#define ITEM_SOILTEMP12	0x41//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE12	0x42//Soil Moisture(%)	1
#define ITEM_SOILTEMP13	0x43//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE13	0x44//Soil Moisture(%)	1
#define ITEM_SOILTEMP14	0x45//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE14	0x46//Soil Moisture(%)	1

#define ITEM_SOILTEMP15	0x47//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE15	0x48//Soil Moisture(%)	1
#define ITEM_SOILTEMP16	0x49//Soil Temperature(°C)	2
#define ITEM_SOILMOISTURE16	0x4A//Soil Moisture(%)	1
#define ITEM_LOWBATT	0x4C//All sensor lowbatt 16 char	16
#define ITEM_PM25_24HAVG1	0x4D//for pm25_ch1	2
#define ITEM_PM25_24HAVG2	0x4E//for pm25_ch2	2
#define ITEM_PM25_24HAVG3	0x4F//for pm25_ch3	2
#define ITEM_PM25_24HAVG4	0x50//for pm25_ch4	2
#define ITEM_PM25_CH2	0x51//PM2.5 Air Quality Sensor(µg/m3)	2
#define ITEM_PM25_CH3	0x52//PM2.5 Air Quality Sensor(µg/m3)	2
#define ITEM_PM25_CH4	0x53//PM2.5 Air Quality Sensor(µg/m3)	2
#define ITEM_LEAK_CH1	0x58//for Leak_ch1	1
#define ITEM_LEAK_CH2	0x59//for Leak_ch2	1
#define ITEM_LEAK_CH3	0x5A//for Leak_ch3	1
#define ITEM_LEAK_CH4	0x5B//for Leak_ch4	1
#define ITEM_LIGHTNING	0x60//雷电距离 (1~40KM)	1
#define ITEM_LIGHTNING_TIME	0x61//雷电检测到的时间(UTC)	4
#define ITEM_LIGHTNING_POWER	0x62//当天雷电次数	4

```
#define ITEM_TF_USR1      0x63//Temperature(℃)      2
#define ITEM_TF_USR2      0x64//Temperature(℃)      2
#define ITEM_TF_USR3      0x65//Temperature(℃)      2
#define ITEM_TF_USR4      0x66//Temperature(℃)      2
#define ITEM_TF_USR5      0x67//Temperature(℃)      2
#define ITEM_TF_USR6      0x68//Temperature(℃)      2
#define ITEM_TF_USR7      0x69//Temperature(℃)      2
#define ITEM_TF_USR8      0x6A//Temperature(℃)      2
#define ITEM_TF_BATT      0x6B//tf temperature batt  8
```

//-----

#if 1

typedef union //1 低电 0 正常

{

unsigned char batt;

struct

{

unsigned char wh41 : 4; /\* bit 0~3 \*/ // 0~5

unsigned char wh40 : 1; /\* bit 4 \*/

unsigned char wh26 : 1; /\* bit 5 \*/

unsigned char wh25 : 1; /\* bit 6 \*/

unsigned char wh24 : 1; /\* bit 7 \*/ //65, 69

} bits;

```
} _sig_sen;
```

```
typedef union //1 低电 0 正常
```

```
{
```

```
    unsigned char batt;
```

```
    Struct {
```

```
        unsigned char  ch1  : 1;    /* bit 0 */
```

```
        unsigned char  ch2  : 1;    /* bit 1 */
```

```
        unsigned char  ch3  : 1;    /* bit 2 */
```

```
        unsigned char  ch4  : 1;    /* bit 3 */
```

```
        unsigned char  ch5  : 1;    /* bit 4 */
```

```
        unsigned char  ch6  : 1;    /* bit 5 */
```

```
        unsigned char  ch7  : 1;    /* bit 6 */
```

```
        unsigned char  ch8  : 1; /* bit 7 */
```

```
    } bits;
```

```
} _wh31_ch;
```

```
typedef union //val
```

```
{
```

```
    unsigned short batt;
```

```
    struct {
```

```
        unsigned char ch1  : 4; /* bit 0~3 */ // 0~5
```

```
        unsigned char ch2  : 4; /* bit 4~7 */ // 0~5
```

```
        unsigned char ch3    : 4; /* bit 8~11 */ // 0~5

        unsigned char ch4    : 4; /* bit 12~15 */ // 0~5

    } bits;

} _wh41_ch;

typedef union //1 低电 0 正常
{

    unsigned short  batt;

    struct    {

        unsigned char  ch1  : 1; /* bit 0 */

        unsigned char  ch2  : 1;    /* bit 1 */

        unsigned char  ch3  : 1;    /* bit 2 */

        unsigned char  ch4  : 1;    /* bit 3 */

        unsigned char  ch5  : 1;    /* bit 4 */

        unsigned char  ch6  : 1;    /* bit 5 */

        unsigned char  ch7  : 1;    /* bit 6 */

        unsigned char  ch8  : 1; /* bit 7 */

        unsigned char  ch9   : 1; /* bit 8 */

        unsigned char  ch10  : 1;    /* bit 9 */

        unsigned char  ch11  : 1;    /* bit 10 */

        unsigned char  ch12  : 1;    /* bit 11 */

        unsigned char  ch13  : 1;    /* bit 12 */
```

```
        unsigned char    ch14    : 1;    /* bit 13 */

        unsigned char    ch15    : 1;    /* bit 14 */

        unsigned char    ch16    : 1;    /* bit 15 */

    } word;

} _wh51_ch;

// 表示发射传感的电压值与低电

typedef union _sensor_batt

{

    unsigned char all_batt[24]; // 16+8

    //以电压值表示

    struct

    {

        _sig_sen single;

        _wh31_ch wh31;

        _wh51_ch wh51;

        unsigned char wh57; // 0~5

        unsigned char wh68; // 0.02V * val(received val) = wh68(当前电压值);

        unsigned char wh80; // 0.02V * val(received val) = wh80(当前电压值);

        unsigned char uns2; // no use

        _wh41_ch wh41;

        unsigned char wh55[LEAK_CH_MAX];

        //V1.5.9 开始加入的

        unsigned char wh34[TF_CH_MAX]; // TF_CH_MAX = 8
```

```
} val;  
  
} sensor_batt;// 表示发射传感的电压值与低电  
  
#endif  
  
//-----
```

### 3. 配网及局域网内查找设备

配网模式 1：手机 app 以 49123 为端口，建产 TCP Server。 WIFI 模块在 station+AP 模式建立 TCP Client 去连接手机 App;当 WIFI 成功连接到了手机 App 的 TCP Server 时，App 的 TCP Server 发送 CMD\_WRITE\_SSID 命令！

配网模式 2：WIFI 模块在 station+AP 模式以 45000 为端口建立 TCP Server，等待手机 App 连接来连接。当手机 App 成功连接到 WIFI 模块的 TCP Server 后，可以发送 CMD\_WRITE\_SSID 命令！

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_SSID	1	0x11
Size	1	包长度
SSID Size	1	SSID 长度
SSID	n	Max 32
Password Size	1	Password 长度
Password	n	Max 64
Checksum	1	Checksum

WIFI 模块返回数据：

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_SSID	1	0x11



Size	1	包长度
Result	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

当 WIFI 和 APP 都连接到同一个路由器时, APP 用广播 (UDP) 的方式发送这个命令, 模块收到这个广播后回复 WIFI 模块的 MAC、IP、PORT 以及模块 AP 的 SSID 给 APP。(目的端口 46000)

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_BROADCAST	1	0x12
Size	1	包长度
Checksum	1	Checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_BROADCAST	1	0x12
Size	2	包长度(此处返回的数据 size 是 2 Byte)
MAC	6	模块 STA MAC
IP1	1	Eg. 192.168.100.1 中的 192
IP2	1	Eg. 192.168.100.1 中的 168
IP3	1	Eg. 192.168.100.1 中的 100
IP4	1	Eg. 192.168.100.1 中的 1
Port1	1	Eg. 0x1194(45000)中的 0x11

Port2	1	Eg. 0x1194(45000)中的 0x94
AP SSID	n	模块 AP 的 SSID
Checksum	1	checksum

WIFI 模块在以 45000 为端口建立 TCP Server，等待手机 App 连接来连接。当手机 App 成功连接到 WIFI 模块的 TCP Server 后，可以用下面的命令进行通讯：

1) 读取 Ecowitt.net 网站设置命令

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_ECOWITT	1	0x1E
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据：

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_ECOWITT	1	0x1E
Size	1	包长度
Uplaud interval	1	0~5min (0: mean is OFF)
Checksum	1	checksum

2) 改写 Ecowitt.net 网站设置命令

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff

CMD_WRITE_ECOWITT	1	0x1F
Size	1	包长度
Uplaod interval	1	0~5min (0: mean is OFF)
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_ECOWITT	1	0x1F
Size	1	包长度
Result	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

### 3) 读取 Wunderground 网站设置命令

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_WUNDERGROUND	1	0x20
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_WUNDERGROUND	1	0x20

Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 32
Password Size	1	Password Size
Password	n	ASCII 码, max 32
Fix	1	1
Checksum	1	checksum

#### 4) 改写 Wunderground 网站设置命令

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_WUNDERGROUND	1	0x21
Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 32
Password Size	1	Password Size
Password	n	ASCII 码, max 32
Fix	1	1
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff

CMD_WRITE_WUNDERGROUND	1	0x21
Size	1	包长度
Result	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

5) 读取 WeatherObservationsWebsite 网站设置命令

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_WOW	1	0x22
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_WOW	1	0x22
Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 39
Password Size	1	Password Size
Password	n	ASCII 码, max 32
StationNum Size (unused)	1	StationNum size (unused)
StationNum (unused)	n	ASCII 码, max 32 (unused)

Fix	1	1
Checksum	1	checksum

6) 改写 WeatherObservationsWebsite 网站设置命令

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_WOW	1	0x23
Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 39
Password Size	1	Password Size
Password	n	ASCII 码, max 32
StationNum Size (unused)	1	StationNum size (unused)
StationNum (unused)	32	ASCII 码, max 32 (unused)
Fix	1	1
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_WOW	1	0x23
Size	1	包长度
Result	1	0x00:设置成功, 0x01: 设置失败

Checksum	1	checksum
----------	---	----------

7) 读取 Weathercloud 网站设置命令

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_WEATHERCLOUD	1	0x24
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_WEATHERCLOUD	1	0x24
Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 32
Key Size	1	Key Size
Key	n	ASCII 码, max 32
Fix	1	1
Checksum	1	checksum

8) 改写 Weathercloud 网站设置命令。

字 段	长度（字节）	说 明
-----	--------	-----

Fixed header	2	固定 0xffff
CMD_WRITE_WEATHERCLOUD	1	0x25
Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 32
Key Size	1	Key Size
Key	n	ASCII 码, max 32
Fix	1	1
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_WEATHERCLOUD	1	0x25
Size	1	包长度
Result	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

#### 9) 读取 customer server 参数

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_CUSTOMIZED	1	0x2A
Size	1	包长度



Checksum	1	checksum
----------	---	----------

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_CUSTOMIZED	1	0x2A
Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 40
Password Size	1	Password Size
Password	n	ASCII 码, max 40
Server Size	1	Server Size
Server	n	ASCII 码, max 64
Port	2	0-65535
Interval	2	16-600
Type	1	0:EC 1WU
Active	1	0:Disable 1:Enable
Checksum	1	checksum

#### 10) 改写 customer server 参数

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_CUSTOMIZED	1	0x2B

Size	1	包长度
ID Size	1	ID Size
ID	n	ASCII 码, max 40
Password Size	1	Password Size
Password	n	ASCII 码, max 40
Server Size	1	Server Size
Server	n	ASCII 码, max 64
Port	2	0-65535
Interval	2	16-600
Type	1	0:EC 1WU
Active	1	0:Disable 1:Enable
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_CUSTOMIZED	1	0x2B
Size	1	包长度
Result	1	0x00:成功, 0x01: 失败
Checksum	1	checksum

#### 11) 读取 customer usr\_path 参数

字 段	长度 (字节)	说 明
-----	---------	-----

Fixed header	2	固定 0xffff
CMD_READ_USRPATH	1	0x51
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_USRPATH	1	0x51
Size	1	包长度
Ecowitt Path	64	ASCII 码, max 64
WU Path	64	ASCII 码, max 64
Checksum	1	checksum

12) 改写 customer usr\_path 参数

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_USRPATH	1	0x52
Size	1	包长度
Ecowitt Path	64	ASCII 码, max 64
WU Path	64	ASCII 码, max 64
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_USRPATH	1	0x52
Size	1	包长度
Result	1	0x00:成功, 0x01: 失败
Checksum	1	checksum

### 13) 读取 Soilmoisture Sensor 校准参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_GET_SOILHUMIAD	1	0x28
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_GET_SOILHUMIAD	1	0x29
Size	1	包长度
Channel	1	通道号
Current humidity	1	由发射机发出
Current ad	2	由发射机发出
Humidity ad select	1	选项开关

Min ad	1	调校的最小 ad(70~200)
Max ad	2	调校的最大 ad(80~1000)
...	...	
Checksum	1	checksum

#### 14) 改写 Soilmoisture Sensor 校准参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_SET_SOILHUMIAD	1	0x29
Size	1	包长度
Channel	1	通道号
Humidity ad select	1	选项开关
Min ad	1	调校的最小 ad(70~200)
Max ad	2	调校的最大 ad(80~1000)
...	...	
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_SET_SOILHUMIAD	1	0x29
Size	1	包长度
Result	1	0x00:成功, 0x01: 失败

Checksum	1	checksum
----------	---	----------

15) 读取多通道温湿度 OFFSET 参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_GET_MuICH_OFFSET	1	0x2C
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_GET_MuICH_OFFSET	1	0x2C
Size	1	包长度
Channel	1	通道号
humidity offset	1	Range: -10 ~ 10, default: 0
Temperature offset	1	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10
.....	....	
WH31_CHANNEL-1	1	0~7
humidity offset	1	
Temperature offset	1	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10

Checksum	1	checksum
----------	---	----------

16) 改写多通道温湿度 OFFSET 参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_SET_MulCH_OFFSET	1	0x2D
Size	1	包长度
Channel	1	通道号
humidity offset	1	Range: -10 ~ 10, default: 0
Temperature offset	1	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10
.....	....	
WH31_CHANNEL-1	1	0~7
humidity offset	1	
Temperature offset	1	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_SET_MulCH_OFFSET	1	
Size	1	包长度

Result	1	0x00:成功, 0x01: 失败
Checksum	1	checksum

17) 读取多通道 PM2.5OFFSET 校准参数

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_GET_PM25_OFFSET	1	0x2E
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_GET_PM25_OFFSET	1	0x2E
Size	1	包长度
Channel	1	通道号
PM25offset	2	Range: -200 ~ 200, default: 0 Note: (-20~20 ug/m3)x10
.....	....	
PM25_CH_MAX-1	1	0~3
PM25offset	1	
Checksum	1	checksum

18) 改写多通道 PM2.5OFFSET 校准参数



字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_SET_PM25_OFFSET	1	0x2F
Size	1	包长度
Channel	1	通道号
PM25offset	2	Range: -200 ~ 200, default: 0 Note: (-20~20 ug/m3)x10
.....	....	
PM25_CH_MAX-1	1	0~3
PM25offset	1	
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_SET_PM25_OFFSET	1	0x2F
Size	1	包长度
Result	1	0x00:成功, 0x01: 失败
Checksum	1	checksum

19) 读取 MAC 地址

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff

CMD_READ_SATION_MAC	1	0x26
Size	1	包长度
Result	1	0x00:成功, 0x01: 失败
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_SATION_MAC	1	0x26
Size	1	包长度
Sta_mac[6]	6	sta_mac[0];sta_mac[1];sta_mac[2]; sta_mac[3];sta_mac[4];sta_mac[5];
Checksum	1	checksum

20) 读取当前数据, 注意: 返回的数据 size 是 2 Byte

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_GW1000_LIVEDATA	1	0x27
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff

CMD_GW1000_LIVEDATA	1	0x27
Size	2	包长度(此处返回的数据 size 是 2 Byte)
ITEM_PM25	1	
Value	2	Unsigned short (valuex10)
ITEM_PM10	1	
Value	2	Unsigned short (valuex10)
ITEM_CH1_SOil_H	1	
Value	1	0~99
ITEM_CH2_TEMP	1	
Value	2	signed short (valuex10)
... ..		
ITEM_CH7_TEMP	1	
Value	2	
ITEM_CH1_HUMI	1	
Value	1	0~99
ITEM_CH2_HUMI	1	
... ..		
ITEM_CH7_HUMI	1	
Value	1	
ITEM_TF_USR1	1	
Temperature Value	2	signed short (valuex10)
Battery Value	1	0.02V * val

.....	.....	
ITEM_TF_USR8	1	
Temperature Value	2	signed short (valuex10)
Battery Value	1	0.02V * val
.....	.....	
Checksum	1	checksum

## 21) 读取系统参数

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_SSSS	1	0x30
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_SSSS	1	0x30
Size	1	包长度
Frequency	1	Wireless Receive Frequency(Only read)
Sensor Type	1	0:WH24 1:WH65
UTC TIME	4	Unsigned long(Only read)

Timezone	1	Local time zone
Checksum	1	checksum

## 22) 改写系统参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_SSSS	1	0x31
Size	1	包长度
Frequency	1	(Only read)Can't be rewritten.
Sensor Type	1	0:WH24 1:WH65
UTC TIME	4	Unsigned long(Only read)
Timezone	1	Local time zone
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_SSSS	1	0x31
Size	1	包长度
Result	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

## 23) 读取雨量数据

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_RAINDATA	1	0x34
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_RAINDATA	1	0x34
Size	1	包长度
RainRate	4	Range: 0~60000 Note: (0mm ~6000.0mm)x10
RainDay	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
RainWeek	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
RainMonth	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
RainYear	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
Checksum	1	

改写雨量数据

*FOS*

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_RAINDATA	1	0x35
Size	1	包长度
RainDay	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
RainWeek	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
RainMonth	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
RainYear	4	Range: 0~99999 Note: (0mm ~9999.9mm)x10
Checksum	1	

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_RAINDATA	1	0x35
Size	1	包长度
Resulte	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

24) 读取校准系数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_GAIN	1	0x36
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_GAIN	1	0x36
Size	1	包长度
Fixed	2	1267
uvGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
solarRadGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
windGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
rainGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
Reserved	2	Reserved
Checksum	1	

改写校准系数

*FOS*



字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_GAIN	1	0x37
Size	1	包长度
Reserved	2	1267(x 10)
uvGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
solarRadGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
windGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
rainGain	2	Range: 10~500, default: 100 Note: (0.10 ~5.00)x100
Reserved	2	Reserved
Checksum	1	

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_GAIN	1	0x37
Size	1	包长度
Resulte	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

25) 读取 offset 校准值

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_CALIBRATION	1	0x38
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_CALIBRATION	1	0x38
Size	1	包长度
inTempOffset	2	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10
inHumiOffset	1	Range: -10~10, default: 0
AbsOffset	4	Range: -800~800, default: 0 Note: (-80.0hpa~80.0hpa)x10
RelOffset	4	Range: -800~800, default: 0 Note: (-80.0hpa~80.0hpa)x10
outTempOffset	2	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10
outHumiOffset	1	Range: -10~10, default: 0
windDirOffset	2	Range: -180~180, default: 0

Checksum	1	
----------	---	--

## 26) 改写 offset 校准值

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_CALIBRATION	1	0x39
Size	1	包长度
inTempOffset	2	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10
inHumiOffset	1	Range: -10~10, default: 0
AbsOffset	4	Range: -800~800, default: 0 Note: (-80.0hpa~80.0hpa)x10
RelOffset	4	Range: -800~800, default: 0 Note: (-80.0hpa~80.0hpa)x10
outTempOffset	2	Range: -100~100, default: 0 Note: (-10.0℃~10.0℃)x10
outHumiOffset	1	Range: -10~10, default: 0
windDirOffset	2	Range: -180~180, default: 0
Checksum	1	

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff

CMD_WRITE_CALIBRATION	1	0x39
Size	1	包长度
Resulte	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

## 27) 读取 Sensors ID 参数

typedef enum

{

//eWH24\_SENSOR = 0x00,

eWH65\_SENSOR = 0x00, //电压标志位, 1 表示低电压, 0 正常

//eWH69\_SENSOR,

eWH68\_SENSOR, //电压=val\*0.02V 当<=1.2V 时显示低电压

eWH80\_SENSOR,

eWH40\_SENSOR, //电压标志位, 1 表示低电压, 0 正常

eWH25\_SENSOR, //电压标志位, 1 表示低电压, 0 正常

eWH26\_SENSOR, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH1, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH2, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH3, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH4, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH5, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH6, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH7, //电压标志位, 1 表示低电压, 0 正常

eWH31\_SENSORCH8, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH1, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH2, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH3, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH4, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH5, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH6, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH7, //电压标志位, 1 表示低电压, 0 正常

eWH51\_SENSORCH8, //电压标志位, 1 表示低电压, 0 正常

eWH41\_SENSORCH1, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH41\_SENSORCH2, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH41\_SENSORCH3, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH41\_SENSORCH4, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

//-----

eWH57\_SENSOR, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH55\_SENSORCH1, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH55\_SENSORCH2, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH55\_SENSORCH3, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH55\_SENSORCH4, //电量等级 0~5, 当电量等级 $\leq 1$  时显示低电压。

eWH34\_SENSORCH1 = 31, //电压= $\text{val} \times 0.02\text{V}$  当 $\leq 1.2\text{V}$  时显示低电压

eWH34\_SENSORCH2 = 32, //电压= $\text{val} \times 0.02\text{V}$  当 $\leq 1.2\text{V}$  时显示低电压

eWH34\_SENSORCH3 = 33, //电压= $\text{val} \times 0.02\text{V}$  当 $\leq 1.2\text{V}$  时显示低电压

eWH34\_SENSORCH4 = 34, //电压=val\*0.02V 当<=1.2V 时显示低电压

eWH34\_SENSORCH5 = 35, //电压=val\*0.02V 当<=1.2V 时显示低电压

eWH34\_SENSORCH6 = 36, //电压=val\*0.02V 当<=1.2V 时显示低电压

eWH34\_SENSORCH7 = 37, //电压=val\*0.02V 当<=1.2V 时显示低电压

eWH34\_SENSORCH8 = 38, //电压=val\*0.02V 当<=1.2V 时显示低电压

//添加新 sensor 从此处开始添加，前面顺序不可改变

//-----

eMAX\_SENSOR

}SENSOR\_IDT;

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_SENSOR_ID	1	0x3A
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_SENSOR_ID	1	0x3A
Size	1	包长度
WH65_SENSOR	1	0x01
WH65_ID	4	unsigned long

Wh65_signal	1	0~4
battery	1	
WH68_SENSOR	1	0x02
WH68_ID	4	unsigned long
WH68_signal	1	0~4
battery	1	
...SENSOR	1	..
..._ID	4	...
..._signal	1	0~4
battery		
Checksum	1	

## 28) 读取 Sensors 相关参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_SENSOR_ID_NEW	1	0x3C
Size	1	包长度
Checksum	1	checksum

WIFI 模块返回数据:

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_SENSOR_ID_NEW	1	0x3C

Size	2	包长度
WH65_SENSOR	1	0x01
WH65_ID	4	unsigned long
Wh65_signal	1	0~4
battery	1	
WH68_SENSOR	1	0x02
WH68_ID	4	unsigned long
WH68_signal	1	0~4
battery	1	
...SENSOR	1	..
..._ID	4	...
..._signal	1	0~4
battery		
Checksum	1	

## 29) 改写 Sensors ID 参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_SENSOR_ID:	1	0x3B
Size	1	包长度



WH65_SENSOR	1	0x01
WH65_ID	4	Unsigned long
WH68_SENSOR	1	0x02
WH68_ID	4	Unsigned long
...SENSOR	1	SENSOR_IDT
..._ID	4	Unsigned long
Checksum	1	

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_SENSOR_ID	1	0x3B
Size	1	包长度
Resulte	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

备注: 当写入的 ID = 0xFFFFFFFF 时表示重新注册对应发射机的 ID, 当写入的 ID = 0xFFFFFFFFE 时表示 Disable 此发射机。

### 30) 读取固件版本信息

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_FIRMWARE_VERSION	1	0x50
Size	1	包长度
Checksum	1	

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_READ_FIRMWARE_VERSION	1	0x50
Size	1	包长度
Versoin length	1	最大长版本长度为 23Bytes
Version buffer		For example: "EasyWeatherV1.2.0"
Checksum	1	checksum

### 31) 固件升级

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_UPDATE	1	0x43
Size	1	包长度
ServerIP	4	0xc0a80063 //"192.168.0.99"
ServerPort	2	1~65535
Checksum	1	

WIFI 模块返回数据:

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_UPDATE	1	0x43
Size	1	包长度

Resulte	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum

用户点击 Update firmware , app 把服务器 IP 和端口号发送给模块, 模块与服务器建立连接:

connect success

sent "user1.bin" or "user2.bin"

sent firmware data size (eg: 33334566 bytes)

sent "start"

sent packet(1) //(packet size 1460bytes)

sent "continue"

.....

.....

sent packet(n)

sent "continue"

sent packet (n+1)

sent "end"

1.client 连接到服务器后给 server 发 filename, server 回应 file length;

2.client 收到 file length 给 server 发 start, server 回应第 1 包 firmware data

3.client 收到第 1 包 firmware data 后 给 server 发 continue, server 回应第 2 包 firmware data

.....

4.client 收到第 n-1 包 firmware data 后 给 server 发 continue, server 回应第 n 包 firmware data

5.client 收到第 n 包 firmware data 后, 如果成功就给 server 发 end.

32) 重启设备。

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_REBOOT	1	0x40
Size	1	包长度
Checksum	1	

WIFI 模块返回数据：

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_REBOOT	1	0x40
Size	1	包长度
Resulte	1	0x00:设置成功，0x01: 设置失败
Checksum	1	checksum

33) 重置设备，恢复出厂。

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_RESET	1	0x41
Size	1	包长度
Checksum	1	

WIFI 模块返回数据：

字 段	长度（字节）	说 明
-----	--------	-----

Fixed header	2	固定 0xffff
CMD_WRITE_RESET	1	0x41
Size	1	包长度
Resulte	1	0x00:设置成功, 0x01: 设置失败
Checksum	1	checksum