

GW1000/GW1100 通信协议书

历史版本修改记录

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 CMD_GET_MuICH_OFFSET = 0x2C, CMD_SET_MuICH_OFFSET = 0x2D, CMD_GET_PM25_OFFSET = 0x2E, CMD_SET_PM25_OFFSET = 0x2F,
	V1.4.9	2019.07.29	修正
	V1.5.0	2019.08.16	增加 usr_path[128]
	V1.5.1	2019.08.20	增加 eWH57_SENSOR, eWH55_SENSORCH1, eWH55_SENSORCH2, eWH55_SENSORCH3, eWH55_SENSORCH4, #define ITEM_LEAK_CH1 0x58//for Leak_ch1 #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
	V1.5.2	2020.04.30	增加 ITEM_RAINEVENT 数据传出
	V1.5.3	2020.05.06	#define ITEM_TF_USR1 0x63//Temperature(°C) 3Bytes #define ITEM_TF_USR2 0x64//Temperature(°C) 3Bytes #define ITEM_TF_USR3 0x65//Temperature(°C) 3Bytes #define ITEM_TF_USR4 0x66//Temperature(°C) 3Bytes #define ITEM_TF_USR5 0x67//Temperature(°C) 3Bytes #define ITEM_TF_USR6 0x68//Temperature(°C) 3Bytes #define ITEM_TF_USR7 0x69//Temperature(°C) 3Bytes #define ITEM_TF_USR8 0x6A//Temperature(°C) 3Bytes #define ITEM_TF_BATT 0x6B//tf temperature batt
	V1.5.4	2020.05.07	增加命令 CMD_READ_SENSOR_ID_NEW
	V1.5.5	2020.06.12	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。 #define ITEM_TF_USR1 0x63//Temperature(°C) 3Bytes #define ITEM_TF_USR2 0x64//Temperature(°C) 3Bytes #define ITEM_TF_USR3 0x65//Temperature(°C) 3Bytes #define ITEM_TF_USR4 0x66//Temperature(°C) 3Bytes #define ITEM_TF_USR5 0x67//Temperature(°C) 3Bytes #define ITEM_TF_USR6 0x68//Temperature(°C) 3Bytes #define ITEM_TF_USR7 0x69//Temperature(°C) 3Bytes #define ITEM_TF_USR8 0x6A//Temperature(°C) 3Bytes

	V1.5.6	2020.06.22	修正这两个命令: CMD_READ_SSSS = 0x30, //读取系统信息 CMD_WRITE_SSSS = 0x31, //改写系统信息
	V1.5.7	2020.07.16	修正 CMD_READ_SENSOR_ID CMD_READ_SENSOR_ID_NEW
	V1.5.8	2020.07.20	增加 #define ITEM_SENSOR_CO2 0x70
	V1.5.9	2020.08.13	增加: CMD_GET_CO2_OFFSET = 0x53, CMD_SET_CO2_OFFSET = 0x54,
	V1.6.0	2021.01.05	版本: GW1000_Firmware V1.6.5 {leaf wetness sensor} 增加 ITEM ("ITEM_LEAF_WETNESS_CHx"(x=1~8)) #define ITEM_LEAF_WETNESS_CH1 0x72 #define ITEM_LEAF_WETNESS_CH2 0x73 #define ITEM_LEAF_WETNESS_CH3 0x74 #define ITEM_LEAF_WETNESS_CH4 0x75 #define ITEM_LEAF_WETNESS_CH5 0x76 #define ITEM_LEAF_WETNESS_CH6 0x77 #define ITEM_LEAF_WETNESS_CH7 0x78 #define ITEM_LEAF_WETNESS_CH8 0x79
	V1.6.1	2021.08.17	修正 CMD_READ_SENSOR_ID 和 CMD_READ_SENSOR_ID_NEW 命令中 sensor 的电量状态
	V1.6.2	2021.11.4	增加 WS90 的 sensor id. CMD_READ_RSTRAIN_TIME = 0x55, CMD_WRITE_RSTRAIN_TIME = 0x56,
	V1.6.3	2021.11.4	修正, WN34 的多通道, 数据 ITEM_TF_USR(1~8). 有效长度为 3 bytes;
	V1.6.4	2022.03.05	CMD_READ_RAIN = 0x57, CMD_WRITE_RAIN = 0x58,
	V1.6.5	2022.04.19	修改 ITEM_RAINHOUR 为 ITEM_RAIN_Gain (The is Rain Gain). ITEM_Piezo_Hourly_Rain (Reserved, not used.)
	V1.6.6	2022.04.20	增加 ITEM_RAIN_Priority 0x7A // 1 rain priority
	V1.6.7	2022.06.17	修正
	V1.6.8	2023.02.27	为 CMD_READ_RAIN, CMD_WRITE_RAIN 的命令保持一致, 加入 ITEM_radcompensation

	V1.6.9	2024.01.15	增加 ITEM_HEAP_FREE 到 CMD_GW1000_LIVEDATA 命令中

目录

GW1000 通信协议书.....	1
历史版本修改记录.....	1
1. 数据交换格式:	5
2. 命令及数据结构定义:	5
3. 配网及局域网内查找设备.....	13
1) 读取 Ecowitt.net 网站设置命令.....	14
2) 改写 Ecowitt.net 网站设置命令.....	15
3) 读取 Wunderground 网站设置命令.....	15
4) 改写 Wunderground 网站设置命令.....	15
5) 读取 WeatherObservationsWebsite 网站设置命令.....	16
6) 改写 WeatherObservationsWebsite 网站设置命令.....	16
7) 读取 Weathercloud 网站设置命令.....	17
8) 改写 Weathercloud 网站设置命令。.....	17
9) 读取 customer server 参数.....	18
10) 改写 customer server 参数.....	18
11) 读取 customer usr_path 参数.....	19
12) 改写 customer usr_path 参数.....	19
13) 读取 Soilmoisture Sensor 校准参数.....	19
14) 改写 Soilmoisture Sensor 校准参数.....	20
15) 读取多通道温湿度 OFFSET 参数.....	20
16) 改写多通道温湿度 OFFSET 参数.....	21
17) 读取多通道 PM2.5OFFSET 校准参数.....	21
18) 改写多通道 PM2.5OFFSET 校准参数.....	22
19) 读取 CO2 OFFSET 校准参数.....	22
20) 改写 CO2 OFFSET 校准参数.....	23
21) 读取 Rain 相关的参数.....	23
22) 改写 Rain 相关的参数.....	24
23) 读取 MAC 地址.....	25

24) 读取当前数据，注意：返回的数据 size 是 2 Byte.....	26
25) 读取系统参数.....	27
26) 改写系统参数.....	27
27) 读取雨量数据.....	28
28) 读取校准系数.....	29
29) 读取 offset 校准值.....	30
30) 改写 offset 校准值.....	31
31) 读取 Sensors ID 参数.....	31
32) 读取 Sensors 相关参数.....	33
33) 改写 Sensors ID 参数.....	34
34) 读取固件版本信息.....	34
35) 固件升级.....	35
36) 重启设备。.....	30
37) 重置设备，恢复出厂。.....	30

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 Bytes
```

```
    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 Bytes
```

```
    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_READ_CUSTOMIZED_PATH
```

```
= 0x51,
```

```
    CMD_WRITE_CUSTOMIZED_PATH
```

```
= 0x52,
```

```
    CMD_GET_CO2_OFFSET
```

```
= 0x53, //读取 CO2 OFFSET 参数
```

```
    CMD_SET_CO2_OFFSET
```

```
= 0x54, //改写 CO2 OFFSET 参数
```

```
    CMD_READ_Email_Account
```

```
= 0x55,
```

```
    CMD_WRITE_Email_Account
```

```
= 0x56,
```

```
    CMD_READ_RAIN
```

```
= 0x57,
```

```
    CMD_WRITE_RAIN
```

```
= 0x58,
```

```
    CMD_LIST_UNKNOWN,
```

```

}CMD_LT;
/*****
#define SOIL_CH_MAX            8
#define WH31_CHANNEL          8
#define PM25_CH_MAX           4
#define LEAK_CH_MAX           4
#define LEAF_CH_MAX           8
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,

    eWH45_SENSOR = 39,
    // GW1000 Firmware V1.5.6 之后加入的
    eWH35_SENSORCH1 = 40,
    eWH35_SENSORCH2 = 41,
    eWH35_SENSORCH3 = 42,
    eWH35_SENSORCH4 = 43,
    eWH35_SENSORCH5 = 44,
    eWH35_SENSORCH6 = 45,
    eWH35_SENSORCH7 = 46,
    eWH35_SENSORCH8 = 47,

    eWH90_SENSOR = 48,

```

//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_RAIN_Gain        0x0F//Rain Gain(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(°C)      3
#define ITEM_TF_USR2           0x64//Temperature(°C)      3
#define ITEM_TF_USR3           0x65//Temperature(°C)      3
#define ITEM_TF_USR4           0x66//Temperature(°C)      3
#define ITEM_TF_USR5           0x67//Temperature(°C)      3
#define ITEM_TF_USR6           0x68//Temperature(°C)      3
#define ITEM_TF_USR7           0x69//Temperature(°C)      3
#define ITEM_TF_USR8           0x6A//Temperature(°C)      3
#define ITEM_HEAP_FREE         0x6C // heap free      4

//封装时解析时的数据在包中的顺序不可改变
#define ITEM_SENSOR_CO2        0x70      16
/* -----Ecowitt-----
1  tf_co2      short      C      x10
2  humi_co2    unsigned char %
3  pm10_co2    unsigned short ug/m3 x10
4  pm10_24h_co2 unsigned short ug/m3 x10
5  pm25_co2    unsigned short ug/m3 x10
6  pm25_24h_co2 unsigned short ug/m3 x10
7  co2         unsigned short ppm
8  co2_24h     unsigned short ppm
9  co2_batt    u8      (0~5)
----- */
//-----
#define ITEM_PM25_AQI          0x71 //only for amb
// ITEM_PM25_AQI length(n*2)(1byte) 1-aqi_pm25 2-aqi_pm25_24h ... n-aqi
/*
aqi_pm25      AQI derived from PM25 int
aqi_pm25_24h  AQI derived from PM25, 24 hour running average int

```


aqi_pm25_in AQI derived from PM25 IN int
aqi_pm25_in_24h AQI derived from PM25 IN, 24 hour running average int
aqi_pm25_aqin AQI derived from PM25, AQIN sensor int
aqi_pm25_24h_aqin AQI derived from PM25, 24 hour running average, AQIN sensor int

.... n

*/

//-----

#define ITEM_LEAF_WETNESS_CH1 0x72// 1

#define ITEM_LEAF_WETNESS_CH2 0x73// 1

#define ITEM_LEAF_WETNESS_CH3 0x74// 1

#define ITEM_LEAF_WETNESS_CH4 0x75// 1

#define ITEM_LEAF_WETNESS_CH5 0x76// 1

#define ITEM_LEAF_WETNESS_CH6 0x77// 1

#define ITEM_LEAF_WETNESS_CH7 0x78// 1

#define ITEM_LEAF_WETNESS_CH8 0x79// 1

//-----

#define ITEM_RAIN_Priority 0x7A// 1 rain priority

#define ITEM_radcompensation 0x7B// 1 Radiation compensation

//-----

#define ITEM_Piezo_Rain_Rate 0x80// 2

#define ITEM_Piezo_Event_Rain 0x81// 2

#define ITEM_Piezo_Hourly_Rain 0x82// 2 Reserved, not used

#define ITEM_Piezo_Daily_Rain 0x83// 4

#define ITEM_Piezo_Weekly_Rain 0x84// 4

#define ITEM_Piezo_Monthly_Rain 0x85// 4

#define ITEM_Piezo_yearly_Rain 0x86// 4

#define ITEM_Piezo_Gain10 0x87// 2*10

#define ITEM_RST_RainTime 0x88// 3

//-----

#if 0 // GW1000 Firmware V1.6.5 开始去除了这个数据结构不用。 在 CMD_READ_SENSOR_ID_NEW 命中为每一个 sensor 做了一个单一的电量电池数据解析。

~~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[16];
—— Struct // 以电压值表示
—— {
—— _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 wh45; // 0~5
—— _wh41_ch wh41; // 电量等级 0~5, 当电量等级 <= 1 时显示低电压。
—— unsigned char wh55[LEAK_CH_MAX]; // 电量等级 0~5, 当电量等级 <= 1 时显示低电压。
—— } val;
} _sensor_batt; // 表示发射传感的电压值与低电
#endif
// V1.5.9 开始加入的
unsigned char wh34[TF_CH_MAX]; // 0.02V * val(received val) = wh34(当前电压值);
// -----

```

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	2	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 Bytes)
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
Upload 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
Upload 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_MulCH_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_MulCH_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°C~10.0°C)x10
.....	
WH31_CHANNEL-1	1	0~7
humidity offset	1	
Temperature offset	1	Range: -100~100, default: 0 Note: (-10.0°C~10.0°C)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°C~10.0°C)x10
.....	
WH31_CHANNEL-1	1	0~7
humidity offset	1	
Temperature offset	1	Range: -100~100, default: 0 Note: (-10.0°C~10.0°C)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	2	
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	2	
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 CO2 OFFSET 校准参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_GET_CO2_OFFSET	1	0x53
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_CO2_OFFSET	1	0x53
Size	1	包长度
CO2 offset	2	Range: -600 ~ 10000, default: 0
PM25offset	2	Range: -200 ~ 200, default: 0 Note: (-20~20 ug/m3)x10
PM10offset	2	Range: -200 ~ 200, default: 0 Note: (-20~20 ug/m3)x10
Checksum	1	checksum

20) 改写 CO2 OFFSET 校准参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_SET_CO2_OFFSET	1	0x54
Size	1	包长度
CO2 offset	2	Range: -600 ~ 10000, default: 0
PM25offset	2	Range: -200 ~ 200, default: 0 Note: (-20~20 ug/m3)x10
PM10offset	2	Range: -200 ~ 200, default: 0 Note: (-20~20 ug/m3)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_CO2_OFFSET	1	0x54
Size	1	包长度
Result	1	0x00:成功, 0x01: 失败
Checksum	1	checksum

21) 读取 Rain 相关的参数

字 段	长度（字节）	说 明
Fixed header	2	固定 0xffff
CMD_READ_RAIN	1	0x57
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_RAIN	1	0x57
Size	2	包长度
ITEM_RAINRATE	2	任意选择其中几个都可 Note: The received rain gain is the value magnified by 100 times, and the rain is the value magnified by 10 times.
ITEM_RAINDAY	4	
ITEM_RAINWEEK	4	
ITEM_RAINMONTH	4	
ITEM_RAINYEAR	4	
ITEM_RAINEVENT	2	
ITEM_RAIN_Gain	2	
ITEM_Piezo_Rain_Rate	2	
ITEM_Piezo_Event_Rain	2	
ITEM_Piezo_Daily_Rain	4	
ITEM_Piezo_Weekly_Rain	4	
ITEM_Piezo_Monthly_Rain	4	
ITEM_Piezo_yearly_Rain	4	
ITEM_RAIN_Priority(Range:0,1,2)	1	
ITEM_Piezo_Gain10(gain0 ~ gain9: Gain0 to gain4 are used, and gain5 ~ gain9 are reserved)	2x10	
ITEM_RST_RainTime (1、rstRainDayTime Range: 0 ~ 23, 2、rstRainWeekTime Range: 0 or 1, Sunday=0, 1:Monday=0; 3、rstRainYearTime Range: 0 ~ 11 , 0:January ~ 11:December. By default all are 0)	3	
Checksum	1	checksum

22) 改写 Rain 相关的参数

字 段	长度 (字节)	说 明
Fixed header	2	固定 0xffff
CMD_WRITE_RAIN	2	0x58
Size	2	包长度
ITEM_RAINRATE	2	任意改写其中几个都可
ITEM_RAINDAY	4	
ITEM_RAINWEEK	4	
ITEM_RAINMONTH	4	
ITEM_RAINYEAR	4	

ITEM_RAINEVENT	2	
ITEM_RAIN_Gain	2	
ITEM_Piezo_Rain_Rate	2	
ITEM_Piezo_Event_Rain	2	
ITEM_Piezo_Daily_Rain	4	
ITEM_Piezo_Weekly_Rain	4	
ITEM_Piezo_Monthly_Rain	4	
ITEM_Piezo_yearly_Rain	4	
ITEM_RAIN_Priority(Range:0,1,2)	1	
ITEM_Piezo_Gain10(gain0 ~ gain9: Gain0 to gain4 are used, and gain5 ~ gain9 are reserved)	2x10	
RST_RainTime (1、rstRainDayTime Range: 0 ~ 23, 2、rstRainWeekTime Range: 0 or 1, Sunday=0, 1:Monday=0; 3、rstRainYearTime Range: 0 ~ 11 , 0:January ~ 11:December. By default all are 0)	3	
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_RAIN	1	0x58
Size	1	包长度
Result	1	0x00:成功, 0x01: 失败
Checksum	1	checksum

23) 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
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

24) 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 Bytes)
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_LOWBATT	1	
Value	16	typedef union _sensor_batt
... ..		
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
.....	
ITEM_SENSOR_CO2	1	
tf_co2 value	2	signed short (valuex10)
humi_co2 value	1	
pm10_co2 value	2	unsigned short(valuex10)
pm10_24h_co2 value	2	unsigned short(valuex10)
pm25_co2 value	2	unsigned short(valuex10)
pm25_24h_co2 value	2	unsigned short(valuex10)
Co2 value	2	unsigned short

co2_24h value	2	unsigned short
co2_batt value	1	0~5
Heap free value	4	Uint32_t
Checksum	1	checksum

25) 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:

typedef union

```
{
    unsigned char flag;
    struct
    {
        unsigned char dst_switch : 1; /* bit 0 DST On(1)/OFF(0) */
        unsigned char auto_tz : 1; /* bit 1 Auto(0)/manual(1) timezone*/
        unsigned char received : 6;
    } bits;
} tz_option_t;
```

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 (注)	1	Wireless Receive Frequency(Only read)
Sensor Type	1	0:WH24 1:WH65
Local TIME	4	Unsigned long(Only read)
Timezone Index	1	Local time zone index
DST Status (tz_option_t)	1	(DST: On(1)/OFF(0)) (Auto_TZ:Auto(0)/manual(1) timezone)
Checksum	1	checksum

typedef enum

```
{
    RFM433M = (unsigned char) 0, // 433MHz
    RFM868M = (unsigned char) 1, // 868MHz
    RFM915M = (unsigned char) 2, // 915MHz
    RFM920M = (unsigned char) 3 // 920MHz
}
```

} freq_t;

extern freq_t Frequency;

26) 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 Index	1	Local time zone index
DST Status	1	True or False
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

27) 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

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

28) 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

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

29) 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°C~10.0°C)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°C~10.0°C)x10
outHumiOffset	1	Range: -10~10, default: 0
windDirOffset	2	Range: -180~180, default: 0
Checksum	1	

30) 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°C~10.0°C)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°C~10.0°C)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

31) 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, // 0.02V * val(received val) = wh80(当前电压值);
    eWH40_SENSOR, //电压=val*0.1V 当<=1.2V 时显示低电压

    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=val*0.1V 当<=1.2V 时显示低电压
    eWH51_SENSORCH2, //电压=val*0.1V 当<=1.2V 时显示低电压
    eWH51_SENSORCH3, //电压=val*0.1V 当<=1.2V 时显示低电压
    eWH51_SENSORCH4, //电压=val*0.1V 当<=1.2V 时显示低电压
    eWH51_SENSORCH5, //电压=val*0.1V 当<=1.2V 时显示低电压
    eWH51_SENSORCH6, //电压=val*0.1V 当<=1.2V 时显示低电压
    eWH51_SENSORCH7, //电压=val*0.1V 当<=1.2V 时显示低电压
    eWH51_SENSORCH8, //电压=val*0.1V 当<=1.2V 时显示低电压
    eWH41_SENSORCH1, // 0~6 : 电量等级 0~5, 当电量等级<=1 时显示低电压; 6 是接 DC
    eWH41_SENSORCH2, // 0~6 : 电量等级 0~5, 当电量等级<=1 时显示低电压; 6 是接 DC
    eWH41_SENSORCH3, // 0~6 : 电量等级 0~5, 当电量等级<=1 时显示低电压; 6 是接 DC
    eWH41_SENSORCH4, // 0~6 : 电量等级 0~5, 当电量等级<=1 时显示低电压; 6 是接 DC

    //-----
    eWH57_SENSOR, //电量等级 0~5, 当电量等级<=1 时显示低电压。
    eWH55_SENSORCH1, //电量等级 0~5, 当电量等级<=1 时显示低电压。
```

eWH55_SENSORCH2, //电量等级 0~5, 当电量等级<=1 时显示低电压。
eWH55_SENSORCH3, //电量等级 0~5, 当电量等级<=1 时显示低电压。
eWH55_SENSORCH4, //电量等级 0~5, 当电量等级<=1 时显示低电压。
eWH34_SENSORCH1 = 31, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH34_SENSORCH2 = 32, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH34_SENSORCH3 = 33, //电压=val*0.02V 当<=1.2V 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
eWH45_SENSOR = 39, // 0~6 : 电量等级 0~5, 当电量等级<=1 时显示低电压; 6 是接 DC
//CMD_READ_SENSOR_ID 命令只能读取到以上的 sensor 的电量状态:

//CMD_READ_SENSOR_ID_NEW 命令才能读取到以下的 sensor 的电量状态:

eWH35_SENSORCH1 = 40, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH35_SENSORCH2 = 41, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH35_SENSORCH3 = 42, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH35_SENSORCH4 = 43, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH35_SENSORCH5 = 44, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH35_SENSORCH6 = 45, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH35_SENSORCH7 = 46, //电压=val*0.02V 当<=1.2V 时显示低电压
eWH35_SENSORCH8 = 47, //电压=val*0.02V 当<=1.2V 时显示低电压

eWH90_SENSOR = 48, // 0.02V * val(received val) = ws90(当前电压值);

//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
Battery	1	
Wh65_signal	1	0~4
WH68_SENSOR	1	0x02
WH68_ID	4	unsigned long
battery	1	
WH68_signal	1	0~4
...SENSOR	1	..
... ID	4	...
battery		
... signal	1	0~4
Checksum	1	

32) 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
battery	1	
Wh65_signal	1	0~4
WH68_SENSOR	1	0x02
WH68_ID	4	unsigned long
battery	1	
WH68_signal	1	0~4
...SENSOR	1	..
... ID	4	...
battery		
..._signal	1	0~4
Checksum	1	

33) 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。

34) 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

35) 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.

36) 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

37) 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