TCP API Interface Communication Protocol

Revision History

Author/Modifier	Version	Date	Remarks
Liao Chaogao	V1.0	2016-09-07	Initial Version
	V1.2.1	2018-08-20	
	V1.4.5	2019-03-15	
	V1.4.6	2019-03-27	Added CMD_READ_SENSOR_ID, CMD_WRITE_SENSOR_ID
	V1.4.7		
	V1.4.8	2019-07-06	Added CMD_GET_MulCH_OFFSET = 0x2C, CMD_SET_MulCH_OFFSET = 0x2D, CMD_GET_PM25_OFFSET = 0x2E, CMD_SET_PM25_OFFSET = 0x2F
	V1.4.9	2019-07-29	Correction
	V1.5.0	2019-08-16	Added usr_path[128]
	V1.5.1	2019-08-20	Added eWH57_SENSOR, eWH55_SENSORCH1...eWH55_SENSORCH4, ITEM_LEAK_CH1...CH4, ITEM_THUNDERDISTANCE
	V1.5.2	2020-04-30	Added ITEM_RAINEVENT data output

	V1.5.3	2020-05-06	Added ITEM_TF_USR1 to ITEM_TF_USR8, ITEM_TF_BATT
	V1.5.4	2020-05-07	Added command CMD_READ_SENSOR_ID_NEW
	V1.5.5	2020-06-12	Deleted ITEM_TF_BATT. Added 1 byte battery data after ITEM_TF_USRch (ch = 1... 8) when reading Live data.
	V1.5.6	2020-06-22	Corrected: CMD_READ_SSSS (Read system info), CMD_WRITE_SSSS (Write system info)
	V1.5.7	2020-07-16	Corrected CMD_READ_SENSOR_ID, CMD_READ_SENSOR_ID_NEW
	V1.5.8	2020-07-20	Added #define ITEM_SENSOR_CO2 0x70
	V1.5.9	2020-08-13	Added: CMD_GET_CO2_OFFSET = 0x53, CMD_SET_CO2_OFFSET = 0x54
	V1.6.0	2021-01-05	Version: GW1000_Firmware V1.6.5 {leaf wetness sensor}. Added ITEM ("ITEM_LEAF_WETNESS_CHx"(x=1~8))
	V1.6.1	2021-08-17	Corrected sensor battery status in CMD_READ_SENSOR_ID and CMD_READ_SENSOR_ID_NEW commands
	V1.6.2	2021-11-04	Added sensor id for WS90. Added CMD_READ_RSTRAIN_TIME, CMD_WRITE_RSTRAIN_TIME
	V1.6.3	2021-11-04	Correction: For WN34 multi-channel, ITEM_ TF_USR(1~8) valid length is 3 bytes.
	V1.6.4	2022-03-05	Added CMD_READ_RAIN = 0x57, CMD_WRITE_RAIN = 0x58
	V1.6.5	2022-04-19	Modified ITEM_RAINHOUR to ITEM_RAIN_Gain (This is Rain Gain). ITEM_Piezo_Hourly_Rain (Reserved, not used.)
	V1.6.6	2022-04-20	Added ITEM_RAIN_Priority 0x7A // 1 rain priority
	V1.6.7	2022-06-17	Correction

	V1.6.8	2023-02-27	To keep CMD_READ_RAIN and CMD_WRITE_RAIN consistent, added ITEM_radcompensation
	V1.6.9	2024-01-15	Added ITEM_HEAP_FREE to CMD_GW1000_LIVEDATA command
	V1.7.0	2024-05-27	Added ITEM_SENSOR_CO2_WH46 to CMD_GW1000_LIVEDATA command

Table of Contents

26. Write System Parameters
27. Read Rain Data
28. Read Calibration Gains
29. Read Calibration Offsets
30. Write Calibration Offsets
31. Read Sensor IDs
32. Read Sensor Related Parameters
33. Write Sensor IDs
34. Read Firmware Version
35. Firmware Update
36. Reboot Device
37. Factory Reset

1. Data Exchange Format

Format: `Fixed header, CMD, SIZE, DATA1, DATA2, ... , DATAn, CHECKSUM`

- **Fixed header:** 2 bytes, fixed as `0xffff`
- **CMD:** 1 byte, Command
- **SIZE:** 1 byte, Size of the packet (from CMD to CHECKSUM)
- **DATA:** n bytes, Transmitted data, variable length
- **CHECKSUM:** 1 byte, Checksum = `CMD + SIZE + DATA1 + DATA2 + ... + DATAn`

2. Command and Data Structure Definitions

```
typedef  enum  {
//General Commands
CMD_WRITE_SSID              = 0x11, // Send router SSID and Password to WIFI module
CMD_BROADCAST               = 0x12, // Broadcast command to find device in LAN (Note:
Return data size is 2 Bytes)
CMD_READ_ECOWITT            = 0x1E, // Read Ecowitt.net settings
CMD_WRITE_ECOWITT           = 0x1F, // Write Ecowitt.net settings
CMD_READ_WUNDERGROUND       = 0x20, // Read Wunderground settings
CMD_WRITE_WUNDERGROUND      = 0x21, // Write Wunderground settings
CMD_READ_WOW                = 0x22, // Read WOW settings
CMD_WRITE_WOW               = 0x23, // Write WOW settings
CMD_READ_WEATHERCLOUD       = 0x24, // Read Weathercloud settings
CMD_WRITE_WEATHERCLOUD      = 0x25, // Write Weathercloud settings
CMD_READ_SATION_MAC         = 0x26, // Read MAC address
CMD_READ_CUSTOMIZED         = 0x2A, // Read Customized server parameters
```

```
CMD_WRITE_CUSTOMIZED        = 0x2B, // Write Customized server parameters
CMD_WRITE_UPDATE            = 0x43, // Firmware update
CMD_READ_FIRMWARE_VERSION   = 0x50, // Read firmware version
CMD_READ_USR_PATH           = 0x51,
CMD_WRITE_USR_PATH          = 0x52,

// The following commands are supported by GW1000 and WH2650 only:
CMD_GW1000_LIVEDATA         = 0x27, // Read live data (Note: Return data size is 2 Bytes)
CMD_GET_SOILHUMIAD          = 0x28, // Read Soil Moisture Sensor calibration parameters
CMD_SET_SOILHUMIAD          = 0x29, // Write Soil Moisture Sensor calibration parameters
CMD_GET_MulCH_OFFSET        = 0x2C, // Read multi-channel Temp/Humidity OFFSET parameters
CMD_SET_MulCH_OFFSET        = 0x2D, // Write multi-channel Temp/Humidity OFFSET parameters
CMD_GET_PM25_OFFSET         = 0x2E, // Read multi-channel PM2.5 OFFSET calibration
parameters
CMD_SET_PM25_OFFSET         = 0x2F, // Write multi-channel PM2.5 OFFSET calibration
parameters
CMD_READ_SSSS               = 0x30, // Read system info
CMD_WRITE_SSSS              = 0x31, // Write system info
CMD_READ_RAINDATA           = 0x34, // Read rain data
CMD_WRITE_RAINDATA          = 0x35, // Write rain data
CMD_READ_GAIN               = 0x36, // Read calibration gains
CMD_WRITE_GAIN              = 0x37, // Write calibration gains
CMD_READ_CALIBRATION        = 0x38, // Read calibration offsets
CMD_WRITE_CALIBRATION       = 0x39, // Write calibration offsets
CMD_READ_SENSOR_ID          = 0x3A, // Read Sensors ID parameters
CMD_WRITE_SENSOR_ID         = 0x3B, // Write Sensors ID parameters
CMD_READ_SENSOR_ID_NEW      = 0x3C, // Read Sensors ID parameters (Added for expansion)
CMD_WRITE_REBOOT            = 0x40, // Reboot system
CMD_WRITE_RESET             = 0x41, // Reset to factory settings
CMD_READ_CUSTOMIZED_PATH    = 0x51,
CMD_WRITE_CUSTOMIZED_PATH   = 0x52,
CMD_GET_CO2_OFFSET          = 0x53, // Read CO2 OFFSET parameters
CMD_SET_CO2_OFFSET          = 0x54, // Write CO2 OFFSET parameters
CMD_READ_Email_Account      = 0x55,
CMD_WRITE_Email_Account     = 0x56,
CMD_READ_RAIN               = 0x57,
CMD_WRITE_RAIN              = 0x58,
CMD_LIST_UNKNOW,
} 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 (Transmission interval 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,
    // Added after GW1000 Firmware V1.5.6
    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,
    // Add new sensors starting from here, previous order cannot be changed
    //-------------
    eMAX_SENSOR
} SENSOR_IDT;

//------------------------------------------------------------------------------------------
-------------------------------------------------
#define ITEM_INTEMP             0x01 // Indoor Temperature (℃) 2 bytes
#define ITEM_OUTTEMP            0x02 // Outdoor Temperature (℃) 2 bytes
#define ITEM_DEWPOINT           0x03 // Dew point (℃) 2 bytes
#define ITEM_WINDCHILL          0x04 // Wind chill (℃) 2 bytes
#define ITEM_HEATINDEX          0x05 // Heat index (℃) 2 bytes
#define ITEM_INHUMI             0x06 // Indoor Humidity (%) 1 byte
#define ITEM_OUTHUMI            0x07 // Outdoor Humidity (%) 1 byte
#define ITEM_ABSBARO            0x08 // Absolutely Barometric (hpa) 2 bytes
#define ITEM_RELBARO            0x09 // Relative Barometric (hpa) 2 bytes
#define ITEM_WINDDIRECTION      0x0A // Wind Direction (360°) 2 bytes
#define ITEM_WINDSPEED          0x0B // Wind Speed (m/s) 2 bytes
#define ITEM_GUSTSPEED          0x0C // Gust Speed (m/s) 2 bytes
#define ITEM_RAINEVENT          0x0D // Rain Event (mm) 2 bytes
#define ITEM_RAINRATE           0x0E // Rain Rate (mm/h) 2 bytes
#define ITEM_RAIN_Gain          0x0F // Rain Gain (mm) 2 bytes
#define ITEM_RAINDAY            0x10 // Rain Day (mm) 2 bytes
#define ITEM_RAINWEEK           0x11 // Rain Week (mm) 2 bytes
#define ITEM_RAINMONTH          0x12 // Rain Month (mm) 4 bytes
#define ITEM_RAINYEAR           0x13 // Rain Year (mm) 4 bytes
#define ITEM_RAINTOTALS         0x14 // Rain Totals (mm) 4 bytes
#define ITEM_LIGHT              0x15 // Light (lux) 4 bytes
#define ITEM_UV                 0x16 // UV (uW/m2) 2 bytes
#define ITEM_UVI                0x17 // UVI (0-15 index) 1 byte
#define ITEM_TIME               0x18 // Date and time 6 bytes
#define ITEM_DAYLWINDMAX        0x19 // Day max wind (m/s) 2 bytes
#define ITEM_TEMP1              0x1A // Temperature 1 (℃) 2 bytes
```

```
#define ITEM_TEMP2              0x1B // Temperature 2 (℃) 2 bytes
#define ITEM_TEMP3              0x1C // Temperature 3 (℃) 2 bytes
#define ITEM_TEMP4              0x1D // Temperature 4 (℃) 2 bytes
#define ITEM_TEMP5              0x1E // Temperature 5 (℃) 2 bytes
#define ITEM_TEMP6              0x1F // Temperature 6 (℃) 2 bytes
#define ITEM_TEMP7              0x20 // Temperature 7 (℃) 2 bytes
#define ITEM_TEMP8              0x21 // Temperature 8 (℃) 2 bytes
#define ITEM_HUMI1              0x22 // Humidity 1, 0-100% 1 byte
#define ITEM_HUMI2              0x23 // Humidity 2, 0-100% 1 byte
#define ITEM_HUMI3              0x24 // Humidity 3, 0-100% 1 byte
#define ITEM_HUMI4              0x25 // Humidity 4, 0-100% 1 byte
#define ITEM_HUMI5              0x26 // Humidity 5, 0-100% 1 byte
#define ITEM_HUMI6              0x27 // Humidity 6, 0-100% 1 byte
#define ITEM_HUMI7              0x28 // Humidity 7, 0-100% 1 byte
#define ITEM_HUMI8              0x29 // Humidity 8, 0-100% 1 byte
#define ITEM_PM25_CH1           0x2A // PM2.5 Air Quality Sensor (μg/m3) 2 bytes
#define ITEM_SOILTEMP1          0x2B // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE1      0x2C // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP2          0x2D // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE2      0x2E // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP3          0x2F // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE3      0x30 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP4          0x31 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE4      0x32 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP5          0x33 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE5      0x34 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP6          0x35 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE6      0x36 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP7          0x37 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE7      0x38 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP8          0x39 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE8      0x3A // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP9          0x3B // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE9      0x3C // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP10         0x3D // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE10     0x3E // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP11         0x3F // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE11     0x40 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP12         0x41 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE12     0x42 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP13         0x43 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE13     0x44 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP14         0x45 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE14     0x46 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP15         0x47 // Soil Temperature (℃) 2 bytes
```

```
#define ITEM_SOILMOISTURE15     0x48 // Soil Moisture (%) 1 byte
#define ITEM_SOILTEMP16         0x49 // Soil Temperature (℃) 2 bytes
#define ITEM_SOILMOISTURE16     0x4A // Soil Moisture (%) 1 byte
#define ITEM_LOWBATT            0x4C // All sensor lowbatt 16 char 16 bytes
#define ITEM_PM25_24HAVG1       0x4D // for pm25_ch1 2 bytes
#define ITEM_PM25_24HAVG2       0x4E // for pm25_ch2 2 bytes
#define ITEM_PM25_24HAVG3       0x4F // for pm25_ch3 2 bytes
#define ITEM_PM25_24HAVG4       0x50 // for pm25_ch4 2 bytes
#define ITEM_PM25_CH2           0x51 // PM2.5 Air Quality Sensor (μg/m3) 2 bytes
#define ITEM_PM25_CH3           0x52 // PM2.5 Air Quality Sensor (μg/m3) 2 bytes
#define ITEM_PM25_CH4           0x53 // PM2.5 Air Quality Sensor (μg/m3) 2 bytes
#define ITEM_LEAK_CH1           0x58 // for Leak_ch1 1 byte
#define ITEM_LEAK_CH2           0x59 // for Leak_ch2 1 byte
#define ITEM_LEAK_CH3           0x5A // for Leak_ch3 1 byte
#define ITEM_LEAK_CH4           0x5B // for Leak_ch4 1 byte
#define ITEM_LIGHTNING          0x60 // Lightning distance (1~40KM) 1 byte
#define ITEM_LIGHTNING_TIME     0x61 // Lightning detection time (UTC) 4 bytes
#define ITEM_LIGHTNING_POWER    0x62 // Lightning strikes today 4 bytes
#define ITEM_TF_USR1            0x63 // Temperature (℃) 3 bytes
#define ITEM_TF_USR2            0x64 // Temperature (℃) 3 bytes
#define ITEM_TF_USR3            0x65 // Temperature (℃) 3 bytes
#define ITEM_TF_USR4            0x66 // Temperature (℃) 3 bytes
#define ITEM_TF_USR5            0x67 // Temperature (℃) 3 bytes
#define ITEM_TF_USR6            0x68 // Temperature (℃) 3 bytes
#define ITEM_TF_USR7            0x69 // Temperature (℃) 3 bytes
#define ITEM_TF_USR8            0x6A // Temperature (℃) 3 bytes
#define ITEM_SENSOR_CO2_WH46    0x6B // ITEM_SENSOR_CO2 + pm1 + pm4;
#define ITEM_HEAP_FREE          0x6C // heap free 4 bytes

// The order of data in the packet cannot be changed during encapsulation/parsing
#define ITEM_SENSOR_CO2         0x70 // 16 bytes
/* ------------------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 byte
#define ITEM_LEAF_WETNESS_CH2   0x73 // 1 byte
#define ITEM_LEAF_WETNESS_CH3   0x74 // 1 byte
#define ITEM_LEAF_WETNESS_CH4   0x75 // 1 byte
#define ITEM_LEAF_WETNESS_CH5   0x76 // 1 byte
#define ITEM_LEAF_WETNESS_CH6   0x77 // 1 byte
#define ITEM_LEAF_WETNESS_CH7   0x78 // 1 byte
#define ITEM_LEAF_WETNESS_CH8   0x79 // 1 byte
//--------------------------------------------------------------------------------------------
-----------------------------------
#define ITEM_RAIN_Priority      0x7A // 1 byte rain priority
#define ITEM_radcompensation    0x7B // 1 byte Radiation compensation
//--------------------------------------------------------------------------------------------
-----------------------------------
#define ITEM_Piezo_Rain_Rate    0x80 // 2 bytes
#define ITEM_Piezo_Event_Rain   0x81 // 2 bytes
#define ITEM_Piezo_Hourly_Rain  0x82 // 2 bytes Reserved, not used
#define ITEM_Piezo_Daily_Rain   0x83 // 4 bytes
#define ITEM_Piezo_Weekly_Rain  0x84 // 4 bytes
#define ITEM_Piezo_Monthly_Rain 0x85 // 4 bytes
#define ITEM_Piezo_yearly_Rain  0x86 // 4 bytes
#define ITEM_Piezo_Gain10       0x87 // 2*10 bytes
#define ITEM_RST_RainTime       0x88 // 3 bytes
//--------------------------------------------------------------------------------------------
-------------------------------------------------

#if 0 // GW1000 Firmware V1.6.5 removed this data structure. In CMD_READ_SENSOR_ID_NEW, a
single battery data parsing is done for each sensor.
typedef union // 1 Low Batt, 0 Normal
{
    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 Low Batt, 0 Normal
{
    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 Low Batt, 0 Normal
{
    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;

// Represents sensor voltage value and low battery status
typedef union _sensor_batt
{
    unsigned char all_batt[16];
    struct // Represented by voltage value
    {
        _sig_sen single;
        _wh31_ch wh31;
        _wh51_ch wh51;
        unsigned char wh57; // 0~5
        unsigned char wh68; // 0.02V * val(received val) = wh68(current voltage);
        unsigned char wh80; // 0.02V * val(received val) = wh80(current voltage);
        unsigned char wh45; // 0~5
        _wh41_ch wh41;      // Battery level 0~5, displays low voltage when level <= 1.
        unsigned char wh55[LEAK_CH_MAX]; // Battery level 0~5, displays low voltage when
level <= 1.
    } val;
} sensor_batt; // Represents sensor voltage value and low battery status
#endif

// Added since V1.5.9
unsigned char wh34[TF_CH_MAX]; // 0.02V * val(received val) = wh34(current voltage);
//------------------------------------------------------------------------------------
-------------------------------------------------
```

3. Network Provisioning and Device Discovery in LAN

Provisioning Mode 1: The mobile app sets up a TCP Server on port 49123. The WIFI module (in station+AP mode) creates a TCP Client to connect to the App. Upon successful connection,

the App's TCP Server sends the `CMD_WRITE_SSID` command.

Provisioning Mode 2: The WIFI module (in station+AP mode) sets up a TCP Server on port 45000 and waits for the mobile App to connect. Once connected, the App sends the `CMD_WRITE_SSID` command.

Command: `CMD_WRITE_SSID` (0x11)

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_SSID	1	0x11
Size	2	Packet length
SSID Size	1	SSID length
SSID	n	Max 32
Password Size	1	Password length
Password	n	Max 64
Checksum	1	Checksum

WIFI Module Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_SSID	1	0x11
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

When both WIFI and APP are connected to the same router, the APP sends a broadcast (UDP) command. The module replies with its MAC, IP, PORT, and AP SSID. (Destination port 46000)

Command: `CMD_BROADCAST` (0x12)

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_BROADCAST	1	0x12
Size	1	Packet length
Checksum	1	Checksum

WIFI Module Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_BROADCAST	1	0x12
Size	2	Packet length (Return data size is 2 Bytes)
MAC	6	Module STA MAC
IP1	1	Eg. 192 in 192.168.100.1
IP2	1	Eg. 168 in 192.168.100.1
IP3	1	Eg. 100 in 192.168.100.1
IP4	1	Eg. 1 in 192.168.100.1
Port1	1	Eg. 0x11 in 0x1194(45000)
Port2	1	Eg. 0x94 in 0x1194(45000)
AP SSID	n	Module AP SSID
Checksum	1	Checksum

The WIFI module sets up a TCP Server on port 45000 and waits for the mobile App to connect. Once connected, the following commands can be used:

1) Read Ecowitt.net Settings

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_ECOWITT	1	0x1E
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_ECOWITT	1	0x1E
Size	1	Packet length
Upload interval	1	0~5min (0: mean is OFF)
Checksum	1	Checksum

2) Write Ecowitt.net Settings

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_ECOWITT	1	0x1F
Size	1	Packet length
Upload interval	1	0~5min (0: mean is OFF)
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description

Fixed header	2	Fixed 0xffff
CMD_WRITE_ECOWITT	1	0x1F
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

3) Read Wunderground Settings

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_READ_WUNDERGROUND	1	0x20
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_READ_WUNDERGROUND	1	0x20
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 32
Password Size	1	Password Size
Password	n	ASCII code, max 32
Fix	1	1
Checksum	1	Checksum

4) Write Wunderground Settings

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_WUNDERGROUND	1	0x21
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 32
Password Size	1	Password Size
Password	n	ASCII code, max 32
Fix	1	1
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_WUNDERGROUND	1	0x21
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

5) Read WOW Settings

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_WOW	1	0x22

Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_WOW	1	0x22
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 39
Password Size	1	Password Size
Password	n	ASCII code, max 32
StationNum Size (unused)	1	StationNum size (unused)
StationNum (unused)	n	ASCII code, max 32 (unused)
Fix	1	1
Checksum	1	Checksum

6) Write WOW Settings

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_WOW	1	0x23
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 39

Password Size	1	Password Size
Password	n	ASCII code, max 32
StationNum Size (unused)	1	StationNum size (unused)
StationNum (unused)	32	ASCII code, max 32 (unused)
Fix	1	1
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_WOW	1	0x23
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

7) Read Weathercloud Settings

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_WEATHERCLOUD	1	0x24
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description

Fixed header	2	Fixed 0xffff
CMD_READ_WEATHERCLOUD	1	0x24
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 32
Key Size	1	Key Size
Key	n	ASCII code, max 32
Fix	1	1
Checksum	1	Checksum

8) Write Weathercloud Settings

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_WRITE_WEATHERCLOUD	1	0x25
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 32
Key Size	1	Key Size
Key	n	ASCII code, max 32
Fix	1	1
Checksum	1	Checksum

Response:

Field	**Length (Byte)**	**Description**

Fixed header	2	Fixed 0xffff
CMD_WRITE_WEATHERCLOUD	1	0x25
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

9) Read Customer Server Parameters

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_READ_CUSTOMIZED	1	0x2A
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_READ_CUSTOMIZED	1	0x2A
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 40
Password Size	1	Password Size
Password	n	ASCII code, max 40
Server Size	1	Server Size
Server	n	ASCII code, max 64

Port	2	0-65535
Interval	2	16-600
Type	1	0:EC 1:WU
Active	1	0:Disable 1:Enable
Checksum	1	Checksum

10) Write Customer Server Parameters

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_WRITE_CUSTOMIZED	1	0x2B
Size	1	Packet length
ID Size	1	ID Size
ID	n	ASCII code, max 40
Password Size	1	Password Size
Password	n	ASCII code, max 40
Server Size	1	Server Size
Server	n	ASCII code, max 64
Port	2	0-65535
Interval	2	16-600
Type	1	0:EC 1:WU
Active	1	0:Disable 1:Enable
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_CUSTOMIZED	1	0x2B
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

11) Read Customer User Path

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_USRPATH	1	0x51
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_USRPATH	1	0x51
Size	1	Packet length
Ecowitt Path	64	ASCII code, max 64
WU Path	64	ASCII code, max 64
Checksum	1	Checksum

12) Write Customer User Path

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_USRPATH	1	0x52
Size	1	Packet length
Ecowitt Path	64	ASCII code, max 64
WU Path	64	ASCII code, max 64
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_USRPATH	1	0x52
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

13) Read Soil Moisture Sensor Calibration

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_SOILHUMIAD	1	0x28
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_SOILHUMIAD	1	0x29
Size	1	Packet length
Channel	1	Channel Number
Current humidity	1	Sent by transmitter
Current ad	2	Sent by transmitter
Humidity ad select	1	Option switch
Min ad	1	Min calibration AD (70~200)
Max ad	2	Max calibration AD (80~1000)
...	...	
Checksum	1	Checksum

14) Write Soil Moisture Sensor Calibration

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_SET_SOILHUMIAD	1	0x29
Size	1	Packet length
Channel	1	Channel Number
Humidity ad select	1	Option switch
Min ad	1	Min calibration AD (70~200)
Max ad	2	Max calibration AD (80~1000)
...	...	

Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_SET_SOILHUMIAD	1	0x29
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

15) Read Multi-channel Temp/Humidity Offsets

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_MulCH_OFFSET	1	0x2C
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_MulCH_OFFSET	1	0x2C
Size	1	Packet length
Channel	1	Channel Number
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) Write Multi-channel Temp/Humidity Offsets

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_SET_MulCH_OFFSET	1	0x2D
Size	1	Packet length
Channel	1	Channel Number
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

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_SET_MulCH_OFFSET	1	
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

17) Read Multi-channel PM2.5 Offsets

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_PM25_OFFSET	1	0x2E
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_PM25_OFFSET	1	0x2E
Size	1	Packet length
Channel	1	Channel Number
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) Write Multi-channel PM2.5 Offsets

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_SET_PM25_OFFSET	1	0x2F
Size	1	Packet length
Channel	1	Channel Number
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

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_SET_PM25_OFFSET	1	0x2F
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

19) Read CO2 Offset Calibration

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_CO2_OFFSET	1	0x53
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GET_CO2_OFFSET	1	0x53
Size	1	Packet length
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) Write CO2 Offset Calibration

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_SET_CO2_OFFSET	1	0x54
Size	1	Packet length
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

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_SET_CO2_OFFSET	1	0x54
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

21) Read Rain Related Parameters

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_RAIN	1	0x57
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_RAIN	1	0x57

Size	2	Packet length
ITEM_RAINRATE	2	Can choose any (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	1	Range: 0,1,2
ITEM_Piezo_Gain10	2x10	Gain0 to gain4 are used, and gain5 ~ gain9 are reserved
ITEM_RST_RainTime	3	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
Checksum	1	Checksum

22) Write Rain Related Parameters

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_RAIN	2	0x58
Size	2	Packet length
ITEM_RAINRATE	2	Can rewrite any
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	1	Range: 0,1,2
ITEM_Piezo_Gain10	2x10	Gain0 to gain4 are used, and gain5 ~ gain9 are reserved
RST_RainTime	3	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

Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_RAIN	1	0x58
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

23) Read MAC Address

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_SATION_MAC	1	0x26
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_SATION_MAC	1	0x26
Size	1	Packet length
Sta_mac[6]	6	sta_mac[0];...;sta_mac[5];
Checksum	1	Checksum

24) Read Live Data

Note: Return data size is 2 Bytes

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GW1000_LIVEDATA	1	0x27
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_GW1000_LIVEDATA	1	0x27
Size	2	Packet length (Return data size is 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
ITEM_SENSOR_CO2_WH46	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
pm1_co2 value	2	unsigned short(valuex10)
pm1_24h_co2 value	2	unsigned short(valuex10)
Pm4_co2 value	2	unsigned short(valuex10)
Pm4_24h_co2 value	2	unsigned short(valuex10)
Checksum	1	Checksum

25) Read System Parameters

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff

CMD_READ_SSSS	1	0x30
Size	1	Packet length
Checksum	1	Checksum

Response Structure:

```
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;
```

Response Table:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_SSSS	1	0x30
Size	1	Packet length
Frequency (Note)	1	Wireless Receive Frequency (Read Only)
Sensor Type	1	0:WH24 1:WH65
Local TIME	4	Unsigned long (Read Only)
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) Write System Parameters

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_SSSS	1	0x31
Size	1	Packet length
Frequency	1	(Read Only) Can't be rewritten.
Sensor Type	1	0:WH24 1:WH65
UTC TIME	4	Unsigned long (Read Only)
Timezone Index	1	Local time zone index
DST Status	1	True or False
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_SSSS	1	0x31
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail

Checksum	1	Checksum

27) Read Rain Data

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_RAINDATA	1	0x34
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_RAINDATA	1	0x34
Size	1	Packet length
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	

28) Read Calibration Gains

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff

CMD_READ_GAIN	1	0x36
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_GAIN	1	0x36
Size	1	Packet length
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	

29) Read Calibration Offsets

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_CALIBRATION	1	0x38
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_CALIBRATION	1	0x38
Size	1	Packet length
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	

30) Write Calibration Offsets

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_CALIBRATION	1	0x39
Size	1	Packet length
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	

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_CALIBRATION	1	0x39
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

31) Read Sensor IDs

```
typedef enum {
    //eWH24_SENSOR = 0x00,
    eWH65_SENSOR = 0x00, // Voltage flag, 1 means low voltage, 0 normal
    //eWH69_SENSOR,
    eWH68_SENSOR, // Voltage = val*0.02V, shows low voltage when <=1.2V
    eWH80_SENSOR, // 0.02V * val(received val) = wh80(current voltage);
    eWH40_SENSOR, // Voltage = val*0.1V, shows low voltage when <=1.2V
    eWH25_SENSOR, // Voltage flag, 1 means low voltage, 0 normal
    eWH26_SENSOR, // Voltage flag, 1 means low voltage, 0 normal
    eWH31_SENSORCH1, // Voltage flag, 1 means low voltage, 0 normal
    // ... (CH2 to CH8 omitted, same as CH1)
    eWH51_SENSORCH1, // Voltage = val*0.1V, shows low voltage when <=1.2V
```

```
    // ... (CH2 to CH8 omitted, same as CH1)
    eWH41_SENSORCH1, // 0~6: battery level 0~5, low voltage when <=1; 6 is DC power
    // ... (CH2 to CH4 omitted, same as CH1)
    eWH57_SENSOR, // battery level 0~5, low voltage when <=1
    eWH55_SENSORCH1, // battery level 0~5, low voltage when <=1
    // ... (CH2 to CH4 omitted, same as CH1)
    eWH34_SENSORCH1 = 31, // Voltage = val*0.02V, shows low voltage when <=1.2V
    // ... (CH2 to CH8 omitted, same as CH1)
    eWH45_SENSOR = 39, // 0~6: battery level 0~5, low voltage when <=1; 6 is DC power
    // CMD_READ_SENSOR_ID command can only read battery status of the above sensors.
    // CMD_READ_SENSOR_ID_NEW command can read battery status of the following sensors:
    eWH35_SENSORCH1 = 40, // Voltage = val*0.02V, shows low voltage when <=1.2V
    // ... (CH2 to CH8 omitted, same as CH1)
    eWH90_SENSOR = 48, // 0.02V * val(received val) = ws90(current voltage);
    eMAX_SENSOR
} SENSOR_IDT;
```

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_SENSOR_ID	1	0x3A
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_SENSOR_ID	1	0x3A
Size	1	Packet length
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) Read Sensor Related Parameters

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_SENSOR_ID_NEW	1	0x3C
Size	1	Packet length
Checksum	1	Checksum

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_SENSOR_ID_NEW	1	0x3C
Size	2	Packet length
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) Write Sensor IDs

Field	**Length (Byte)**	**Description**
Fixed header	2	Fixed 0xffff
CMD_WRITE_SENSOR_ID	1	0x3B
Size	1	Packet length
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	

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_SENSOR_ID	1	0x3B
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

Remark: Writing ID = 0xFFFFFFFF means re-registering the corresponding transmitter ID.
Writing ID = 0xFFFFFFFE means disabling this transmitter.

34) Read Firmware Version

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_FIRMWARE_VERSION	1	0x50
Size	1	Packet length
Checksum	1	

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_READ_FIRMWARE_VERSION	1	0x50
Size	1	Packet length

Version length	1	Max length 23 Bytes
Version buffer		For example: "EasyWeatherV1.2.0"
Checksum	1	Checksum

35) Firmware Update

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_UPDATE	1	0x43
Size	1	Packet length
ServerIP	4	0xc0a80063 // "192.168.0.99"
ServerPort	2	1~65535
Checksum	1	

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_UPDATE	1	0x43
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

Update Process:

User clicks "Update firmware", App sends server IP and port to the module, module connects to server:

1. Client connects to server, sends filename ("user1.bin" or "user2.bin"). Server responds with file length.

2. Client receives file length, sends "start". Server sends 1st packet of firmware data (packet size 1460 bytes).
3. Client receives 1st packet, sends "continue". Server responds with 2nd packet.
4. ...
5. Client receives n-1 packet, sends "continue". Server responds with n packet.
6. Client receives nth packet. If successful, sends "end".

36) Reboot Device

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_REBOOT	1	0x40
Size	1	Packet length
Checksum	1	

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_REBOOT	1	0x40
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum

37) Factory Reset

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_RESET	1	0x41
Size	1	Packet length

Checksum	1	

Response:

Field	Length (Byte)	Description
Fixed header	2	Fixed 0xffff
CMD_WRITE_RESET	1	0x41
Size	1	Packet length
Result	1	0x00: Success, 0x01: Fail
Checksum	1	Checksum