传感器总计 : 电量 温度、湿度、光照、PM2.5

# 1 帧格式：

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 说明 |
| 1 | Power\_H | 电量整数位 |
| 2 | Power\_L | 电量小数位 |
| 3 | humidity\_H | 湿度整数位 |
| 4 | humidity\_L | 湿度小数位 |
| 5 | temperature\_H | 温度整数位 |
| 6 | temperature\_L | 温度小数位 |
| 7 | light\_H | 光照高位 |
| 8 | light\_L; | 光照低位 |
| 9 | PM2\_5\_H | PM2.5高位 |
| 10 | PM2\_5\_L | PM2.5低位 |

# 2 数据解析格式：

## 2.1. 温度

两个字节长度，高位（temperature\_H），低位（temperature\_L）

单位: 摄氏度

范围：-20°~80°

温度高位: temperature\_H = 温度整数

温度低位: temperature\_L = 温度小数

例: temperature\_H = 0x0f

temperature\_L = 0x01

实际温度值: 15.1℃

## 2.2 湿度

两个字节长度，高位（humidity\_H），低位（humidity\_L）

单位 %RH ,单位体积气体(通常为空气)中水蒸气含量与相同体积气体中水蒸气饱和时水蒸气含量的百分比.。

范围 0~100

湿度高位: humidity\_H = 湿度整数

湿度低位: humidity\_L = 湿度小数

例: humidity\_H = 0x2f

humidity\_L = 0x03

实际湿度值: 47.3%

## 2.3 光照

两个字节长度 高位(light\_H), 低位（light\_L）。

单位: 勒克司lux 。为距离一个光强为1cd的光源，在1米处接受的照明强度

范围: 0~65535

实际光照强度 = light\_H \* 256 + light\_L

## 2.5 电量

两个字节长度 高位（Power\_H），低位（Power\_L）

单位: V伏。

范围 0~ 4.2V

电压高位: Power\_H = 电压整数

电压低位: Power\_L = 电压小数

实际电压:

例: Power\_H = 0x03

Power\_L = 0x02

实际湿度值: 3.2V

## 2.6 PM2.5

两个字节长度，PM2\_5\_H，和PM2\_5\_L。

单位: 微克每立方米，ug/m^3

范围: 0~6000.

实际PM2.5 = PM2\_5\_H \*256 +PM2\_5\_L 。

# 3.错误信息标识

## 3.1 温度

|  |  |  |
| --- | --- | --- |
| 最终数值 | 象征 | 说明 |
| -110 | 获取超时 | 未读取到传感器数据，传感器丢失或损坏 |
| -111 | 校验错误 | 读取的传感器数据校验错误 |
| -112 | 数值异常 | 采集的相关数值异常 |

## 3.2 湿度

|  |  |  |
| --- | --- | --- |
| 最终数值 | 象征 | 说明 |
| -110 | 获取超时 | 未读取到传感器数据，传感器丢失或损坏 |
| -111 | 校验错误 | 读取的传感器数据校验错误 |
| -112 | 数值异常 | 采集的相关数值异常 |

## 3.3 光照

|  |  |  |
| --- | --- | --- |
| 最终数值 | 象征 | 说明 |
| -110 | 获取超时 | 未读取到传感器数据，传感器丢失或损坏 |
| -111 | 校验错误 | 读取的传感器数据校验错误 |
| -112 | 数值异常 | 采集的相关数值异常 |

## 3.6 电压

|  |  |  |
| --- | --- | --- |
| 最终数值 | 象征 | 说明 |
| -110 | 获取超时 | 未读取到传感器数据，传感器丢失或损坏 |
| -111 | 校验错误 | 读取的传感器数据校验错误 |
| -112 | 数值异常 | 采集的相关数值异常 |

## 3.7 PM2.5

注，传感器数据范围虽然有0到6000

|  |  |  |
| --- | --- | --- |
| 最终数值 | 象征 | 说明 |
| -110 | 获取超时 | 未读取到传感器数据，传感器丢失或损坏 |
| -111 | 校验错误 | 读取的传感器数据校验错误 |
| -120 | 数值异常 | 采集的相关数值异常 |

enum{

SIX\_BOX\_POWER\_DATA\_TYPE = 1,

SIX\_BOX\_HUMI\_DATA\_TYPE = 2,

SIX\_BOX\_TEMP\_DATA\_TYPE = 3,

SIX\_BOX\_LIGHT\_DATA\_TYPE = 4

SIX\_BOX\_PM25\_DATA\_TYPE = 5,

};

//data error code

enum{

SIX\_BOX\_SENSOR\_TIMEOUT = -110,//超时

SIX\_BOX\_SENSOR\_CRC\_ERROR = -111,//CRC错误

SIX\_BOX\_SENSOR\_DATA\_UNUSUAL = -112,//数据异常

};

struct sensor\_data{

uint8\_t type;

uint8\_t data\_H;

uint8\_t data\_L;

};

这些是我这边.h中定义的宏 ，就和STM32里面那个IOT-sensor.h一样