## jsonAnalysis文件函数说明

## 1.函数原型：SJON\_CMD\_TYPE\*JsonCmdAnalysis(char \*Payload);

1.1功能:解析json串格式的指令，解析结果以SJON\_CMD\_TYPE\*返回。

1.2输入项：char \* Payload 。被解析的json串的地址

1.3输出项：SJON\_CMD\_TYPE \*。输出为json格式数据解析后的结果。cmd成员用于存放解析出来的指令码，如果cmd=CMD\_Error,则表示指令解析出错。解析结果 根据解析出来的指令码来确定解析的数据是从StringType，BleSleepType，BleMqttSetType中的哪一个获得。如数据是一个字符串则从StringType中读取，如果指令是‘设置休眠时间命令’则从 BleSleepType中读取，如果指令是‘1.15设备mqtt通讯参数设置’则从BleMqttSetType中读取

## **typedef enum**

## **{**

## **SET\_BLE\_PRW=0, //设置蓝牙发射功率命令**

## **SET\_BLE\_SEARCH\_INTERVAL, //设置蓝牙搜索间隔命令**

## **SET\_BLE\_SEARCH\_WINDOW, //设置蓝牙搜索窗口时间命令**

## **SET\_BLE\_REPORT\_INTERVAL, //设置蓝牙信息上报间隔命令**

## **SET\_BLE\_BATTERY\_VOLTAGE, //设置电池电压命令**

## **SET\_BLE\_BATTERY\_CURRENT, //设置电池电流命令**

## **SET\_BLE\_BATTERY\_MAX\_VOLTAGE,//设置外接电源电压上限命令**

## **SET\_BLE\_BATTERY\_MAX\_CURRENT,//设置外接电源电流上限命令**

## **SET\_BLE\_SLEEP\_TIME, //设置休眠时间命令**

## **SET\_BLE\_RESET, //设备复位命令**

## **SET\_BLE\_MQTT\_PARAMETER, //设备mqtt通讯参数设置**

## **CMD\_Error, //指令错误**

## **}E\_S2C\_CMD; //cmd成员的取值**

## 

## **typedef struct**

## **{**

## **E\_C2S\_CMD cmd; // 指令码**

## **char \*StringType; //若解析的结果只有一个字符型参数则数据从这个指针中读出**

## **BLE\_SLEEP\_TIME BleSleepType; //若解析的指令SET\_BLE\_SLEEP\_TIME指令则从此成员读出**

## **BLE\_MQTT\_SET BleMqttSetType; //若解析的指令**SET\_BLE\_MQTT\_PARAMETER**指令则从此成员读出**

## **}SJON\_CMD\_TYPE; //解析后的json指令**

## 1.4流程逻辑：

## 2函数原型：void FreeAnalysisMem(SJON\_CMD\_TYPE \*pMem);

2.1功能:释放JsonCmdAnalysis()解析时为返回的数据开的内存。

2.2输入项:SJON\_CMD\_TYPE \*pMem 需要释放的解析结果所对应的指针

2.3输出项:无

2.4流程逻辑：

## 3函数原型：int CreateBleInfoList(void);

3.1功能:建立一个蓝牙信息的链表,该链表用于记录蓝牙信息，

## **typedef struct**

## **{**

## **char sn[50];**

## **char timestamp[50];**

## **char cycle[50];**

## **char num[50];**

## **}BLE\_LIST\_HEAD; //表头信息**

## **typedef struct BLE\_LIST**

## **{**

## **struct BLE\_LIST \*next, \*previous;**

## **BLE\_LIST\_HEAD \*ble\_list;**

## **BLE\_INFO\_TYPE Ble\_info;**

## **}JSON\_BLE\_CMD\_LINK\_NODE; //链表**

3.2输入项:无；

3.3输出项：-1创建失败可能的远因是内存分配失败。0:创建链表成功

2.4流程逻辑：

## 4函数原型：int SetBleInfoList(char \*pSN , char \*pTimestamp , char \*pCycle, char \*pNum)

4.1功能:设置蓝牙信息链表的表头

4.2输入项：

char \*pSN 该参数用于设置“sn”对象的值，给参数以指针的形式传递参数

char \*pTimestamp 该参数用于设置“timestamp”对象的值，给参数以指针的形式传递参数

char \*cycle该参数用于设置“cycle”对象的值，给参数以指针的形式传递参数

char \*num该参数用于设置“num”对象的值，给参数以指针的形式传递参数

4.3输出项：-2:设置失败,可能的原因是在CreateBleInfoList之前调用了该函数；0设置成功

4.4流程逻辑：

## 5函数原型： int AddBleInfoToList(char \*name , char \*mac , char \*rssi, char \*RAWdata )

5.1功能:给蓝牙信息中添加一组蓝牙信息

5.2输入项：

char \*name 该参数用于设置蓝牙名，给参数以指针的形式传递参数

char \*mac 该参数用于设置MAC地址，给参数以指针的形式传递参数

char \*rssi该参数用于设置rssi的值，给参数以指针的形式传递参数

char \*RAWdata 该参数用于设置设置报文的内容（报文长度小于256byte），参数以指针的形式传递参数

5.3输出项：-2:设置失败,可能的原因是在CreateBleInfoList之前调用了该函数；0设置成功

5.4流程逻辑：

## 6函数原型：int DeleteBleInfoToList(const char \* mac )

6.1功能: 从蓝牙信息中删除一组蓝牙信息。

6.2输入项：

char \* mac 需要删除的蓝牙的MAC地址，以字符穿形式表示。传入的参数为该字符串的指针。

6.3输出项：-2:设置失败,可能的原因是在CreateBleInfoList之前调用了该函数；-1:没有匹配的蓝牙; 0:删除成功

6.4流程逻辑:

## 7函数原型：char \* BleInfoToJson(void)

7.1功能: 将蓝牙链表中的信息转换为json串;

7.2输入项：无

7.3输出项：返回转换后的json串的指针，如果返回为NULL则表明转换失败，可能的原因是在CreateBleInfoList之前调用了该函数输出的json串在使用完成后需要调用 FreeJsonString()来释放json串所占用内存;

## 8函数原型：char \*BleHeartbeatToJson(char \* mac,

## char \*timestamp,

## char \*bat\_voltage,

## char \*bat\_current,

## char \*pwr\_voltage,

## char \*pwr\_current);

8.1功能: 将心跳信息转换为json串；

8.2输入项：

mac : mac地址码的指针

timestamp: 时间戳的指针

bat\_voltage: 电池电压字符串的指针

bat\_current: 电池电流字符串的指针

pwr\_voltage: 电源电压字符串的指针

pwr\_current: 电源电流字符串的指针

8.3输出项：输出的json串在使用完成后需要调用FreeJsonString()来释放json串所占用内存

8.4流程逻辑:

## 9函数原型：char \*AckForJsonCmd(char \*type, char \*sn)

9.1功能: 下发设置命令后，设备回复设置完成命令；

9.2输入项：

type: mac地址码的指针；

sn: 时间戳的指针；

9.3输出项：输出的json串在使用完成后需要调用FreeJsonString()来释放json串所占用内存

9.4流程逻辑:

## 10函数原型：void FreeJsonString(char \*pJsonStr)

10.1功能: 释放转换后的json串的内存；

10.2输入项：

type: mac地址码的指针；

sn: 时间戳的指针；

10.3输出项：输出的json串在使用完成后需要调用FreeJsonString()来释放json串所占用内存

10.4流程逻辑: