

开发指南

编制人	AndyGao	审核人	Sean	批准人	
产品名称		产品编号		文档编号	
会签日期			版本	0.5	

修改记录:

修改时间	修改记录	修改人	版本	备注
20160510	初建	AndyGao	V0.1	
20160602	增加 gizwits_prodect\gizwits_protocol 模块文件描述	AndyGao	V0.2	
20160612	增加目录树图片，修正函数介绍位置	AndyGao	V0.3	
20160909	适配末端工程加以修正	TerryLi	V0.4	
20170913	修改目录树图片，修改函数介绍	lfCui	V0.5	

目录:

1 文件介绍.....	4
2 API 介绍.....	5
void gizwitsInit(void).....	5
void gizwitsSetMode(uint8_t mode).....	5
void gizwitsHandle(dataPoint_t *dataPoint).....	5
int32_t gizwitsPassthroughData(uint8_t * data, uint32_t len).....	5
int8_t gizwitsEventProcess(eventInfo_t *info, uint8_t *data, uint32_t len).....	5
3 二次开发.....	6
配置入网.....	6
数据采集.....	6
事件处理.....	6
其他.....	6

1 文件介绍

app	
driver	
hal_key.c	//按键驱动程序
gen_misc.bat	
gen_misc.sh	//编译工具，执行./gen_misc.sh
Gizwits	
gizwits_product.c	//产品相关的处理函数，用户相关的函数等
gizwits_product.h	//gizwits_product.c的头文件
gizwits_protocol.c	//gizwits协议相关的处理模块，API的封装等
gizwits_protocol.h	//gizwits_protocol.c头文件，包括协议相关结构体，数据点相关结构体等
include	
driver	
hal_key.h	//hal_key.c头文件
ssl	
user	
user_main.c	//程序入口函数user_init()在文件，包括各模块的初始化，task创建等
Utils	
common.c	//工具函数
common.h	
dataPointTools.c	//数据点相关压缩、解压函数等
dataPointTools.h	
ringbuffer.c	//环形缓冲区模块
ringBuffer.h	
bin	
at	
at_sdio	
temp_by_dlttool	
upgrade	
user1.4096.new.6.bin	//编译生成的执行文件，烧录使用
include	
gagent_soc.h	//gagent接口头文件
ld	
lib	
libgagent.a	//gagent封装库文件
tools	
SoC_ESP8266_API介绍_V0.5.pdf	//使用说明

1. libgagent.a
该文件为机智云设备接入协议库文件,文件位于 lib 目录下。
2. gagent_soc.h
该文件为 libgagent.a 对应头文件,两个文件配合使用。
3. gizwits_product.c
该文件包括产品相关处理函数，如 gizwitsEventProcess(); 用户相关的初始化 userInit() 以及数据采集 userHandle()函数。
4. gizwits_product.h
该文件为 gizwits_product.c 的头文件，如 HARDWARE_VERSION、SOFTWARE_VERSION。
5. gizwits_protocol.c
该文件为 SDK API 接口函数定义文件。
6. gizwits_protocol.h
该文件为 gizwits_protocol.c 对应头文件,相关 API 的接口声明均在此文件中。
7. 其他文件
 - a) app/driver/hal_key.c
按键模块函数,实现了 2 个 key 的长短按键检测功能,使用 demo 见 user_main.c 文件。
 - b) app/include/driver/hal_key.h
hal_key.c 模块的头文件,声明相关接口函数。
 - c) app/user/user_main.c

Esp8266 程序入口函数所在文件,入口函数为 `void user_init(void)`。

2 API 介绍

`void gizwitsInit(void)`

`gizwits` 协议初始化接口

用户调用该接口可以完成 `Gizwits` 协议相关初始化（包括协议相关定时器、串口的初始化）。

`void gizwitsSetMode(uint8_t mode)`

参数 `mode[in]`:为输入型参数,仅支持 0,1 和 2,其他数据无效

参数为 0,恢复模组出厂配置接口,调用会清空所有配置参数,恢复到出厂默认配置。

配置模式切换接口,支持 `SoftAP` 和 `AirLink` 模式.参数为 1 进入 `SoftAP` 模式,参数为 2 进入 `AirLink` 模式。

`void gizwitsHandle(dataPoint_t *dataPoint)`

参数 `dataPoint[in]`:用户设备数据点

`gizwits` 数据点更新上报处理，用户调用该接口可以完成设备数据的变化上报。

`int32_t gizwitsPassthroughData(uint8_t *data, uint32_t len)`

参数 `data[in]`:输入的私有协议数据

参数 `len [in]`:输入的私有协议数据长度

`gizwits` 上报透传数据接口，用户调用该接口可以完成私有协议数据的上报。

`int8_t gizwitsEventProcess(eventInfo_t *info, uint8_t *data, uint32_t len)`

参数 `info[in]`:事件队列

参数 `data[in]`:协议数据

参数 `len [in]`:协议数据长度

用户数据处理函数,包括 `wifi` 状态更新事件和控制事件。

a) `Wifi` 状态更新事件

`WIFI_`开头的事件为 `wifi` 状态更新事件,`data` 参数仅在 `WIFI_RSSI` 有效,`data` 值为 `RSSI` 值,数据类型为 `uint8_t`,取值范围 0~7。

b) 控制事件

与数据点相关,本版本生成代码会打印相关事件信息，相关数值也一并打印输出，用户只需要做相应具体处理即可。

注意：SOC_ESP8266 关于浮点型数据打印尚存在问题，但是不影响数据的正确性，只影响打印输出，若需要请联系机智云工程师。

3 二次开发

配置入网

Esp8266 支持 SoftAp 和 AirLink 两种方式配置入网,相应接口为 `gizwitsSetMode()`,本版软件采用按键的方式,相关代码参考 `user_main.c` 文件的 `key` 相关操作。

另外,可以通过 `gizwitsSetMode()`接口复位模组，恢复默认出厂设置。

数据采集

使用自动化工具生成的 SOC 代码默认给用户开启了 `userTimer` 定时器，定时周期为 1s，并且在 `userHandle()`函数中以伪代码形式提示用户在此处获取需要上报的数据点数据。用户只关心如何采集数据即可。数据点的初始化见 `userInit()`函数，待用户完善。`userHandle()`和 `userInit()`函数均在 `gizwits_product.c` 文件中。特别提醒，该定时器周期为 1s，需要针对不同的需求，用户调整数据点数据的采集周期。

事件处理

数据点方式将转换成数据点事件，开发者只需要在 `gizwits_product.c` 文件的 `gizwitsEventProcess()`相应事件下作具体处理即可，

透传方式数据下达后会生成 `SIG_PASSTHROUGH` 信号,内容和长度见注释提示。

其他

Wifi 状态

参考接口 `gizwitsEventProcess()`,本版软件已经将 `wifi` 状态数据转换成了 `event`,开发者仅关注相应事件即可。