

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 | 0.3 | |

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 | |

目录:

| | |
|---|---|
| 1 文件介绍..... | 4 |
| 2 API 介绍..... | 5 |
| void gizConfigReset(void)..... | 5 |
| void gizSetMode(uint8_t mode)..... | 5 |
| int32_t gizReportData(uint8_t action, uint8_t *data, uint32_t len)..... | 5 |
| uint32_t gizGetTimeStamp(void)..... | 5 |
| void gizEventProcess(event_info_t *info, uint8_t *data)..... | 6 |
| 3 二次开发..... | 6 |
| 配置入网..... | 6 |
| 数据上报..... | 6 |
| 数据接收..... | 6 |
| 其他..... | 7 |

1 文件介绍

| | | | |
|----|--|----------------------|---|
| 1 | | app | |
| 2 | | driver | |
| 3 | | hal_key.c | //按键驱动程序 |
| 4 | | gen_misc.bat | |
| 5 | | gen_misc.sh | //编译工具，执行./gen_misc.sh |
| 6 | | Gizwits | |
| 7 | | gizwits_product.c | //产品相关的处理函数，如gizEventProcess() |
| 8 | | gizwits_product.h | //gizwits_product.c头文件，主要定义软硬件版本号 |
| 9 | | gizwits_protocol.c | //gizwits协议相关的处理模块，API的封装等 |
| 10 | | 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 |
| 11 | | include | |
| 12 | | driver | |
| 13 | | hal_key.h | //hal_key.c头文件 |
| 14 | | ssl | |
| 15 | | user | |
| 16 | | user_main.c | //程序入口函数user_init()所在文件，包括各模块的初始化，task创建等 |
| 17 | | bin | |
| 18 | | at | |
| 19 | | 1024+1024 | |
| 20 | | 512+512 | |
| 21 | | noboot | |
| 22 | | _temp_by_dlttool | |
| 23 | | upgrade | |
| 24 | | user1.4096.new.6.bin | //编译生成的执行文件，烧录使用 |
| 25 | | include | |
| 26 | | gagent_external.h | //gagent接口头文件 |
| 27 | | ld | |
| 28 | | lib | |
| 29 | | libgagent.a | //gagent封装库文件 |
| 30 | | tools | |
| 31 | | 开发指南V0.1.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_external.h

该文件为 libgagent.a 对应头文件,两个文件配合使用.

3. gizwits_product.c

该文件为产品相关处理函数，如 gizEventProcess().

4. gizwits_product.h

5. 该文件为 gizwits_product.c 的头文件，如 HARDWARE_VERSION、SOFTWARE_VERSION.

6. gizwits_protocol.c

该文件为 SDK API 接口函数定义文件,

7. 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.

8. 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 gizwitsInit()函数.

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 gizConfigReset(void)

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.

void gizSetMode(uint8_t mode)

参数 mode,为输入型参数,仅支持 1 和 2,其他数据无效.

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.

int32_t gizReportData(uint8_t action, uint8_t *data, uint32_t len)

数据上报接口,参数 action 代表上报类型,参数 data 和 len 用于传递上报数据和大小.

action 描述:

ACTION_REPORT_DEV_STATUS 设备上报传感器状态,如温度\湿度.

ACTION_D2W_PASSTHROUGH 设备上报透传数据.

uint32_t gizGetTimeStamp(void)

获取网络时间戳接口,返回值为当前时间的秒数.

void gizEventProcess(event_info_t *info, uint8_t *data)

参数 info[in]:描述事件个数及事件值.

参数 data[in]:部分事件数据参数传递.

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,本版软件暂未做数据点解析,如需此部分请联系机智云.

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 [gizSetMode](#),本版软件采用按键的方式,相关代码参考 user_main.c 文件的 line 68/75.

数据上报

数据上报接口参考 [gizReportData](#),本版软件采用定时上报的方式演示此功能,相关代码参考 user_main.c 文件的 line 53,void user_handle(void)函数.

数据接收

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 gizEventProcess()处理即可,参考 line 47~106.

透传方式数据下达后会生成 SIG_PASSTHROUGH 信号,实现相关代码位于 gizwits.c line 596~603.

其他

Wifi 状态

参考接口 [gizEventProcess](#),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.

获取时间戳

接口参考 [gizGetTimeStamp](#),相关实现代码请参考文件 gizwits_protocol.c line550.