

开发指南

编制人	TerryLi	审核人	AndyGao	批准人	
产品名称		产品编号		文档编号	
会签日期			版本	0.1	

修改记录:

修改时间	修改记录	修改人	版本	备注
20170921	初建	TerryLi	V0.1	

目录:

1 文件介绍.....	4
2 API 介绍.....	4
void gizwitsInit(void).....	4
void gizwitsSetMode(uint8_t mode).....	5
void gizwitsGetNTP(void).....	5
void gizwitsHandle(dataPoint_t *dataPoint).....	5
int8_t gizwitsEventProcess(eventInfo_t *info, uint8_t *data,.....	5
uint32_t len).....	5
int32_t gizwitsPassthroughData(uint8_t *data, uint32_t len).....	6
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



重要文件解读：

1. gizwits_product.c

该文件为用户相关处理函数以及平台相关硬件初始化，如串口、定时器等，同时包含数据采集 userHandle()函数和数据点初始化 userInit()函数。

2. gizwits_product.h

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

4. gizwits_protocol.c

该文件为 SDK API 接口函数定义文件。

5. 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。

6. 其他文件

a) User/main.c

MCU 程序入口函数所在文件,入口函数为 main(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]: WIFI_MODE_TYPE_T 枚举值

- 参数为 WIFI_RESET_MODE, 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。
- 参数为 WIFI_SOFTAP_MODE 或 WIFI_AIRLINK_MODE, 配置模式切换接口, 支持 SoftAP 和 AirLink 模式。参数为 WIFI_SOFTAP_MODE 时配置模组进入 SoftAp 模式, 参数为 WIFI_AIRLINK_MODE 配置模组进入 AirLink 模式。
- 参数为 WIFI_PRODUCTION_TEST, 模组进入产测模式。
- 参数为 WIFI_NINABLE_MODE, 模组进入可绑定模式, 可绑定时间为 NINABLETIME(gizwits_protocol.h 中声明), 默认为 0, 表示模组永久可绑定。

void gizwitsGetNTP(void)

获取 NTP 时间接口。

用户调用该接口可以获取当前网络时间, MCU 发起请求, 模组回复后将产生 WIFI_NTP 事件, 用户可在 gizwitsEventProcess 函数中进行相应处理。

void gizwitsHandle(dataPoint_t *dataPoint)

参数 dataPoint[in]: 用户设备数据点。

该函数中完成了相应协议数据的处理及数据上报的等相关操作。

int8_t gizwitsEventProcess(eventInfo_t *info, uint8_t *data,

uint32_t len)

参数 info[in]: 事件队列

参数 data[in]: 数据

参数 len [in]: 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, 用户只需要做命令的具体执行即可。

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

参数 data[in]:数据

参数 len [in]:数据长度

用户调用该接口可以完成私有协议数据的上报。

3 二次开发

配置入网

支持 AirLink 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 Hal_key/Hal_key.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

该工程代码默认在 Gizwits/gizwits_product.c 文件中 `userHandle()`函数实现传感器数据采集，并且该函数在 `while` 循环执行，原则上用户只需要关心如何采集数据。**特别提醒，默认 `while` 循环执行速度较快，需要针对不同的需求，用户可调整数据点数据的采集周期和接口实现位置，预防由于传感器数据采集过快引发的不必要的问题。**数据点变量的初始化见 Gizwits/gizwits_product.c 文件 `userInit` 函数，待用户完善。

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。

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。