

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	0.6	

修改记录:

修改时间	修改记录	修改人	版本	备注
20160909	初建	TerryLi	V0.1	
20161030	增加模组产测、绑定和获取 NTP 时间接口	TerryLi	V0.2	
20161228	增加透传通道支持	TerryLi	V0.3	
20170913	修改文件目录结构	lfCui	V0.4	
20171106	适配 STC15 内容，添加“平台性能限制说明”相关内容	True	V0.5	
20180523	文档排版	AndyGao	V0. 6	

目录:

1 文件介绍.....	4
2 API 介绍.....	4
void gizwitsInit(void).....	4
void gizwitsSetMode(uint8_t mode).....	4
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).....	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
4 平台性能限制说明.....	6

1 文件介绍

```
├── Gizwits
│   ├── gizwits_product.c //与用户数据点处理相关的程序
│   ├── gizwits_product.h //定义软硬件版本号
│   ├── gizwits_protocol.c //gizwits协议处理模块，提供对用户接口封装
│   └── gizwits_protocol.h //包括协议相关结构体
├── Hal
│   ├── Hal_key //按键驱动程序
│   └── Hal_Usart //串口驱动程序
├── Project
│   └── STC15F2K60S2.uvproj //工程入口文件
├── User
│   └── main.c //程序入口文件
└── Utils //工具文件夹
    ├── common.c
    ├── common.h
    ├── dataPointTools.c
    ├── dataPointTools.h
    ├── ringbuffer.c
    └── ringBuffer.h
```

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。

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 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 `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

该工程代码默认在 `userHandle()` 中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。

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。

4 平台性能限制说明

由于 8051 等型号 MCU 片上资源有限（STC15F2K60S2 SRAM 空间大小：2K = 2048 字节），当云端定义的产品中数据点较多时会导致编译后的程序剩余内存过少，进而导致串口数据解析异常、程序卡死等情况，故开发者需确认已创建的数据点长度以及内存的情况。

数据点长度建议少于 20 个字节，对应的 RAM 的使用率需在 60% 以内。

数据点长度查看方式：在开发者中心下载《xxx- 机智云独立 MCU 方案接入通信协议文档》，在“3.2 WiFi 模组控制设备”中可以看到“`attr_flags`”+“`attr_vals`”之和即为数据点长度的总大小（字节）：

3.2 WiFi模组控制设备

WiFi模组=>MCU

序号	字段名称	字节长度(Byte)	内容说明
1	固定包头	2	0xFFFF
2	包长度	2	len(命令...校验和)
3	命令	1	0x03
4	包序号	1	0x##
5	flags	2	0x0000
6	action	1	0x01
7	attr_flags	(2B)	是否设置标志位
8	attr_vals	(18B)	设置数据值
9	校验和	1	0x##

查询内存使用情况的方式（STC15F2K60S2 平台的 keil5 工程为例），工程编译后显示如下：

Build Output

```
CALLER1: ?PR?UART1_RCV?HAL_UART
CALLER2: ?C_C51STARTUP
*** WARNING L15: MULTIPLE CALL TO SEGMENT
SEGMENT: ?PR?RBCANREAD?RINGBUFFER
CALLER1: ?PR?UART1_RCV?HAL_UART
CALLER2: ?C_C51STARTUP
*** WARNING L15: MULTIPLE CALL TO SEGMENT
SEGMENT: ?PR?RBCAPACITY?RINGBUFFER
CALLER1: ?PR?UART1_RCV?HAL_UART
CALLER2: ?C_C51STARTUP
Program Size: data=21.5 xdata=1167 code=30735
creating hex file from "STC15F2K60S2"...
"STC15F2K60S2" - 0 Error(s), 83 Warning(s).
Build Time Elapsed: 00:00:04
```

其中“xdata”即为程序已使用的SRAM（STC15F2K60S2平台总SRAM = 2048字节），经测试建议用户将此参数限制在1200字节以内（占总RAM的60%以内），超过此范围，程序运行过程中就会因为剩余内存空间过少，导致串口数据解析异常、程序卡死等情况。**请开发者注意！**