

智嵌物联 4G 系列产品二次开发说明书



深圳总部

地址：广东省深圳市宝安区新桥街道新桥社区

新和大道 6-18 号 1203

网址：www.zhiqwl.com

电话：0755-23203231

北京办事处

地址：北京市房山城区德润街6号院8号楼3层

电话：18210365439



天猫店铺



淘宝店铺



京东店铺



微信公众号



公司官网

修订历史

版本	日期	原因
V1.0	2021.06.06	发布
V1.1	2021.12.29	增加部分函数接口
V1.2	2022.02.08	增加二次开发 demo 例程

目 录

1. 二次开发流程	1
1.1 开发流程	1
1.2 智嵌物联 4G 产品线命名规则	2
1.2.1 智嵌物联 4G RTU 产品线命名规则如图 1.1 所示	2
1.2.2 智嵌物联 4G DTU 产品线命名规则如图 1.2 所示	2
1.2.3 相关资料下载:.....	2
2. 搭建开发环境	3
2.1 硬件连接并对设备参数进行配置.....	3
2.1.1 硬件连接.....	3
2.1.2 下载二次开发资料.....	3
2.1.3 配置设备网络参数.....	4
2.2 用户编程	5
2.2.1 安装 Source Insight.....	5
2.2.2 新建工程.....	5
2.3 编译生成 app.bin 文件	7
2.4 下载 app.bin 文件	7
2.5 异常情况处理.....	8
2.5.1 下错程序致使设备无法正常下载程序.....	8
3. 函数库说明	9
3.1.1 zq_rcv_cb	9
3.1.2 zq_set_rcv_cb.....	9
3.1.3 zq_get_net_status.....	9
3.1.4 zq_get_serverA_status.....	9
3.1.5 zq_get_serverB_status.....	10
3.1.6 zq_get_mqtt_status	10
3.1.7 zq_iot_network_get_CSQ	10
3.1.8 zq_serverA_send.....	10
3.1.9 zq_serverB_send.....	11
3.1.10 zq_mqtt_send	11
3.1.11 zq_uart_write.....	11
3.1.12 zq_get_Y.....	11
3.1.13 zq_get_X.....	12
3.1.14 zq_set_Y	12
3.1.15 zq_get_ADC.....	12
3.1.16 zq_sleep	13
3.1.17 zq_create_timer	13
3.1.18 zq_start_timer.....	13
3.1.19 zq_stop_timer	13
3.1.20 void zq_restart	14
3.1.21 int zq_msg_send	14
3.1.22 int zq_flash_erase	14
3.1.23 int zq_flash_write.....	14

3.1.24	int zq_flash_read.....	15
3.1.25	void zq_dbg_uart_init.....	15
3.1.26	int zq_dbg_uart_read.....	15
3.1.27	int zq_dbg_uart_write	15
4.	二次开发 demo 介绍	17
4.1.1	demo_cjson.....	17
4.1.2	demo_gpio	17
4.1.3	demo_http	17
4.1.4	demo_modbus_poll.....	17
4.1.5	demo_mqtt	17
4.1.6	demo_ntp.....	17
4.1.7	demo_serverA.....	18
4.1.8	demo_serverB.....	18
4.1.9	demo_timer	18
	销售网络.....	19

1. 二次开发流程

1.1 开发流程

1 搭建硬件环境



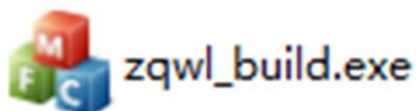
2 用户编程



按照用户的需求，编写程序，并保存



3 编译生成app.bin文件



编译上一步骤的Project，
并生成app.bin文件



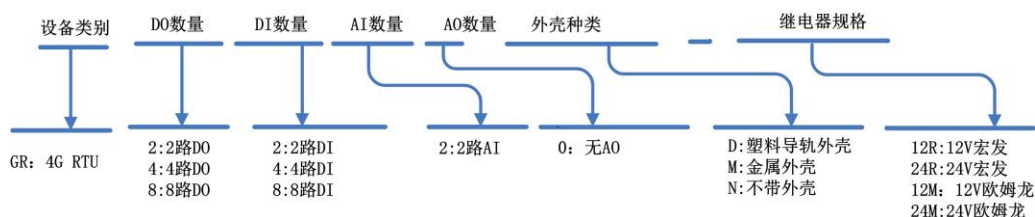
4 下载app.bin文件



① 鼠标点击以上步骤中的图片即可跳转到相应章节。

1.2 智嵌物联 4G 产品线命名规则

1.2.1 智嵌物联 4G RTU 产品线命名规则如图 1.1 所示

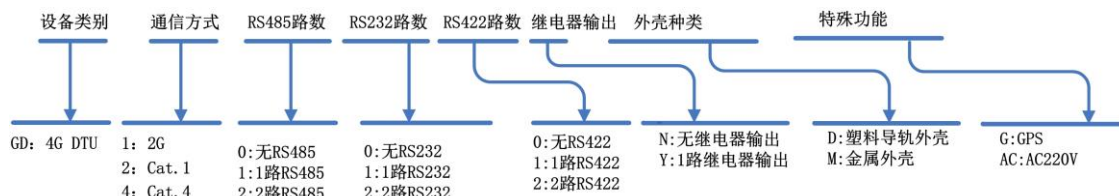


如: **ZQWL-GR8800M-24M**

4g RTU、8路24V欧姆龙继电器输出, 8路DI 金属外壳

图 1.1 4G RTU 命名规则

1.2.2 智嵌物联 4G DTU 产品线命名规则如图 1.2 所示



如: **ZQWL-GD2110YM**

4G DTU CAT.1、1路RS485、1路RS232、1路继电器输出 金属外壳

图 1.2 智嵌 4G DTU 命名规则

1.2.3 相关资料下载:

ZQWL-GR0400M 产品规格书下载: [点击下载](#)

ZQWL-GR1200M 产品规格书下载: [点击下载](#)

ZQWL-GR2220M 产品规格书下载: [点击下载](#)

ZQWL-GR4420M 产品规格书下载: [点击下载](#)

ZQWL-GR8800M 产品规格书下载: [点击下载](#)

ZQWL-1DX1S8 产品规格书下载: [点击下载](#)

ZQWL-GM100 产品规格书下载: [点击下载](#)

ZQWL-GD2110YM 产品规格书下载: [点击下载](#)

ZQWL- GD2100YD 产品规格书下载: [点击下载](#)

ZQWL- GD2100ND-AC 产品规格书下载: [点击下载](#)

二次开发资料下载: [点击下载](#)

配置工具下载: [点击下载](#)

2. 搭建开发环境

2.1 硬件连接并对设备参数进行配置

2.1.1 硬件连接



图 2.1 硬件连接

1. 将 SIM 卡插到设备的 SIM 卡插槽中，并将 4G 天线正确安装；
2. 设备的 RS485 接口通过“USB 转 485 工具”连接到电脑。连接示意图如图 2.1 所示。
3. 设备上电；

2.1.2 下载二次开发资料

在官网或淘宝链接内下载“4G RTU&DTU 二次开发资料”（下载地址：[点击下载](#)），解压后，有 3 个文件夹，如图 2.2 所示。

名称	修改日期	类型
4G DTU&RTU配置软件	2021/6/21 星期...	文件夹
project	2021/6/21 星期...	文件夹
sourceInsight_downcc	2021/6/21 星期...	文件夹

配置软件：用于下载app.bin文件、配置设备参数
二次开发SDK包、编译软件
Source Insight软件安装包

图 2.2 压缩包文件

1. 4G RTU&DTU 配置软件 4G DTU&RTU配置软件.exe

配置软件用于配置设备的网络参数、串口参数、使能二次开发功能以及下载用户编译生成的 app.bin 文件到设备中。

2. project 文件夹

project 文件夹内的文件是二次开发的 SDK 包，文件如图 2.3 所示。用户只需关心“project”文件，其他文件用户无需关心。

名称	修改日期	类型	大小
api	2021/5/7 星期五 ...	文件夹	
app	2021/5/7 星期五 ...	文件夹	
cmake	2021/5/7 星期五 ...	文件夹	
components	2021/5/7 星期五 ...	文件夹	
core	2021/6/21 星期...	文件夹	
demo	2021/5/7 星期五 ...	文件夹	
hex	2021/6/21 星期...	文件夹	
ldscripts	2021/5/7 星期五 ...	文件夹	
out	2021/6/21 星期...	文件夹	
prebuilts	2021/5/7 星期五 ...	文件夹	
project	2021/6/21 星期...	文件夹	
tools	2021/5/7 星期五 ...	文件夹	
.gitignore	2021/2/7 星期日 ...	GITIGNORE 文件	1 KB
CMakeLists.txt	2021/2/7 星期日 ...	文本文档	19 KB
config.cmake	2021/2/7 星期日 ...	CMAKE 文件	1 KB
LICENSE	2021/2/7 星期日 ...	文件	2 KB
README.md	2021/2/7 星期日 ...	MD 文件	3 KB
start.gdb	2021/6/21 星期...	GDB 文件	1 KB

用户只需关心此文件，其他文件夹无需关心，该文件内有demo、编译工具及编译生成的app.bin文件也在此路径内

图 2.3 project 文件夹内容

project 文件：文件夹内有 demo 文件、编译工具以及编译后生成的 app.bin 文件。

可发布说明书 > 4G系列 > 二次开发资料 > 4GRTU&DTU二次开发资料 > project_v23044 > project > project			
名称	修改日期	类型	大小
demo_json	2022/1/25 星期...	文件夹	
demo_gpio	2022/1/25 星期...	文件夹	
demo_http	2022/1/25 星期...	文件夹	
demo_modbus_poll	2022/1/25 星期...	文件夹	
demo_mqtt	2022/1/25 星期...	文件夹	
demo_ntp	2022/1/25 星期...	文件夹	
demo_old	2022/1/25 星期...	文件夹	
demo_serverA	2022/1/25 星期...	文件夹	
demo_serverB	2022/1/25 星期...	文件夹	
demo_timer	2022/1/25 星期...	文件夹	
inc	2022/1/26 星期...	文件夹	
myapp	2022/1/26 星期...	文件夹	
out	2022/1/25 星期...	文件夹	
build.exe	2022/1/25 星期...	应用程序	12,017 KB
dtu-v136.exe	2022/1/25 星期...	应用程序	3,713 KB

Demo文件，用户可将相关的功能代码复制到自己的代码中，快速实现demo的功能，更多demo持续更新

为方便管理用户代码，可自建文件夹，将用户的app.c文件放到该文件内

头文件

编译工具

3. sourceInsight_downcc 文件夹

sourceInsight_downcc 文件夹内有 sourceInsight 软件的安装文件。

2.1.3 配置设备网络参数

打开【4G RTU&DTU 配置软件】，对设备的网络参数、串口参数进行配置，勾选使能“用户代码（二次编程）”，保存参数后，重启设备，具体步骤如图 2.4 所示。



图 2.4 配置设备参数

2.2 用户编程

2.2.1 安装 Source Insight

双击文件中的“setup.exe”，按照 Source Insight 安装说明进行操作。

2.2.2 新建工程

1. 在 Source Insight 窗口的顶部工具栏上依次点击 Project > New Project, 即可开始创建一个新的项目了。

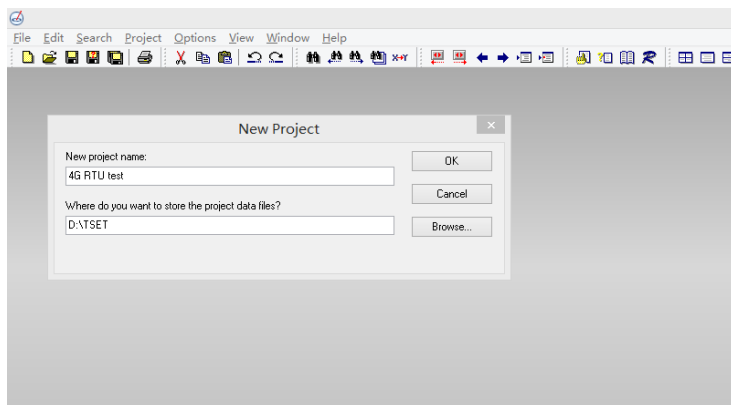


图 2.5 Source Insight 新建工程

2. 在红框部分选择需要打开的源码所在的路径，即二次开发资料压缩包的解压路径-> project -> project->。

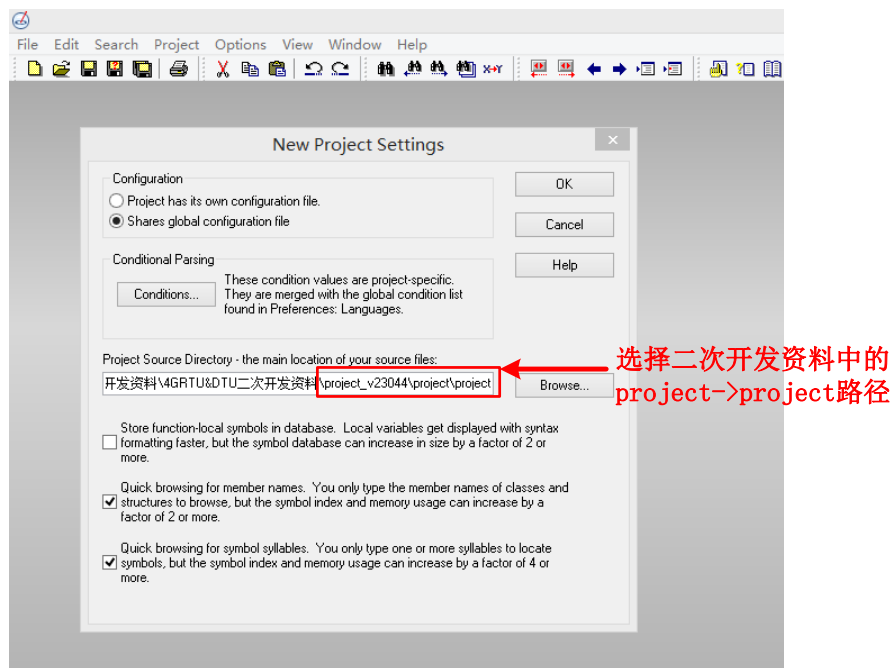


图 2.6 选择源码所在路径

3. 向工程中添加文件：app.c 文件和 user_code.h 文件

按图 2.7 所示步骤操作，将 app.c 文件和 user_code.h 文件添加到工程中。

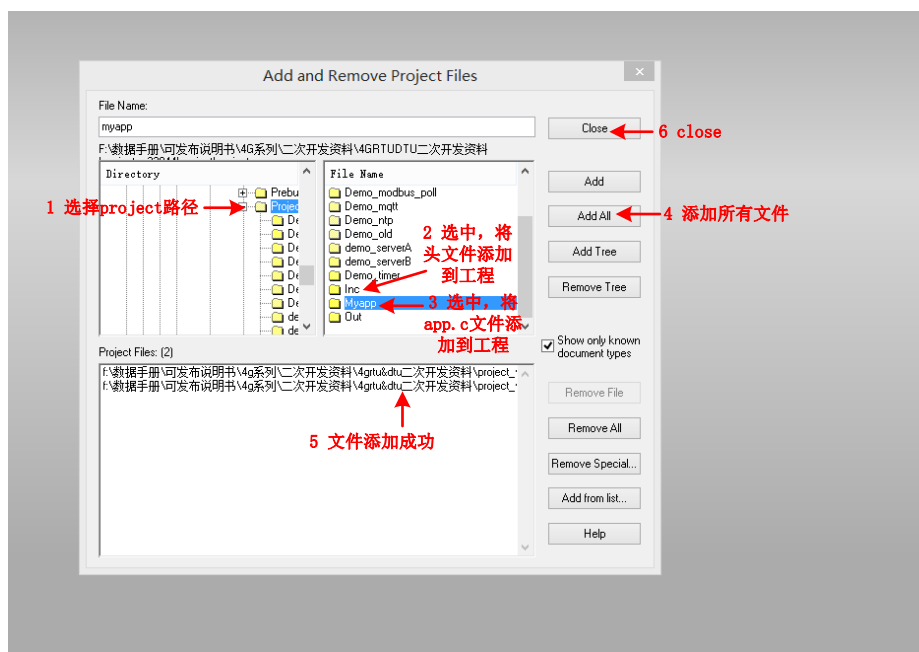


图 2.7 向工程中添加源文件

① 用户也可直接将文件拖拽到 Source Insight 窗口的灰色区域。

4. 编写用户程序

用户按照自己的需求在主程序中编写自己的逻辑程序，编写完成后，保存。之后就可以用编译工具对工程进行编译了。

3 编写完后，点击保存

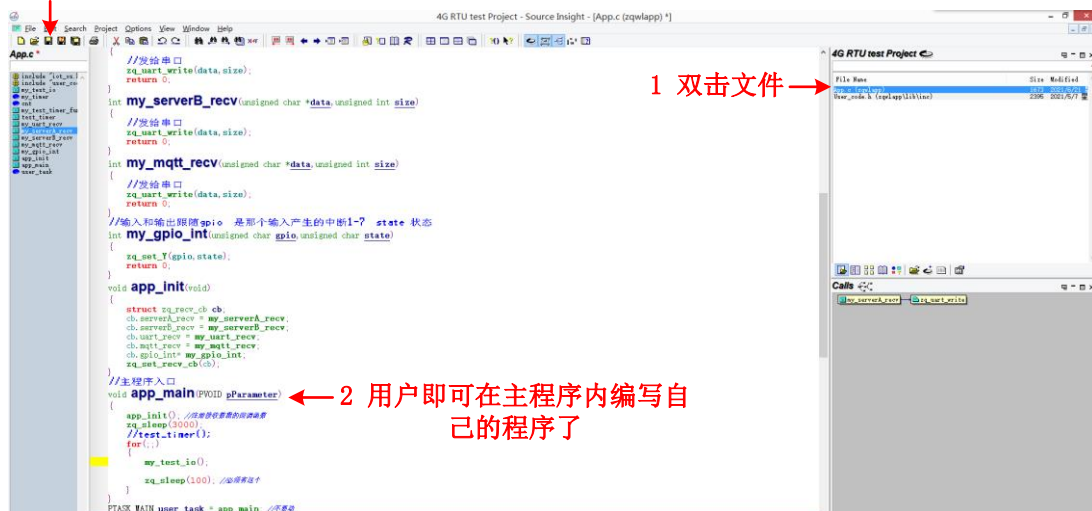


图 2.8 编写用户代码

2.3 编译生成 app.bin 文件

从二次开发资料中找到编译工具： build.exe（路径：二次开发包的解压文件->project ->project-> build.exe）。按照所示，选择对应 app.c 的工程文件，对工程进行编译。编译结束后，会生成 app.bin 文件。

5 编译生成的app. bin文件

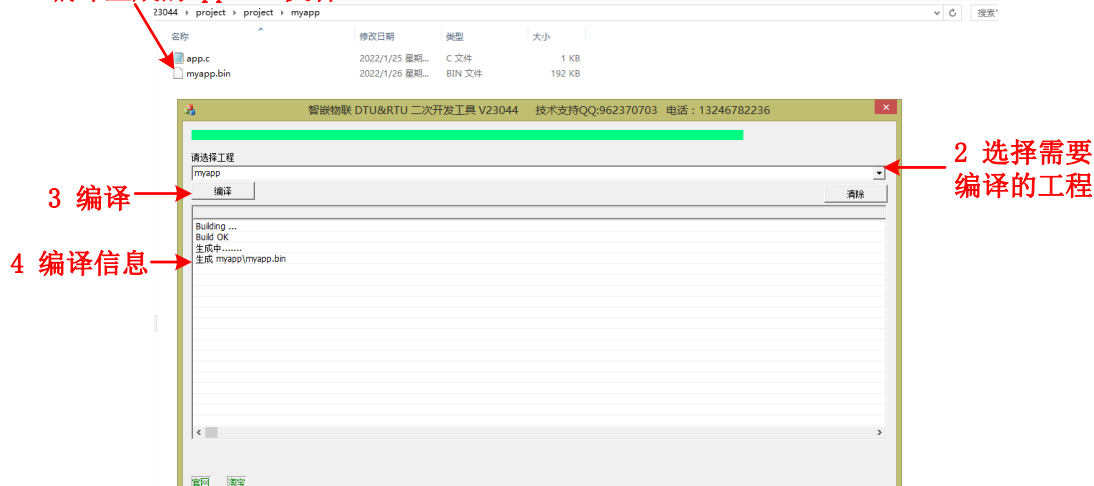


图 2.9 编译生成 app.bin 文件

2.4 下载 app.bin 文件

将上一步编译生成的 app.bin 文件，下载到设备中，具体步骤如图 2.10 所示。

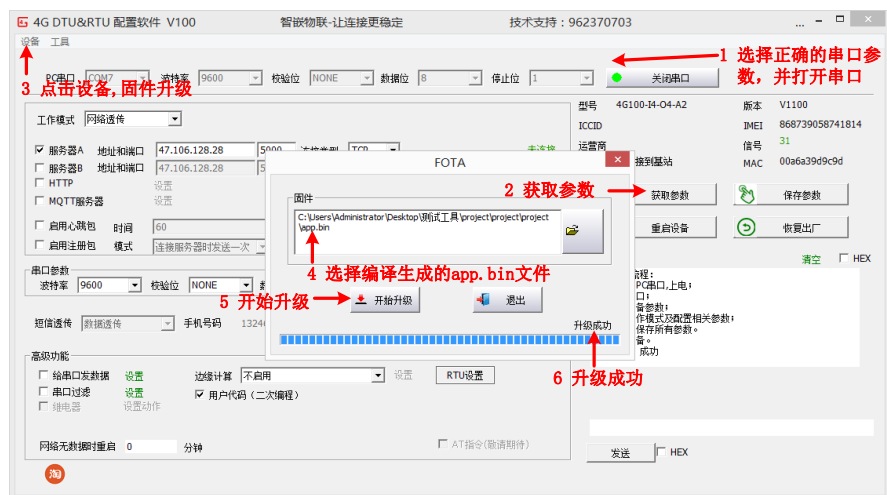


图 2.10 下载 app.bin 文件

至此，用户编写的程序已经下载到设备中了。等待 15 秒左右，设备即可执行用户的代码了。

2.5 异常情况处理

2.5.1 下错程序致使设备无法正常下载程序

如果用户的程序存在严重的 BUG，导致设备无法正常下载程序，此时可进行以下操作使设备正常。

1. 短按设备的“RESET”按键，使设备复位（或重新给设备上电）。
2. 待“RUN”运行灯开始闪烁时，点击配置软件上的【开始升级】按钮。注意要在设备未运行用户代码前点击【开始升级】（即从设备复位到点击【开始升级】按钮，间隔不能超过 10 秒）。

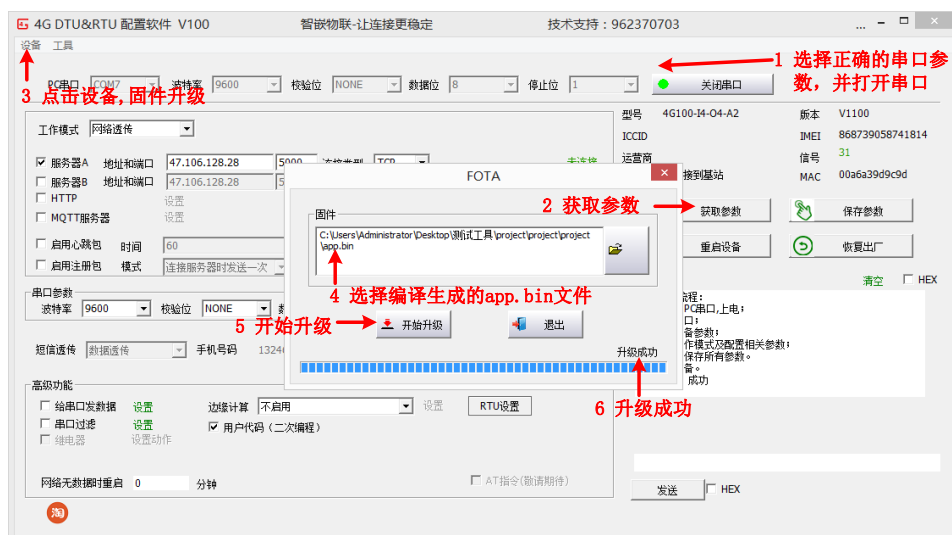


图 2.11 升级固件

3. 若经过以上两步仍然不能升级成功，需对设备进行深度恢复出厂，具体方法为：将设备断电，然后按住“RESET”按键，再将设备上电，等 40 秒后松开按键。

3. 函数库说明

3.1.1 zq_rcv_cb

描述

创建回调函数结构体

```
struct zq_rcv_cb{
    int (*serverA_rcv)(unsigned char *data,unsigned int size);
    int (*serverB_rcv)(unsigned char *data,unsigned int size);
    int (*mqtt_rcv)(unsigned char *data,unsigned int size);
    int (*uart_rcv)(unsigned char *data,unsigned int size);
    int (*gpio_int)(unsigned char gpio,unsigned char state);
};
```

3.1.2 zq_set_rcv_cb

描述

设置接收回调函数，注意不要在回调函数中做费时的操作

```
void zq_set_rcv_cb(struct zq_rcv_cb cb);
```

参数

回调函数

返回值

无

3.1.3 zq_get_net_status

描述

获取网络是否注册到基站

```
unsigned char zq_get_net_status(void);
```

参数

无

返回值

- 0 未注册到
- 1 注册到基站

3.1.4 zq_get_serverA_status

描述

获取服务器 A 的连接状态

```
unsigned char zq_get_serverA_status(void);
```

参数

无

返回值

- 0 未连接
- 1 已成功连接

3.1.5 zq_get_serverB_status

描述

获取服务器 B 的连接状态

```
unsigned char zq_get_serverB_status(void);
```

参数

无

返回值

- 0 未连接
- 1 已成功连接

3.1.6 zq_get_mqtt_status

描述

获取 mqtt 的连接状态

```
unsigned char zq_get_mqtt_status(void);
```

参数

无

返回值

- 0 未连接
- 1 已成功连接

3.1.7 zq_iot_network_get_CSQ

描述

获取设备的 4G 信号强度 范围 0-31

```
unsigned char zq_iot_network_get_CSQ(void);
```

参数

无

返回值

0-31 设备的信号强度大于 15 即可认为 4G 信号很好

3.1.8 zq_serverA_send

描述

向服务器 A 发送数据

```
int zq_serverA_send(unsigned char *data,unsigned int size);
```

参数

- data 要发送的数据
- size 发送数据的大小

返回值

已经发送的数据大小

3.1.9 zq_serverB_send

描述

向服务器 B 发送数据

```
int zq_serverB_send(unsigned char *data,unsigned int size);
```

参数

data 要发送的数据

size 发送数据的大小

返回值

已经发送的数据大小

3.1.10 zq_mqtt_send

描述

向 MQTT 服务器发送数据

```
int zq_mqtt_send(unsigned char *data,unsigned int size);
```

参数

data 要发送的数据

size 发送数据的大小

返回值

已经发送的数据大小

3.1.11 zq_uart_write

描述

向串口发送数据

```
int zq_uart_write(unsigned char *data,unsigned int size);
```

参数

data 要发送的数据

size 发送数据的大小

返回值

已经发送的数据大小

3.1.12 zq_get_Y

描述

获取 DO 输出 Y 的状态

```
char zq_get_Y(unsigned char idx);
```

参数

idx 取值 0-7，分别代表第 1 路~第 8 路 DO 输出

返回值

- 0 输出低电平（继电器常开触点与公共端断开）
- 1 输出高电平（继电器常开触点与公共端闭合）

3.1.13 zq_get_X

描述

获取 DI 输入 X 的状态

```
char zq_get_X(unsigned char idx);
```

参数

idx 取值 0-7，分别代表第 1 路~第 8 路 DI 输入

返回值

- 0 DI 输入接口无有效信号
- 1 DI 输入接口存在有效信号

3.1.14 zq_set_Y

描述

设置 DO 输出 Y 的状态

```
void zq_set_Y(unsigned char idx,unsigned char status);
```

参数

idx 取值 0-7，分别代表第 1 路~第 8 路 DO 输出

status 取值 0、1， 0：输出低电平（继电器常开触点与公共端断开）；
1：输出高电平（继电器常开触点与公共端闭合）。

返回值

- 0 输出低电平（继电器常开触点与公共端断开）
- 1 输出高电平（继电器常开触点与公共端闭合）

3.1.15 zq_get_ADC

描述

获取模拟量 ADC 的值

```
unsigned int zq_get_ADC(unsigned char idx);
```

参数

idx 取值 0-1，分别代表第 1 路、第 2 路 ADC

返回值

adc 的值

① 模拟量计算公式：0~5V 档位： $V = \text{adc} * 10 / 7727$ 。 0~10V 档位： $V = \text{adc} * 20 / 7727$ 。
0~20mA 档位： $I = \text{adc} * 40 / 7727$

3.1.16 zq_sleep

描述

休眠

```
void zq_sleep(unsigned int ms);
```

参数

0~65535

返回值

无

3.1.17 zq_create_timer

描述

创建定时器

```
HANDLE zq_create_timer(PTIMER_EXPFUNC func, void* param);
```

参数

func 定时器执行函数

param 传送的参数

返回值

定时器的句柄

3.1.18 zq_start_timer

描述

启动定时器

```
unsigned char zq_start_timer(HANDLE t,unsigned int ms);
```

参数

t 定时器句柄

ms 定时器时间

返回值

0 失败

1 成功

3.1.19 zq_stop_timer

描述

停止定时器

```
unsigned char zq_stop_timer(HANDLE t);
```

参数

t 定时器句柄

返回值

0 失败

1 成功

3.1.20 void zq_restart

描述

重启设备

```
void zq_restart(void);
```

参数

无

返回值

无

3.1.21 int zq_msg_send

描述

发送短信

```
int zq_msg_send(unsigned char *msg,unsigned char *phone);
```

参数

msg: 短信内容, ascii 格式

phone : 要发送的电话号码

返回值

无

3.1.22 int zq_flash_erase

描述

擦除 64kb 的存储空间, 注意请勿频繁擦除

```
int zq_flash_erase(void);
```

参数

无

返回值

无

3.1.23 int zq_flash_write

描述

写数据到 flash

```
int zq_flash_write(unsigned int addr,unsigned char *data,unsigned int size);
```

参数

addr: 写入数据的起始地址, 范围: 0-0x10000

data: 要写入的数据

size: 数据大小

返回值

-1 失败

大于 0 的数值 实际写入的数据大小

3.1.24 int zq_flash_read

描述

读 flash 数据

```
int zq_flash_read(unsigned int addr,unsigned char *data,unsigned int size);
```

参数

addr: 读取数据的起始地址, 范围: 0-0x10000

data: 要读取的数据

size: 要读的数据大小

返回值

-1 失败

大于 0 的数值 实际读取的数据大小

3.1.25 void zq_dbg_uart_init

描述

初始化调试串口参数

```
void zq_dbg_uart_init(unsigned int bitrate,void *rx_cb);
```

参数

bitrate: 调试串口波特率, 1200,2400, 4800,9600,14400,19200,28800,38400,57600,76800,
115200,230400,460800,576000,921600

rx_cb: 中断接受函数, 不使用可以设置为 NULL

返回值

调试串口接收的数据大小

3.1.26 int zq_dbg_uart_read

描述

调试串口接收数据, 调用前必须要先调用串口初始化函数 zq_dbg_uart_init(115200,NULL);

```
int zq_dbg_uart_read(unsigned char *rx_buf,unsigned int buf_size,unsigned int timeout_ms);
```

参数

rx_buf: 调试串口接收到的数据

buf_size: 调试串口接收到的数据大小

timeout_ms: 调试串口接收数据超时时间

返回值

调试串口接收的数据大小

3.1.27 int zq_dbg_uart_write

描述

调试串口发送数据，调用前必须要先调用串口初始化函数 `zq_dbg_uart_init(115200,NULL);`

```
int zq_dbg_uart_write(unsigned char *tx_buf,unsigned int tx_size);
```

参数

tx_buf: 调试串口发送的数据

tx_size: 调试串口发送的数据的大小

返回值

调试串口发送的数据大小

4. 二次开发 demo 介绍

为方便客户通过产品的二次开发功能，快速实现用户的需求，公司提供多种二次开发 demo。

名称	修改日期	类型	大小
demo_cjson	2022/1/26 星期...	文件夹	
demo_gpio	2022/1/26 星期...	文件夹	
demo_http	2022/1/26 星期...	文件夹	
demo_modbus_poll	2022/1/26 星期...	文件夹	
demo_mqtt	2022/1/26 星期...	文件夹	
demo_ntp	2022/1/26 星期...	文件夹	
demo_old	2022/1/26 星期...	文件夹	
demo_serverA	2022/1/26 星期...	文件夹	
demo_serverB	2022/1/26 星期...	文件夹	
demo_timer	2022/1/26 星期...	文件夹	
inc	2022/1/26 星期...	文件夹	
myapp	2022/1/26 星期...	文件夹	

更多demo，持续更新

4.1.1 demo_cjson

由于 c 语言中，没有直接的字典，字符串数组等数据结构，所以要借助结构体定义，处理 json。如果有对应的数据结构就方便一些，如 python 中用 `json.loads(json)` 就把 json 字符串转变为内建的数据结构处理起来比较方便。因此该 demo 提供了 json 格式数据处理的一种方法。

关于 Cjson 说明参考：<https://www.cnblogs.com/catgatp/p/6379955.html>

4.1.2 demo_gpio

智嵌物联的 4G RTU 系列提供了 8 个开关量输入、8 个开关量输出，用户可根据自己需要对这些 GPIO 进行逻辑编程。

该 demo 实现的功能是：设备的第一路继电器会闭合 3 秒、断开 3 秒，一直循环。

4.1.3 demo_http

该 demo 实现功能：实现设备串口数据与 http 服务器之间的双向透传，即：设备的串口收到的数据，直接发送给用户设置的 http 服务器；http 服务器发送给设备的数据，直接转发到设备的串口上。

4.1.4 demo_modbus_poll

该 demo 实现功能：设备主动轮询设备 RS485 总线上的设备，并将 RS485 总线上设备的应答数据上传到 mqtt 服务器上。

4.1.5 demo_mqtt

该 demo 实现功能：实现设备串口数据与 mqtt 服务器之间的双向透传，即：设备的串口收到的数据，直接发送给用户设置的 mqtt 服务器；mqtt 服务器发送给设备的数据，直接转发到设备的串口上。

4.1.6 demo_ntp

该 demo 实现功能：定时向网络上获得时间，并将时间数据发送到设备的串口上。

4.1.7 demo_serverA

该 demo 实现功能：实现设备串口数据与服务器 A 之间的双向透传，即：设备的串口收到的数据，直接发送给用户设置的服务器 A；服务器 A 发送给设备的数据，直接转发到设备的串口上。

4.1.8 demo_serverB

该 demo 实现功能：实现设备串口数据与服务器 B 之间的双向透传，即：设备的串口收到的数据，直接发送给用户设置的服务器 B；服务器 B 发送给设备的数据，直接转发到设备的串口上。

4.1.9 demo_timer

该 demo 实现功能：通过定时器功能，实现设备的第一路继电器会闭合 3 秒、断开 3 秒，一直循环。

销售网络

智嵌物联，让连接更稳定！

企业愿景：成为国内物联网设备首选品牌！

企业使命：为客户利益而努力创新，为推动工业物联网发展而不懈奋斗！

产品理念：稳定！稳定！还是稳定！

服务理念：客户在哪里，我们就在哪里！



深圳总部

地址：广东省深圳市宝安区新桥街道新桥社区
新和大道 6-18 号 1203

网址：www.zhiqwl.com

电话：0755-23203231

北京办事处

地址：北京市房山城区德润街6号院8号楼3层

电话：18210365439

更多销售网络正在紧张筹备中……



天猫店铺



淘宝店铺



京东店铺



微信公众号



公司官网