

【智嵌物联】智能网关型串口服务器系列二次开发说明书



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



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	2022.03.12	创建文档
V1.1	2022.07.22	1、增加编译失败处理方法

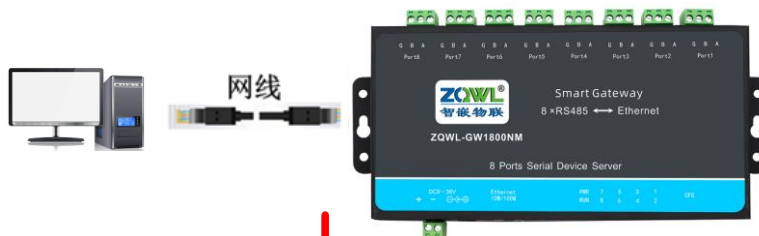
目 录

1. 二次开发流程	1
1.1 开发流程	1
1.2 智嵌物联 Smart Gateway 产品线命名规则	2
1.2.1 智嵌物联 Smart Gateway 产品线命名规则	2
1.2.2 相关资料下载:.....	2
1.2.3 产品价格查询及购买连接.....	2
2. 搭建开发环境	3
2.1 下载二次开发资料.....	3
2.2 安装虚拟机 VirtualBox.....	4
2.3 安装 Source Insight.....	10
2.3.1 新建用户的文件夹“myapp”	10
2.3.2 新建工程.....	10
2.4 编译生成 app.bin 文件	12
2.5 下载 app.bin 文件	13
2.6 异常情况处理.....	14
2.6.1 编译失败.....	14
2.6.2 下错程序致使设备无法正常下载程序.....	14
3. 函数库说明	15
3.1.1 struct can_frame	15
3.1.2 zq_recv_cb	15
3.1.3 zq_set_recv_cb.....	15
3.1.4 dbg_user_printf	16
3.1.5 zq_get_tcp_status	16
3.1.6 zq_tcp_send	16
3.1.7 zq_uart_send	17
3.1.8 zq_get_Y.....	17
3.1.9 zq_get_X.....	17
3.1.10 zq_set_Y	17
3.1.11 zq_sleep	18
3.1.12 zq_create_timer	18
3.1.13 zq_start_timer.....	18
3.1.14 zq_stop_timer	18
3.1.15 zq_restart	19
3.1.16 zq_set_version	19
3.1.17 zq_flash_write.....	19
销售网络.....	20

1. 二次开发流程

1.1 开发流程

1 搭建硬件环境

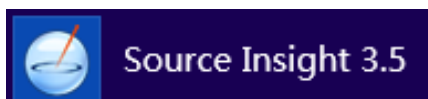


2 安装虚拟机

(若使用云编译功能，则无需安装虚拟机)



3 用户编程



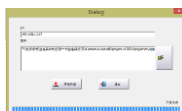
按照用户的需求，编写程序，并保存

4 编译生成app.bin文件



编译上一步骤的Project，
并生成app.bin文件

5 下载app.bin文件



将上一步生成app.bin文件，下载到设备中

1.2 智嵌物联 Smart Gateway 产品线命名规则

1.2.1 智嵌物联 Smart Gateway 产品线命名规则

智嵌物联 Smart Gateway 系列产品命名规则如图 1.1 所示。

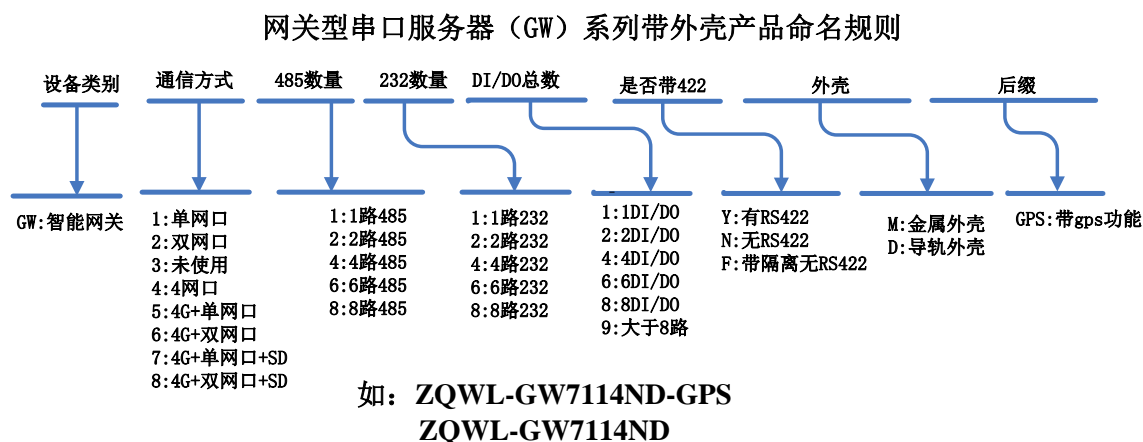


图 1.1 智嵌 Smart Gateway 网关型串口服务器产品线命名规则

1.2.2 相关资料下载:

使用说明书下载: [点击下载](#)

二次开发资料下载: [点击下载](#)

配置工具及测试软件下载: [点击下载](#)

MQTTX 工具下载: [点击下载](#)

虚拟串口软件及使用说明下载: [点击下载](#)

1.2.3 产品价格查询及购买连接

天猫商城: [点击查看](#)

京东商城: [点击查看](#)

2. 搭建开发环境

2.1 下载二次开发资料

在官网或淘宝链接内下载“Smart Gateway 二次开发资料”，解压后，有 5 个文件，如图 2.1 所示。

名称	修改日期	类型
gateway.exe ← 配置软件：用于下载程序、配置参数	2022/3/3 星期四 ...	应用程序
project-SmartGw-V0001.zip ← SDK工程文件、编译软件	2022/3/9 星期三 ...	ZIP 文件
sourceInsight_downcc.rar ← sourceInsight安装包	2022/3/11 星期...	WinRAR 压缩文件
智嵌物联 Smart Gateway设备系列二次开发说明书V1.0.pdf	2022/3/11 星期...	PDF 文件

↑
二次开发使用说明

图 2.1 压缩包文件

1. project 文件夹

project 文件夹内的文件是二次开发的 SDK 包，文件如

名称	修改日期	类型
demo-flash	2022/3/8 星期二 ...	文件夹
demo-gpio	2022/3/7 星期一 ...	文件夹
demo-httpc	2022/3/8 星期二 ...	文件夹
demo-httpc-get	2022/3/8 星期二 ...	文件夹
demo-httpc-post	2022/3/8 星期二 ...	文件夹
demo-mqtt	2022/3/9 星期三 ...	文件夹
demo-openssl-aes	2022/3/9 星期三 ...	文件夹
demo-read-id	2022/3/8 星期二 ...	文件夹
demo-task	2022/3/8 星期二 ...	文件夹
demo-tcpclient	2022/3/8 星期二 ...	文件夹
demo-tcpserver	2022/3/8 星期二 ...	文件夹
demo-timer	2022/3/8 星期二 ...	文件夹
demo-uart	2022/3/8 星期二 ...	文件夹
demo-uart1	2022/3/8 星期二 ...	文件夹
demo-udpclient	2022/3/8 星期二 ...	文件夹
demo-udpserver	2022/3/8 星期二 ...	文件夹
compile.exe ← 编译工具，用于编译用户的代码，生成app.bin文件	2022/3/9 星期三 ...	应用程序
user_code.h ← 头文件，包含设备所有函数库，更多函数库持续更新	2022/3/8 星期二 ...	头文件

← ZQWL提供的demo，更多demo持续更新

图

2.2 所示。

demo 文件：文件夹内有 app.c 文件和 app.bin 文件。app.c 文件是源码文件，app.bin 文件是通过编译工具“compile.exe”自动生成的文件。

compile.exe 文件：编译工具，用于编译用户的代码，生成 app.bin 文件。

user_code.h：头文件。

名称	修改日期	类型
demo-flash	2022/3/8 星期二 ...	文件夹
demo-gpio	2022/3/7 星期一 ...	文件夹
demo-httpc	2022/3/8 星期二 ...	文件夹
demo-httpc-get	2022/3/8 星期二 ...	文件夹
demo-httpc-post	2022/3/8 星期二 ...	文件夹
demo-mqtt	2022/3/9 星期三 ...	文件夹
demo-openssl-aes	2022/3/8 星期二 ...	文件夹
demo-read-id	2022/3/8 星期二 ...	文件夹
demo-task	2022/3/8 星期二 ...	文件夹
demo-tcpclient	2022/3/8 星期二 ...	文件夹
demo-tcpserver	2022/3/8 星期二 ...	文件夹
demo-timer	2022/3/8 星期二 ...	文件夹
demo-uart	2022/3/8 星期二 ...	文件夹
demo-uart1	2022/3/8 星期二 ...	文件夹
demo-udpclient	2022/3/8 星期二 ...	文件夹
demo-udpservice	2022/3/8 星期二 ...	文件夹
compile.exe	2022/3/9 星期三 ...	可执行文件
user_code.h	2022/3/8 星期二 ...	头文件

← ZQWL提供的demo，更多demo持续更新

← 编译工具，用于编译用户的代码，生成app.bin文件

← 头文件，包含设备所有函数库，更多函数库持续更新

图 2.2 project 文件夹内容

2. sourceInsight_downcc 文件夹

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

3. Smart Gateway 配置软件

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。

2.2 安装虚拟机 VirtualBox

提示：当使用智嵌物联提供的云编译功能时，虚拟机就不需要下载安装了。此小节就可以跳过了。

虚拟机下载地址：

链接：<https://pan.baidu.com/s/1CLTL92IAacRFKdeGFIxlyw?pwd=1234>

该文件夹内含有 32 位、64 位系统 VirtualBox 虚拟机的安装包。

名称	修改日期	类型	大小
smartio虚拟机.ova	2021/9/17 星期二 ...	Open Virtualizati...	1,906,289...
virtualBox-32.zip	2021/9/17 星期二 ...	ZIP 文件	107,858 KB
VirtualBox-64.exe	2021/8/7 星期六 ...	应用程序	105,699 KB

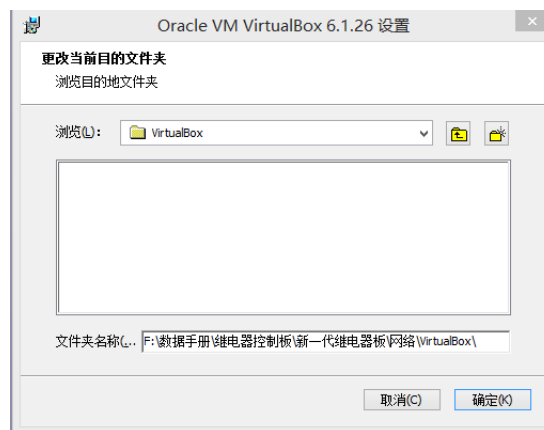
← Smart io虚拟机硬盘文件

← 32位系统安装包

← 64位系统安装包

图 2.3 虚拟机文件夹内容

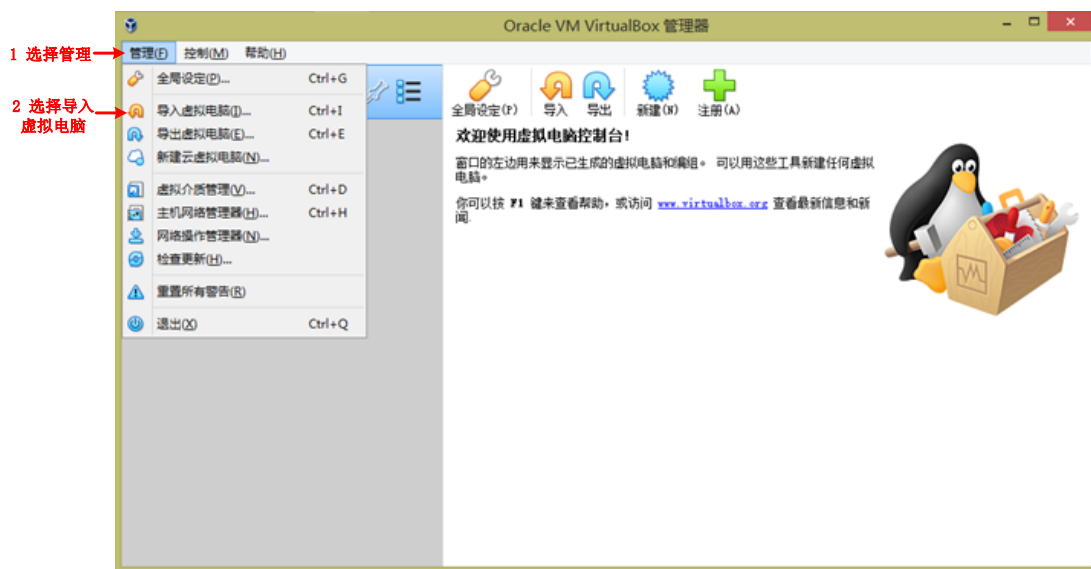
1. 双击文件中的 VirtualBox 安装包，按照安装向导步骤进行安装。



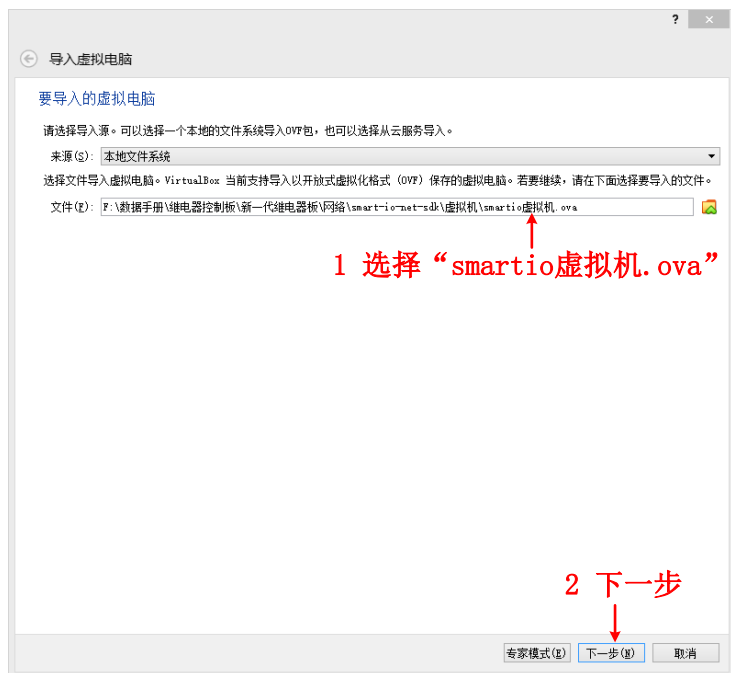


虚拟电脑安装完成，下面需要将 smartio.ova 文件导入到虚拟电脑中。

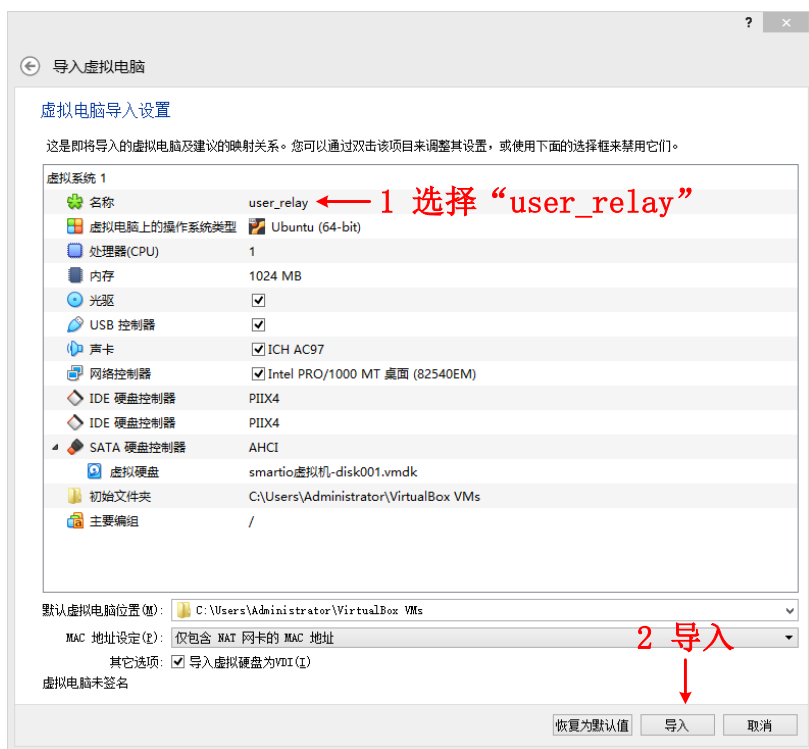
2. 导入虚拟电脑

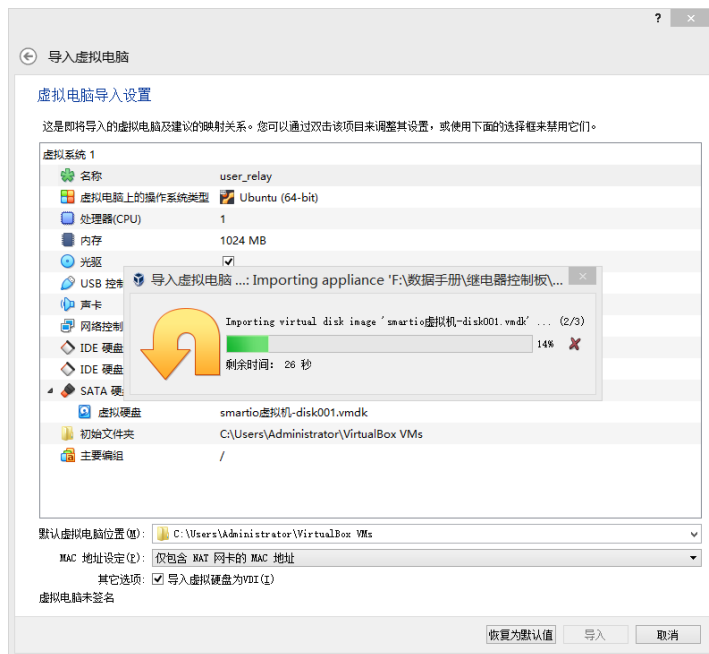


3. 将“smartio 虚拟机.ova”文件导入虚拟电脑



4. 将“user_relay”设置导入虚拟电脑

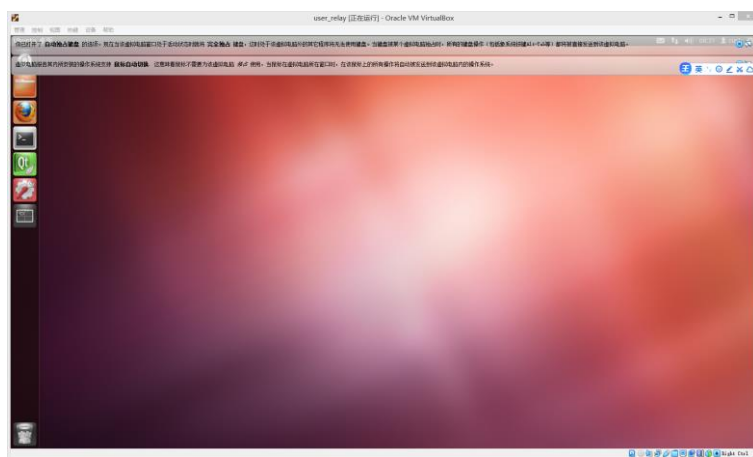




导入时间长一点，耐心等待即可。

5. 打开虚拟电脑

双击“user_relay”，启动虚拟电脑，启动成功后的界面如图所示。

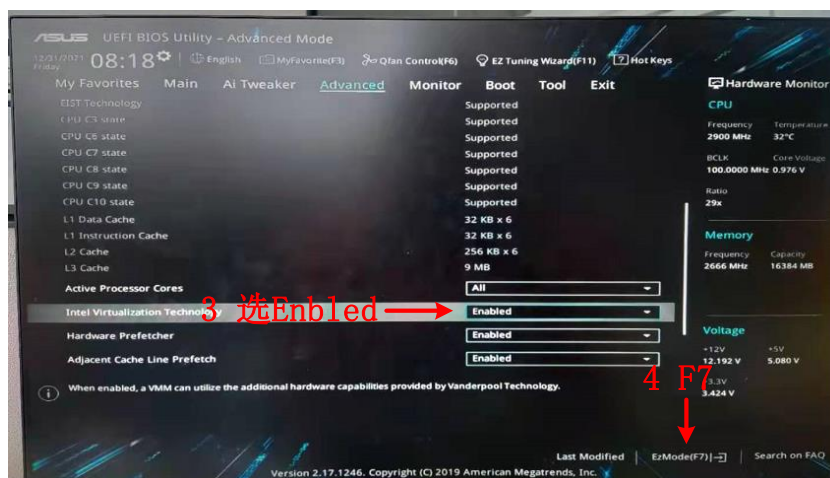
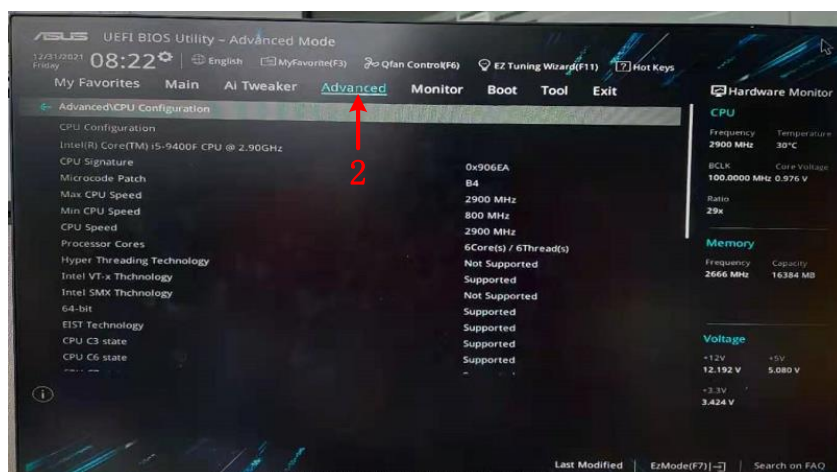
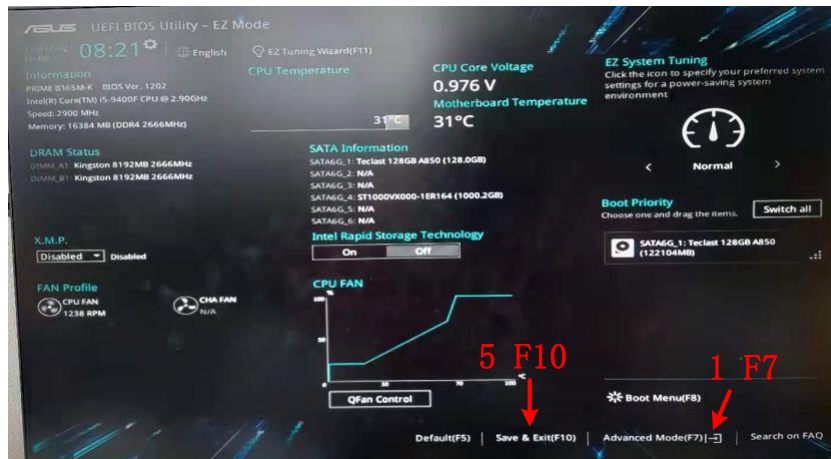


至此，虚拟电脑已经安装好，并将 smart io 的相关设置导入到了虚拟电脑，下面用户就

可以编译自己的 app.c 文件了。

6. 虚拟电脑打开异常处理方法

若用户在打开虚拟电脑的时候，遇到以下问题，则需要将 BIOS 中的 Intel Virtualization Technology 选项设置为 Enable，具体操作步骤如下：



2.3 安装 Source Insight

双击文件中的“setup.exe”，按照 Source Insight 安装向导进行操作直至安装成功。

2.3.1 新建用户的文件夹“myapp”

用户在 Project 目录下新建属于自己的文件夹，名称可自取，例如“myapp”。用户将 demo 文件夹下的 app.c 文件复制到“myapp”文件夹内，并将 user_code.h 文件也复制到 myapp 文件夹内。

名称	修改日期	类型	大小
demo-can	2021/12/14 星期...	文件夹	
demo-gpio	2021/8/7 星期六 ...	文件夹	
demo-modbuspoll	2021/9/17 星期...	文件夹	
myapp	2022/1/3 星期一 ...	文件夹	
compile.exe	2021/12/20 星期...	应用程序	9,926 KB
user_code.h	2021/12/14 星期...	H 文件	3 KB

将demo中的app.c文件
复制到myapp文件中

用户自建文件夹

将该文件复制到myapp文件中

2.3.2 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。

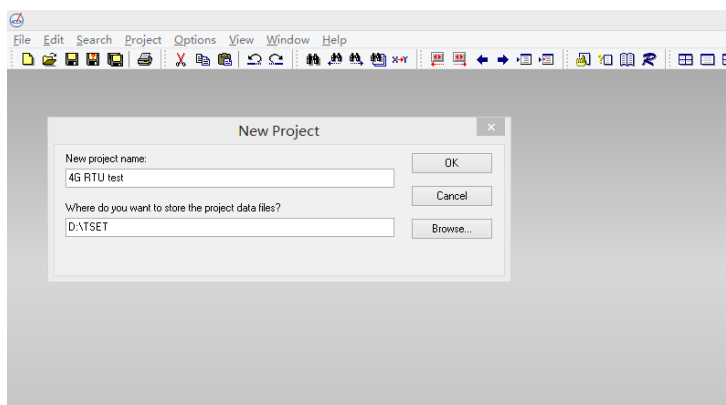


图 2.4 Source Insight 新建工程

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 -> myapp。

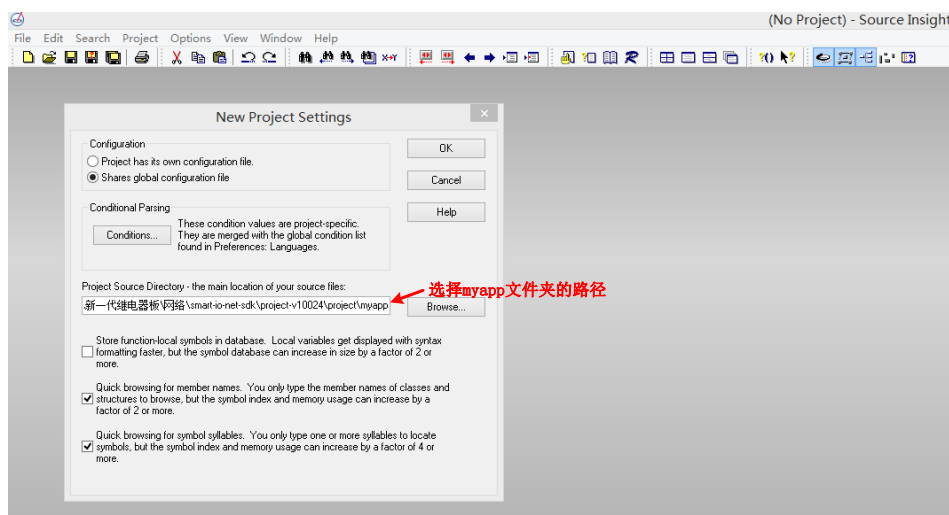


图 2.5 选择源码所在路径

3. 向工程中添加文件：app.c 文件和 user_code.h 文件

按图 2.6 所示步骤操作，将 app.c 文件和 user_code.h 文件添加到工程中。



图 2.6 向工程中添加源文件

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

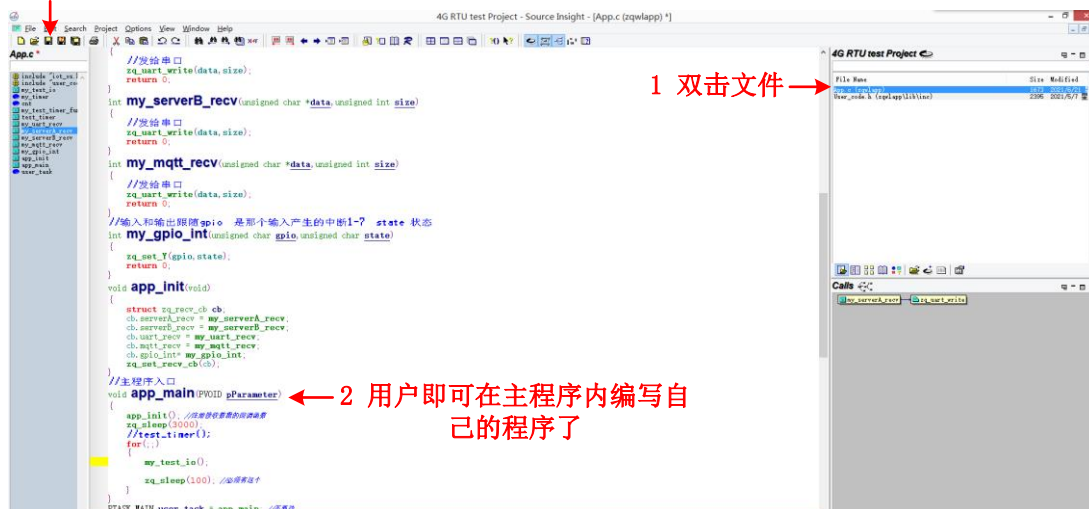


图 2.7 编写用户代码

2.4 编译生成 app.bin 文件

从二次开发资料中找到编译工具：**compile.exe**（路径：二次开发包的解压文件 project>compile.exe）。有两种方法对工程进行编译：通过虚拟机编译和通过云编译。

1. 方法一：虚拟机编译

首先将上节安装的虚拟机打开，按照所示，对用户自己的 **app.c** 文件进行编译。编译结束后，会生成 **app.bin** 文件。

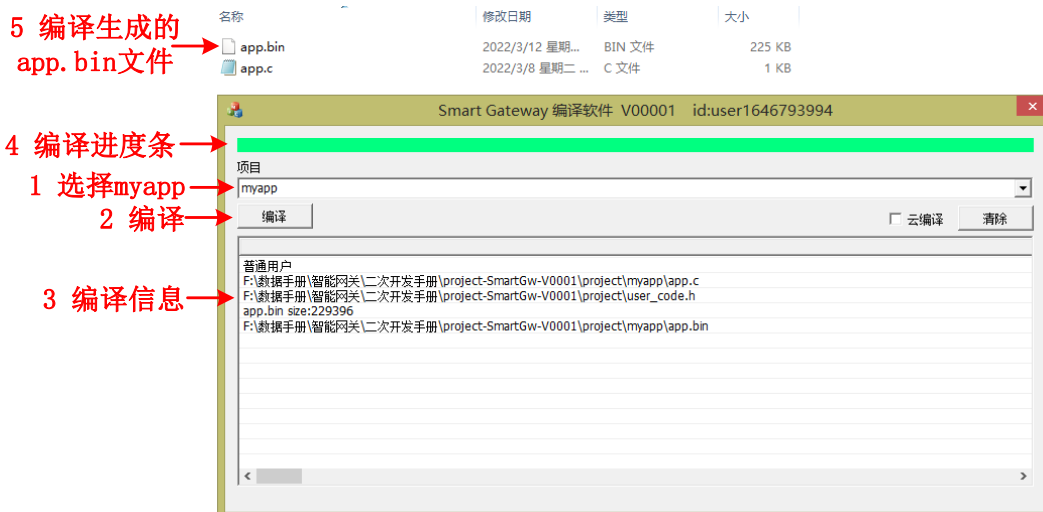


图 2.8 虚拟机编译生成 app.bin 文件

2. 方法二：云编译

由于虚拟机文件很大，且安装麻烦，因此智嵌物联独创开发了云编译功能，用户不需要下载安装虚拟机，只需一台能联网的电脑即可。

首先确保电脑能联外网，比如能上百度等。

按照所示，对用户自己的 app.c 文件进行编译。编译结束后，会生成 app.bin 文件。



图 2.9 云编译生成 app.bin 文件

2.5 下载 app.bin 文件

打开【Smart Gateway 配置软件】，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。

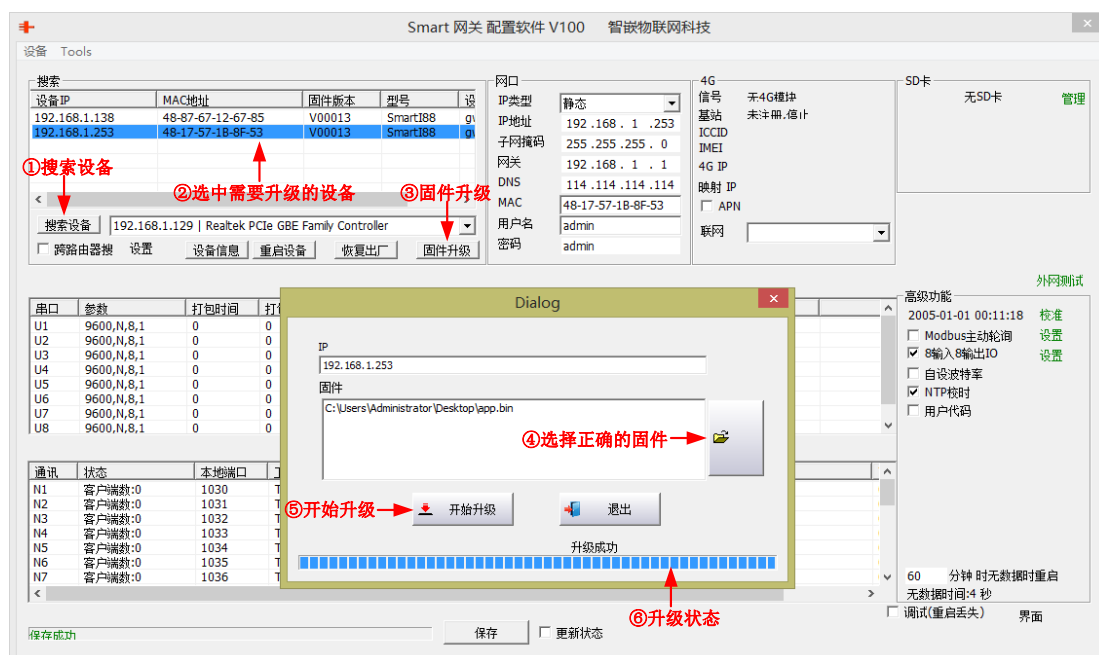


图 2.10 下载 app.bin 文件

至此，用户编写的程序已经下载到设备中了。

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，至此，设备就已经在运行用户自己的代码了。具体步骤如图 2.11 所示。

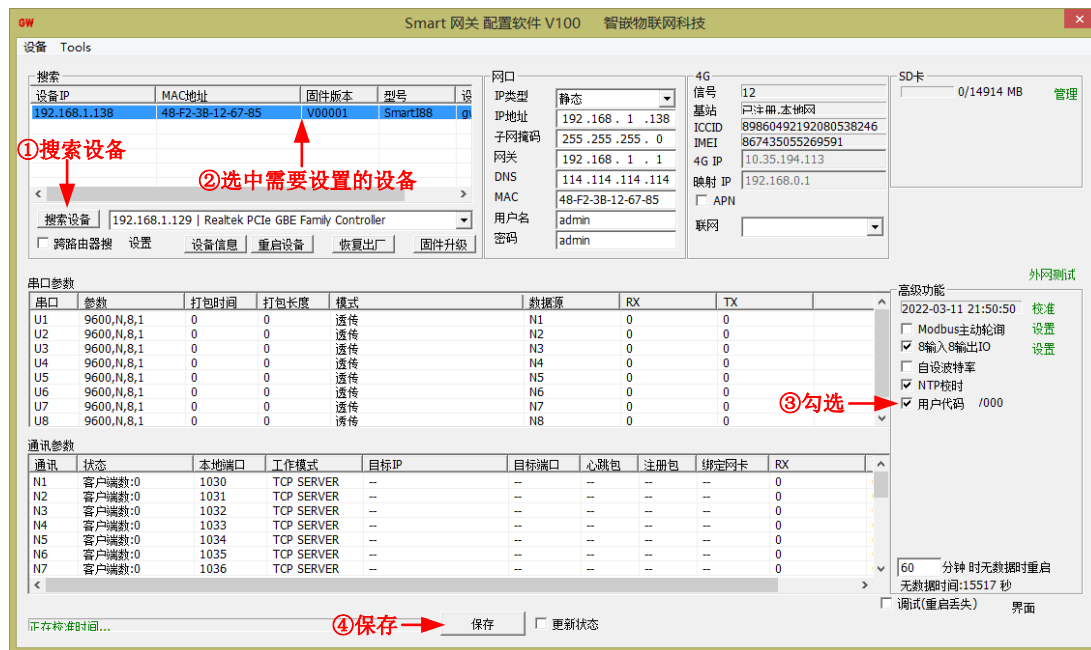


图 2.11 配置设备参数

2.6 异常情况处理

2.6.1 编译失败

1. 第一次编译失败，可多尝试几次编译
2. 如果编译几次之后依然失败，点击编译软件左侧的【清除】，然后再尝试编译
3. 点击清除后，编译依然失败，检查工程目录是否有特殊字符，比如空格。
4. 工程目录正常，编译依然失败，重新解压工程文件的压缩包，然后再尝试编译。

2.6.2 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。

将设备断电，重新上电，等 10 秒，然后一直按住“CFG”按键，直到 RUN 灯快闪，然后再搜索设备，搜到设备之后再重新升级新固件（升级完成之后，再松开按键）。

3. 函数库说明

最新 API 接口，请查看工程目录中的 user_code.h 文件。

名称	修改日期	类型	大小
demo_default-vip	2022/7/1 星期五	文件夹	
demo-cjson	2022/7/22 星期五	文件夹	
demo-config	2022/7/1 星期五	文件夹	
demo-factory-default	2022/7/1 星期五	文件夹	
demo-flash	2022/7/1 星期五	文件夹	
demo-gpio	2022/7/1 星期五	文件夹	
demo-httpc	2022/7/1 星期五	文件夹	
demo-httpc-get	2022/7/1 星期五	文件夹	
demo-httpc-post	2022/7/1 星期五	文件夹	
demo-modbus_poll	2022/7/1 星期五	文件夹	
demo-mqtt	2022/7/1 星期五	文件夹	
demo-openssl-aes	2022/7/1 星期五	文件夹	
demo-power-off	2022/7/1 星期五	文件夹	
demo-read-id	2022/7/1 星期五	文件夹	
demo-task	2022/7/1 星期五	文件夹	
demo-tcpclient	2022/7/1 星期五	文件夹	
demo-tcpserver	2022/7/1 星期五	文件夹	
demo-timer	2022/7/1 星期五	文件夹	
demo-uart	2022/7/1 星期五	文件夹	
demo-uart1	2022/7/1 星期五	文件夹	
demo-udpclient	2022/7/1 星期五	文件夹	
demo-udpsrvr	2022/7/1 星期五	文件夹	
cjson.h	2021/9/24 星期五	H 文件	8 KB
compile-v10023.exe	2022/7/1 星期五	应用程序	5,738 KB
readme.txt	2022/4/2 星期六	文本文件	0 KB
Smart-Gw-V109.exe	2022/6/29 星期五	应用程序	3,744 KB
user_code.h	2022/6/30 星期五	H 文件	15 KB

3.1.1 struct can_frame

描述

创建 CAN 帧的结构体

```
struct can_frame{
    unsigned char length:4;//
    unsigned char res : 2;//donot care
    unsigned char rtr : 1;//0 : data 1 : remote
    unsigned char type :1;//0:sta 1: ext
    unsigned int id ;
    unsigned char data[8];
};
```

3.1.2 zq_recv_cb

描述

创建回调函数结构体

```
struct zq_recv_cb{
    int (*tcp_recv)( unsigned char *data,unsigned int size,unsigned char tcp,int sock,void *p);
    int (*uart_recv)( unsigned char *data,unsigned int size,unsigned char uart);
};
typedef void (*timer_cb)(void *data);
```

3.1.3 zq_set_recv_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

3.1.4 dbg_user_printf

描述

打印调试信息，可用配置软件查看

```
void dbg_user_printf(const char *fmt,...);
```

参数

无

返回值

无

3.1.5 zq_get_tcp_status

描述

获取服务器的连接状态

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

参数

Idx: 0~23, 分别代表网络通道 N1~N24

返回值

0 未连接

1 已成功连接

3.1.6 zq_tcp_send

描述

向服务器发送数据

```
int zq_tcp_send(unsigned char idx,unsigned char *data,unsigned int size,int sock,void *p);
```

参数

Idx 0~23, 分别代表网络通道 N1~N24

data 要发送的数据

size 发送数据的大小

sock 向哪个 socket 发送数据

*p 私有参数

返回值

已经发送的数据大小

3.1.7 zq_uart_send

描述

向串口发送数据

```
int zq_uart_send(unsigned char idx,unsigned char *data,unsigned int size);
```

参数

Idx 0~23，分别代表串口通道 U1~U24

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

3.1.8 zq_get_Y

描述

获取 DO 输出 Y 的状态

```
char zq_get_Y(unsigned char gpio);
```

参数

gpio 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

1 输出高电平

3.1.9 zq_get_X

描述

获取 DI 输入 X 的状态

```
char zq_get_X(unsigned char gpio);
```

参数

gpio 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

3.1.10 zq_set_Y

描述

设置 DO 输出 Y 的状态

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

参数

gpio 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；
1：输出高电平。

返回值

0 输出低电平
1 输出高电平

3.1.11 zq_sleep

描述

休眠

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

参数

0~65535

返回值

无

3.1.12 zq_create_timer

描述

创建定时器

```
int zq_create_timer(timer_cb cb, 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

3.1.13 zq_start_timer

描述

启动定时器

```
unsigned char zq_start_timer(int fd,int ms,int repeat);
```

参数

fd 定时器句柄
ms 定时器时间
repeat 是否重复启动定时器：0，只启动一次；1 循环启动定时器。

返回值

0 失败
1 成功

3.1.14 zq_stop_timer

描述

停止定时器

```
unsigned char zq_stop_timer( int fd);
```

参数

fd 定时器句柄

返回值

0 失败

1 成功

3.1.15 zq_restart

描述

重启设备

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

参数

无

返回值

无

3.1.16 zq_set_version

描述

设置用户代码版本 可以在配置软件中查看

```
void zq_set_version(unsigned int ver);
```

参数

ver 软件版本号

返回值

无

3.1.17 zq_flash_write

描述

向 flash 写入数据，一般是一些不经常修改的配置参数。不要频繁写入及擦除 flash。

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

参数

addr 写入 flash 的数据地址

data 写入 flash 的数据

size 写入 flash 的数据大小

返回值

-1 写 flash 失败

其他 写入 flash 的数据大小

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