

FANCONN™ 开源 NBIOT 开发平台	文档编号	产品版本	密级
	V1.0	V1.0	公开
	文档名称：LiteOne DEMO 指导		共 13 页

# **FANCONN 开源 NBIOT 开发平台**

## **LiteOne DEMO 指导**

广州准捷电子科技有限公司

版本：V1.0

## 版权声明

本手册版权属广州准捷电子科技有限公司（以下简称“准捷电子”）所有，并保留一切权利。非准捷电子同意（书面形式），任何单位及个人不得擅自摘录本手册部分或全部，违者我们将追究其法律责任。

## 版本记录

版本号	作者	版本说明	日期
V1.0	阿黄	初始版本	2018-10-8

## 目录

1.环境搭建.....	4
2. 选择合适的代码编辑工具.....	5
3.添加工程.....	6
4. 常用外设配置.....	13
GPIO 映射.....	13
左边引脚与程序里面的 GPIO 对应（有一些丝印印错了，引脚以下图为准）.....	13
右边引脚与程序里面的 GPIO 对应（有一些丝印印错了，引脚以下图为准）.....	14
4.1 GPIO 配置.....	15
4.2 UART 配置.....	16
4.3 IIC 配置.....	17
5. 结束语.....	17
6. 贡献列表.....	18
六.技术支持.....	19

## 1.环境搭建

目前版本的开发 SDK 由我们的模块厂商上海移远通信技术股份有限公司提供。相关的开发软件环境包也准备的十分丰富。包括编译器，运行的软件库，编译工具等。都有提供

注意，所有的环境和工具都请安装在默认目录，不然到时候可能需要自己将它们导入到环境变量

名称	修改日期	类型	大小
eclipse-cpp-neon-2-win32_plugins.rar	2018/8/27 11:44	WinRAR 压缩文件	221,257 KB
gcc-arm-none-eabi-4_9-2015q2-2015...	2018/8/27 11:43	应用程序	64,923 KB
jdk_8u91_windows_i586_8.0.910.15.exe	2018/8/27 11:43	应用程序	186,669 KB
pip-9.0.1.rar	2018/8/27 11:43	WinRAR 压缩文件	2,936 KB
python-2.7.10.msi	2018/8/27 11:44	Windows Install...	17,992 KB
pywin32-220.win32-py2.7.exe	2018/8/27 11:43	应用程序	6,771 KB
Quectel_BC35-G&BC28_OPENCPU_开...	2018/8/27 11:43	Foxit Reader Plu...	921 KB
scons-2.4.0-setup.exe	2018/8/27 11:43	应用程序	861 KB
UEMonitor-3.22.0.14.msi	2018/8/27 11:43	Windows Install...	6,656 KB
UEUpdaterUI-3.22.0.14.msi	2018/8/27 11:43	Windows Install...	4,940 KB
UpdatePackage-3.22.0.14.msi	2018/8/27 11:44	Windows Install...	4,720 KB



这些资料将会在我们官方网站或 LiteOS 团队的推广渠道整合发放。

## 2. 选择合适的代码编辑工具

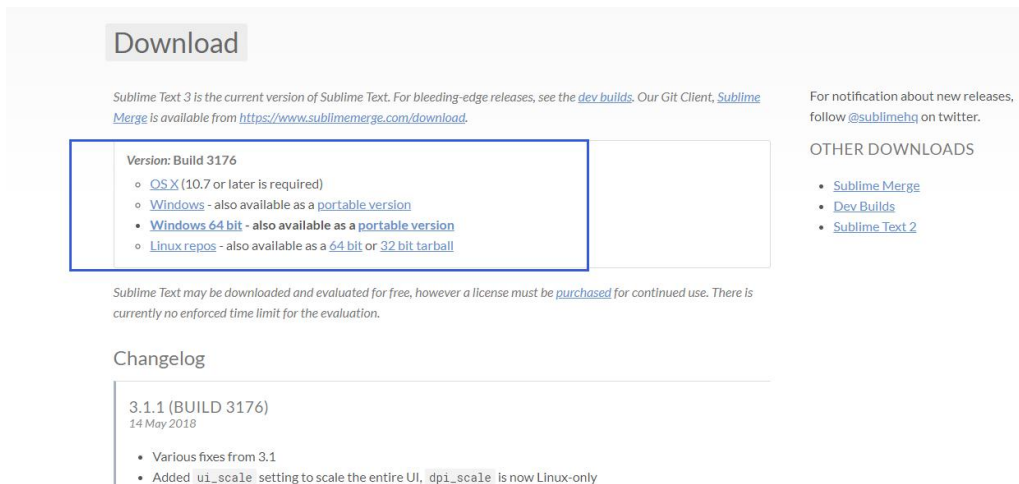
一个好的代码编辑工具对于我们开发人员来说是十分重要的,并且每个人都有自己的偏好去选择自己的代码编辑工具。这里我需要一个代码编辑工具来给大家演示 LiteOne 开发板的工程搭建和修改,所以我在这里给大家推荐 Sublime。

下载:

Sublime 可以直接通过官网下载

<https://www.sublimetext.com/3>

这个工具的最新版本只支持 64 位系统, 所以 32 位的老用户得要注意一下。当然, 这里也推荐大家使用 64 位的系统对 LiteOne 进行开发, 因为移远的 SDK 里面很多环境都是 64 位的。



**Download**

Sublime Text 3 is the current version of Sublime Text. For bleeding-edge releases, see the [dev builds](#). Our Git Client, [Sublime Merge](#) is available from <https://www.sublimemerge.com/download>.

Version: Build 3176

- [OS X](#) (10.7 or later is required)
- [Windows](#) - also available as a [portable version](#)
- [Windows 64 bit](#) - also available as a [portable version](#)
- [Linux repos](#) - also available as a [64 bit](#) or [32 bit tarball](#)

For notification about new releases, follow [@sublimehq](#) on twitter.

**OTHER DOWNLOADS**

- [Sublime Merge](#)
- [Dev Builds](#)
- [Sublime Text 2](#)

Sublime Text may be downloaded and evaluated for free, however a license must be [purchased](#) for continued use. There is currently no enforced time limit for the evaluation.

**Changelog**

3.1.1 (BUILD 3176)  
14 May 2018

- Various fixes from 3.1
- Added `ui_scale` setting to scale the entire UI, `dpi_scale` is now Linux-only

### 3.添加工程

我们 LiteOne 现在的工程搭建是基于移远给出的原版工程在 API 层写入自己的逻辑程序, 首先我们将移远提供的原版工程拷贝到你的开发目录下 (注意, 目录不能有中文和特殊字符, 只能有数字, 英文大小写和下划线)

然后我们打开 `src/example/` 你会发现下面有两个文件夹

地址: 广州市天河区软件路 17 号亿牛大厦 5 楼

电话: +86 18078868880

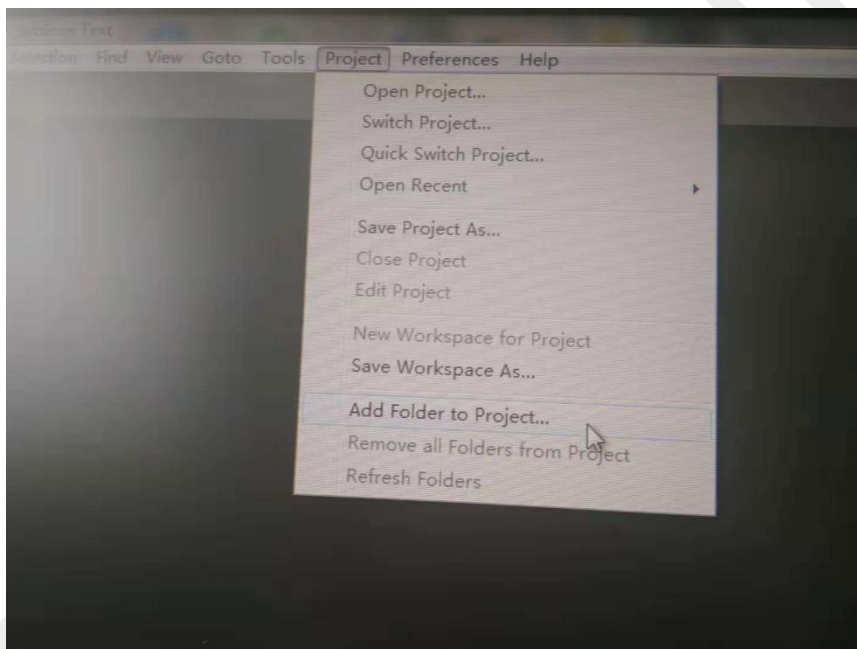
电子邮件: [yellow@fanconn.com](mailto:yellow@fanconn.com)

名称	修改日期	类型
private	2018/8/21 15:12	文件夹
public	2018/8/21 15:12	文件夹
SConscript	2018/8/20 13:49	文件

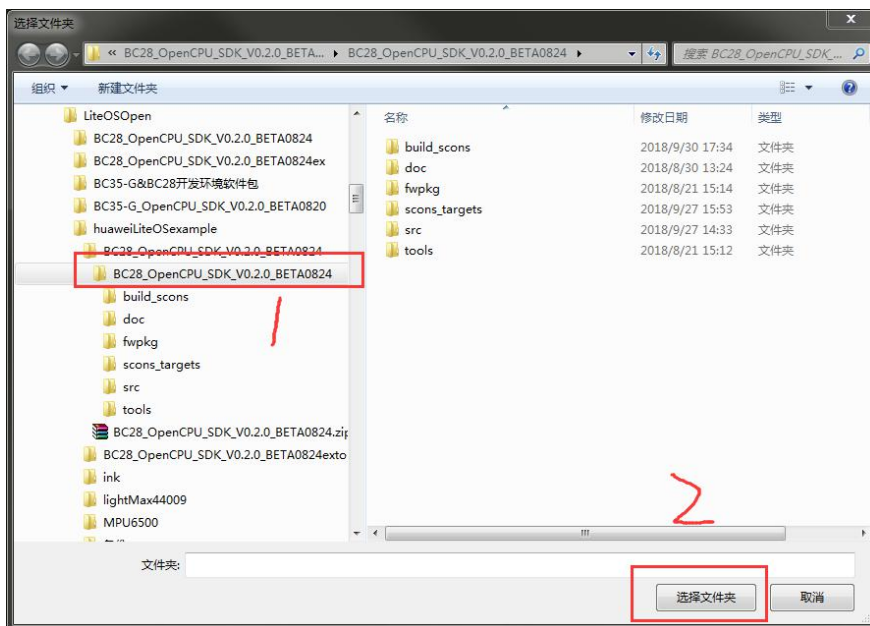
其中 **private** 是用来放.c 文件，**public** 是用来放.h 文件。

因为目前版本的编译脚本限制，想要创建新的工程目录是十分麻烦的，所以目前的程序文件最好直接写在这个 **example** 的 **private** 和 **public** 下。这个 **example** 下的程序都是 **main** 程序，所以一会在编译列表里面只能包含一个。

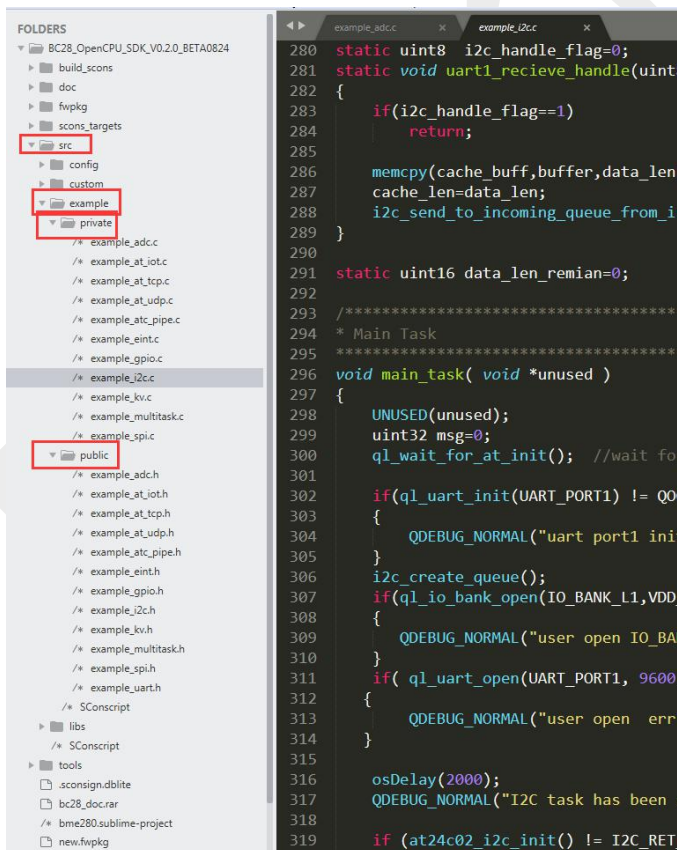
为了方便介绍，我们这里使用 **Sublime** 的工程管理功能，首先，我们先新建一个工程



点击这个 **add** 后会出现选择目录



点击选择后你就可以在你的工程目录下看到整个工程的所有文件

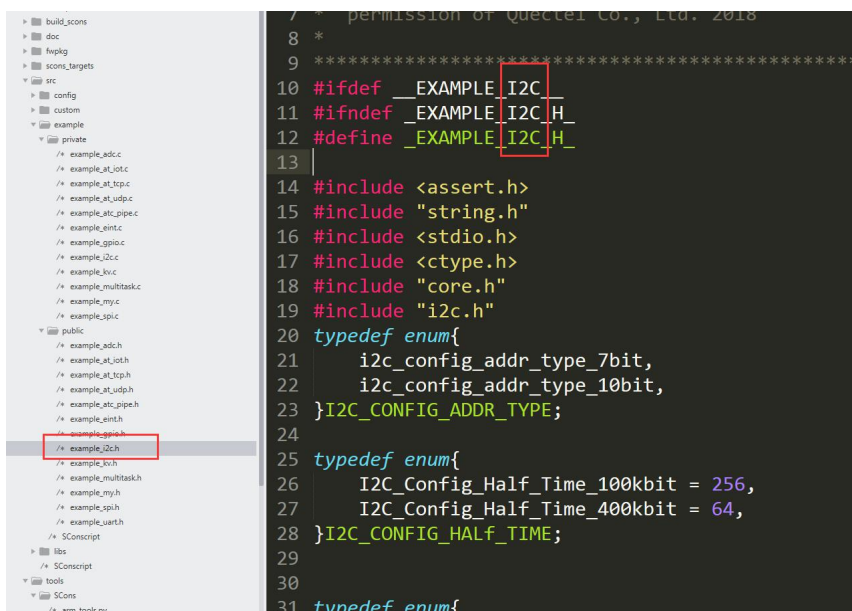


然后现在就可以开始添加你自己的.c 和.H 文件了。

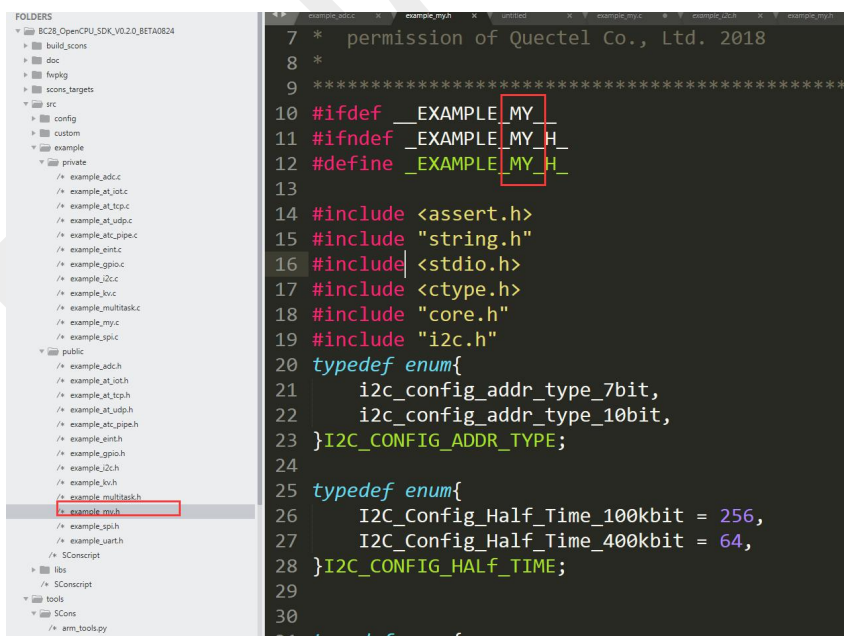


接下来我们正式开始新建我们的程序，这里我们就直接在 `private` 下创建 `example_my.c` 和 `public` 下创建 `example_my.h`

对于代码细节和 `api` 函数细节本文档不做深究，所以我们这边直接复制 `example_iic.c` 和 `example_iic.h` 中的内容做修改，来演示如何搭建 DEMO 并且编译。



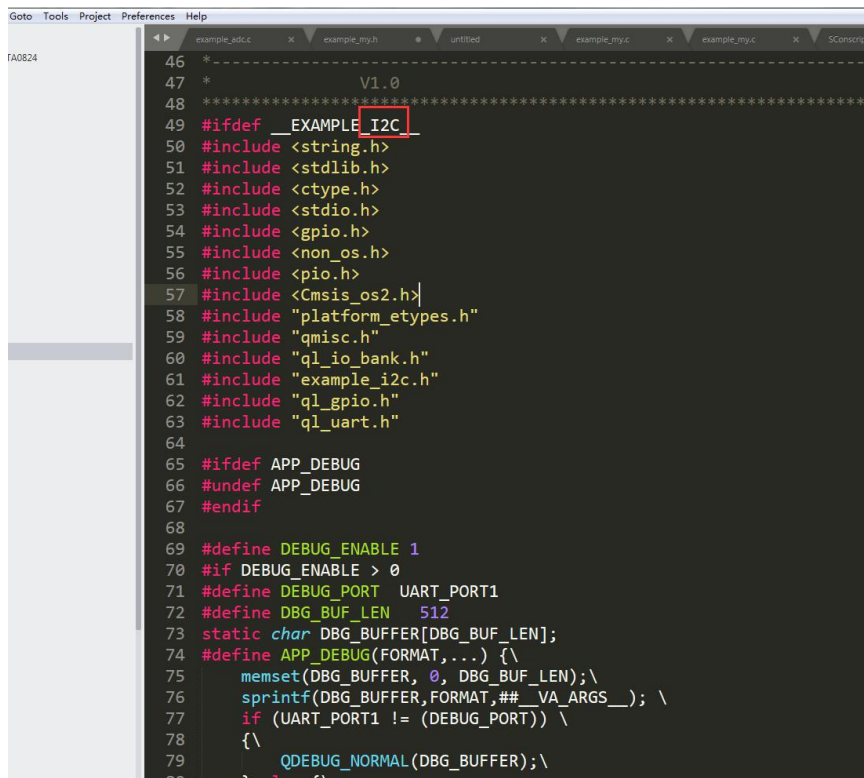
这一个是 `IIC.h` 的原文件，我们现在要将它改成我们的文件并且编译，我们在功能内容不变的情况下只需要对它的命名和宏定义进行修改



从 `IIC_H` 修改成 `MY_H`

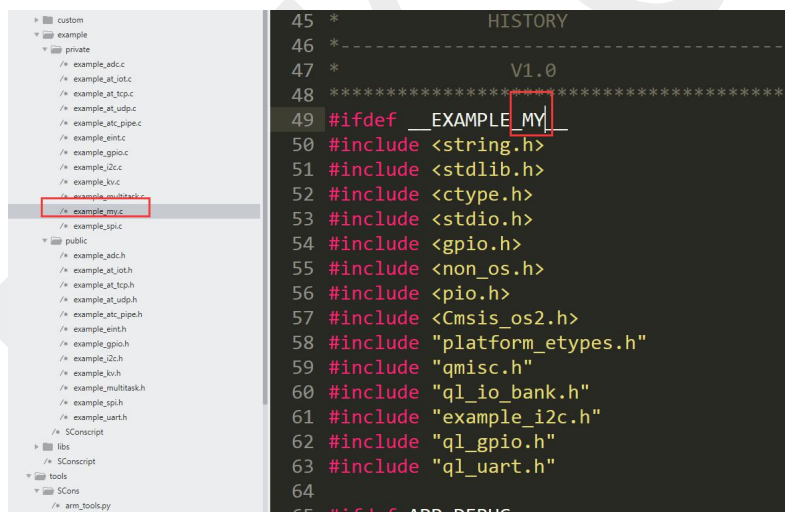


.c 文件也是同样的修改



```
46 *
47 *----- V1.0
48 *****
49 #ifndef __EXAMPLE_I2C_
50 #include <string.h>
51 #include <stdlib.h>
52 #include <ctype.h>
53 #include <stdio.h>
54 #include <gpio.h>
55 #include <non_os.h>
56 #include <pio.h>
57 #include <Cmsis_os2.h>
58 #include "platform_etypes.h"
59 #include "qmisc.h"
60 #include "ql_io_bank.h"
61 #include "example_i2c.h"
62 #include "ql_gpio.h"
63 #include "ql_uart.h"
64
65 #ifdef APP_DEBUG
66 #undef APP_DEBUG
67 #endif
68
69 #define DEBUG_ENABLE 1
70 #if DEBUG_ENABLE > 0
71 #define DEBUG_PORT UART_PORT1
72 #define DBG_BUF_LEN 512
73 static char DBG_BUFFER[DBG_BUF_LEN];
74 #define APP_DEBUG(FORMAT,...) {\
75     memset(DBG_BUFFER, 0, DBG_BUF_LEN);\
76     sprintf(DBG_BUFFER, FORMAT, ##__VA_ARGS__); \
77     if (UART_PORT1 != (DEBUG_PORT)) \
78     {\
79         QDEBUG_NORMAL(DBG_BUFFER);\
80     }\
81 }
```

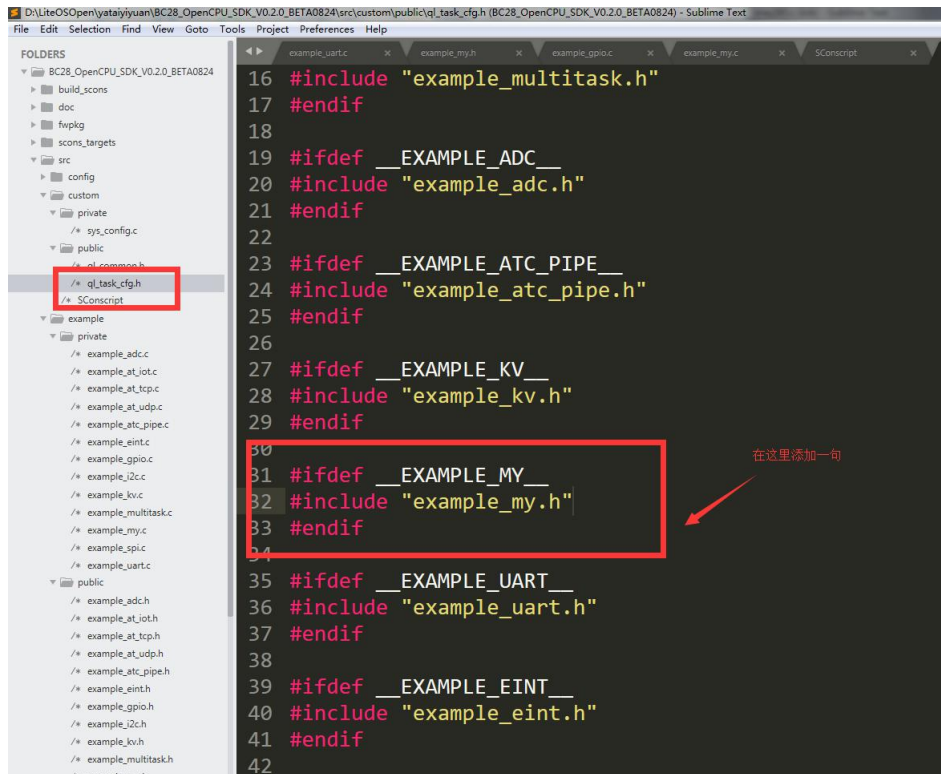
这是原版，然后下面是我们自己的



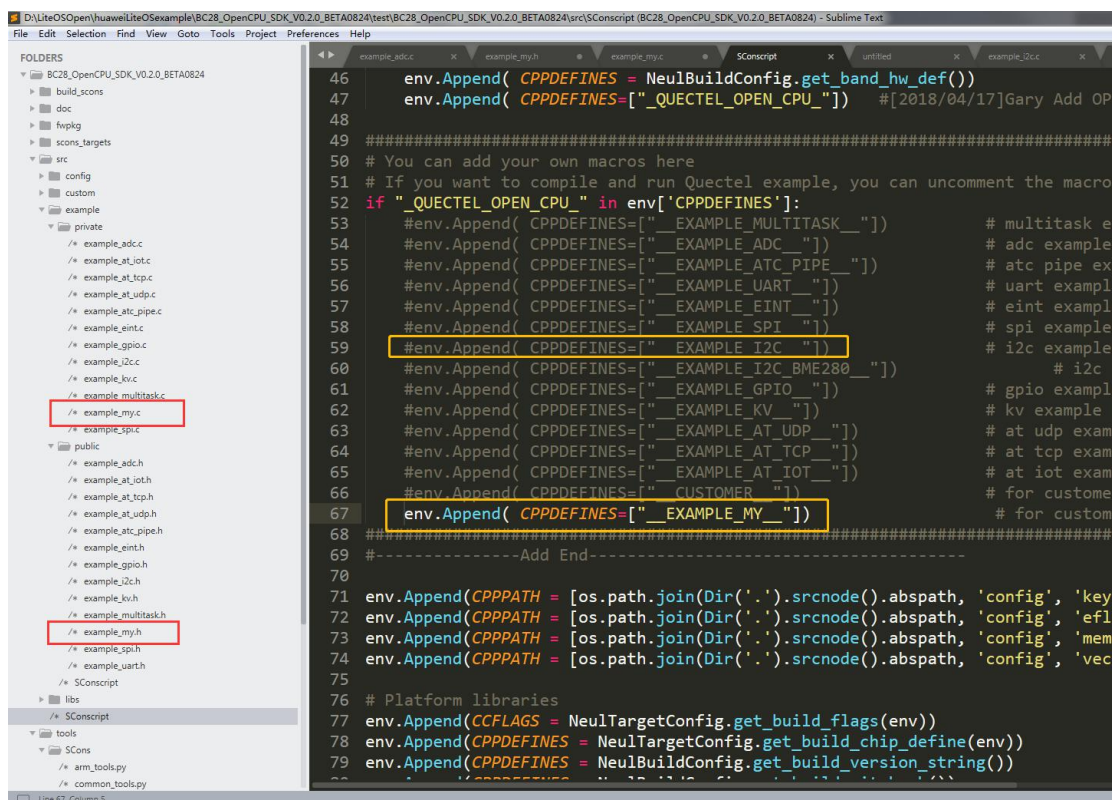
```
45 *
46 *----- HISTORY
47 *----- V1.0
48 *****
49 #ifndef __EXAMPLE_MY_
50 #include <string.h>
51 #include <stdlib.h>
52 #include <ctype.h>
53 #include <stdio.h>
54 #include <gpio.h>
55 #include <non_os.h>
56 #include <pio.h>
57 #include <Cmsis_os2.h>
58 #include "platform_etypes.h"
59 #include "qmisc.h"
60 #include "ql_io_bank.h"
61 #include "example_i2c.h"
62 #include "ql_gpio.h"
63 #include "ql_uart.h"
64
65 #ifdef APP_DEBUG
```

我们将条件编译条件就行了，这个之前在.h 中做定义了的

之后, 我们需要将我们定义的头文件加入到 `ql_task_cfg.h` 里面, 这样才能顺利实现工程链接



完成这些修改之后, 我们只需要在 src 里面的脚本下面 SConscript 中修改链接对象, 在 SConscript 第五十二行的位置上是修改链接对象的部分



我们把 example\_IIC 注释掉换成我们自己的 example\_MY。保存之后我们就可以按照之前提到过的《Quectel\_BC35-G&BC28\_OPENCPU\_开发环境安装指导》中写的一样进行编译和生成固件包文件了

## 5.1. 命令行编译

编译脚本: ..\SDK\src\_release\命令提示符

在命令行界面中输入如下命令完成编译和清除已生成文件操作。

编译命令: scons\_new

清除命令: scons\_new -c

生成的固件包文件:

..\SDK\new.fwpkg

我们先打开这个

名称	修改日期	类型	大小
build_scons	2018/9/30 17:34	文件夹	
doc	2018/8/30 13:24	文件夹	
fwpkg	2018/8/21 15:14	文件夹	
scons_targets	2018/9/27 15:53	文件夹	
src	2018/9/27 14:33	文件夹	
tools	2018/8/21 15:12	文件夹	
.sconsign.dblite	2018/9/30 17:37	DBLITE 文件	136 KB
bc28_doc.rar	2018/9/19 13:29	WinRAR 压缩文件	6,083 KB
bme280.sublime-project	2018/9/27 16:00	SUBLIME-PROJE...	1 KB
bme280.sublime-workspace	2018/10/8 11:33	SUBLIME-WORK...	40 KB
new.fwpkg	2018/9/30 17:37	FWPKG 文件	1,033 KB
scons_new.bat	2018/9/27 17:49	Windows 批处理...	1 KB
SConstruct	2018/8/20 13:49	文件	8 KB
VERSION_STRING	2018/8/20 13:49	文件	1 KB
命令提示符	2018/8/20 13:49	快捷方式	2 KB

点击运行开了之后

输入 `scons_new.bat` 他就会进入编译，之后就会生成 `new.fwpkg`

这个 `new.fwpkg` 就是你的 `example_my` 这个工程所生成的包文件，就可以直接用来烧录了。

注意：下载固件的时候必须按复位

## 4. 常用外设配置

### GPIO 映射

左边引脚与程序里面的 GPIO 对应（有一些丝印印错了，引脚以下图为准）：

注意：该版本的 SDK 开放了 LIGHT，这个 PIN 脚原先在公版 SDK 中预计是不开放的，不过目前 OLED 使用到该引脚，所以该引脚的配置方式有些特别，单独拿出来配置 成为普通 IO，

```
gpio_claim(PIN_11,GPIO_DIRECTION_OUTPUT); //配成 GPIO
gpio_set(PIN_11); //设置高电平
gpio_clear(PIN_11); //设置低电平
其余的 GPIO 配置方式请参考下面的配置
```

	5v	5v	
	GND	GND	
	RST	LIGHT	特殊配置
	RX17	TX18	
PINNAME_DTR_GPIO	DTR	RI	PINNAME_RI
PINNAME_SIM_DETECT	GPIO9	DCD	PINNAME_CTS_AUX
PINNAME_RTS_AUX	RTS	VEXT	
	SW_C	SW_D	
	NC	NC	

## 右边引脚与程序里面的 GPIO 对应（有一些丝印印错了，引脚以下图为准）：

注意：该版本的 SDK 开放了 TXDB，同上，原本公版 SDK 是不开放这个引脚做 IO 的。不过目前 B 版会使用到该引脚，进行低功耗设置，所以该引脚的配置方式有些特别，单独拿出来配置 成为普通 IO，

```
gpio_claim(PIN_16,GPIO_DIRECTION_OUTPUT); //配成 GPIO
```

```
gpio_set(PIN_16); //设置高电平
```

```
gpio_clear(PIN_16); //设置低电平
```

其余的 GPIO 配置方式请参考下面的配置

	3V3	3V3	
	GND	GND	
PINNAME_SPI_S12	SPI1SI	SPI1S0	PINNAME_SPI_S02
PINNAME_SPI_CS2	SPI1CS	SPI1CL	PINNAME_SPI_CLK2
	ADC	NC	
PINNAME_DBG_RXD	RXDB	TXDB	特殊配置
PINNAME_UART_TX3	TX29	RX28	PINNAME_UART_RX3
PINNAME_SPI_S0	SPI2S0	SPI2CS	PINNAME_SPI_CS
PINNAME_SPI_S1	SPI2SI	SPI2CLK	PINNAME_SPI_CLK



## 4.1 GPIO 配置

使用到的函数如下:

初始化函数:

```
ql_gpio_init( PINNAME_SPI_SI, PINDIRECTION_OUT, PINLEVEL_HIGH )
```

第一个是配置引脚, 第二个是输入输出方向, 这里配置为输出, 第三个是引脚初始化之后电平, 这里配置为高电平

再举一个例子:

```
ql_gpio_init( PINNAME_SPI_SI, PINDIRECTION_IN, PINLEVEL_NONE )
```

第一个是配置引脚, 第二个是输入输出方向, 这里配置为输入, 第三个是引脚初始化之后电平, 这里配置为默认引脚配置时的电平。

设置引脚电平:

```
ql_gpio_set_level(PINNAME_SPI_SI, PINLEVEL_LOW)
```

```
ql_gpio_set_level(PINNAME_SPI_SI, PINLEVEL_HIGH)
```

第一个参数是引脚, 第二个参数是设置高低电平

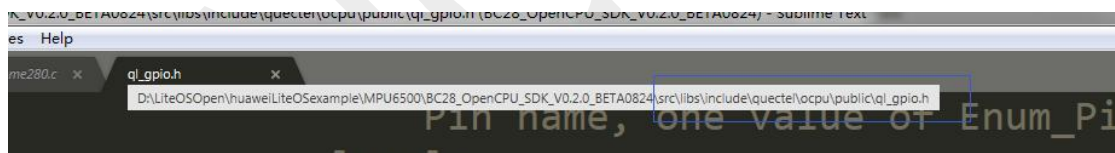
获取电平:

```
ql_gpio_get_level(PINNAME_SPI_SI)
```

返回电平状态

以上就是常用的几个 `gpio` 的函数简单说明, 至于其他函数可以参考例程和说明文档。

GPIO 的一些具体信息在这个头文件里面





## 4.2 UART 配置

使用到的函数如下:

初始化函数:

`ql_uart_init(UART_PORT1)`

配置主串口, UART\_PORT1 是主串口, VIRTUAL\_PORT 是虚拟串口, 这个 VIRTUAL\_PORT 主要是和 AT 指令使用的。

`ql_uart_open(UART_PORT1, 9600, uart1_recieve_handle)`

第一个是选择的串口, 第二个波特率, 建议使用 9600, 至于其他的波特率, 能否配置, 第三个是回调函数, 发生接收后会在那里处理。

`ql_uart_write(UART_PORT1,buffer,data_len);`

这个是发送函数, 第一个是选择的串口, 第二个是数据, 第三个是长度, 不过移远工程师那边帮我构建一个类似于 printf 函数一样的函数, APP\_DEBUG, 格式和 printf 一样, 不过不能打印浮点数, 这个必须自己把代码添加, 否则不能使用。

```
39
40 #ifdef APP_DEBUG
41 #undef APP_DEBUG
42 #endif
43
44 #define DEBUG_ENABLE 1
45 #if DEBUG_ENABLE > 0
46 #define DEBUG_PORT UART_PORT1
47 #define DBG_BUF_LEN 512
48 static char DBG_BUFFER[DBG_BUF_LEN];
49 #define APP_DEBUG(FORMAT,...) {\
50     memset(DBG_BUFFER, 0, DBG_BUF_LEN);\
51     sprintf(DBG_BUFFER,FORMAT,##__VA_ARGS__); \
52     if (UART_PORT1 != (DEBUG_PORT)) \
53     {\
54         QDEBUG_NORMAL(DBG_BUFFER);\
55     } else {\
56         ql_uart_write((DEBUG_PORT), (uint8*)(DBG_BUFFER), strlen((const char *) (DBG_BUFFER))); \
57     } \
58 }
59 #else
60 #define QDEBUG_NORMAL(FORMAT,...)
61 #endif
62
```

具体使用方式请参考例子和说明文档。



## 6. 贡献列表

QUECTEL

移远

中国电信  
CHINA TELECOM  
世界触手可及

中国电信广东研究院



HUAWEI LiteOS

## 六.技术支持

FANCONN™ 开源 NBIOT 开发平台的发展得到了众多产业链伙伴的支持, 这个清单正在不断壮大中。FANCONN™ 物联将与合作伙伴共同为广大开发者提供完善的服务支持。

模组问题技术支持: [quectel@fanconn.com](mailto:quectel@fanconn.com)

硬件设计支持: [pcb@fanconn.com](mailto:pcb@fanconn.com)

软件开发支持: [soft@fanconn.com](mailto:soft@fanconn.com)

云平台技术支持: [cloud@fanconn.com](mailto:cloud@fanconn.com)

ODM 设计生产服务: [odm@fanconn.com](mailto:odm@fanconn.com)

射频天线测试设计服务: [ant@fanconn.com](mailto:ant@fanconn.com)

产品购买服务: [buy@fanconn.com](mailto:buy@fanconn.com)

传感器选型支持: [sensor@fanconn.com](mailto:sensor@fanconn.com)

NBIOT 电池选型: [battery@fanconn.com](mailto:battery@fanconn.com)