

任务：

调通以太网，跑一遍网络例程；

作业 1，连接阿里云

挑战：（通过麦克风）或者事先录制好一段语音，通过百度语义识别功能，获取语音识别结果，根据结果来控制 IO，或者是 LED 灯；

开发环境说明：基于 stm32f103zet6 使用 E:\RTT\bsp\stm32f10x env 0.7.0

参考资料：

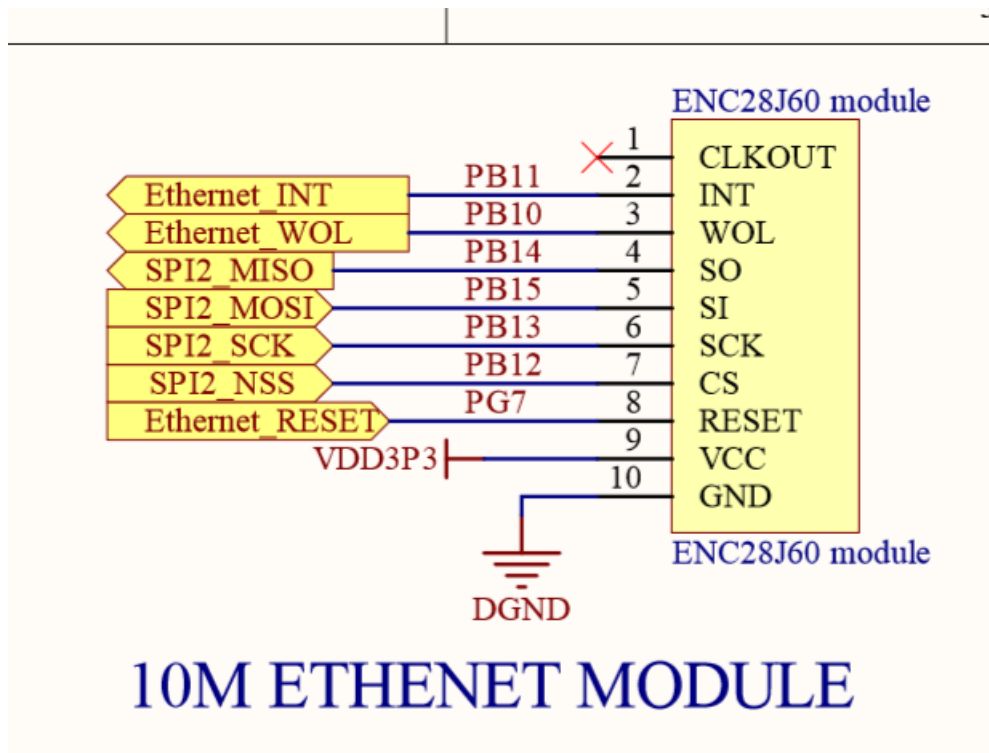
1. <https://www.rt-thread.org/document/site/rtthread-application-note/components/network/an0010-rtthread-system-lwip-driver-porting/> 网络协议栈驱动移植笔记
2. <https://www.rt-thread.org/document/site/rtthread-application-note/driver/spi/an0004-rtthread-driver-spi/> spi 设备应用笔记
3. <https://blog.csdn.net/xukai871105/article/details/13931833> ENC28J60 学习笔记
4. https://detail.tmall.com/item.htm?id=19534495860&ali_refid=a3_430583_1006:1106005875:N:ENC28J60:d0891a036ee1395f163f6ea979bb3385&ali_trackid=1_d0891a036ee1395f163f6ea979bb3385&spm=a230r.1.14.3 enc28j60 购买链接
5. https://www.rt-thread.org/document/site/rtthread-development-guide/rtthread-manual-doc/zh/1chapters/15-chapter_lwip/ 编程指南中 lwip
6. <https://www.rt-thread.org/qa/forum.php?mod=viewthread&tid=6897&highlight=enc28j60> 3.0 版本的 stm32f10x 怎样通过 enc28j60 芯片使用 lwIP?
7. <https://www.rt-thread.org/qa/thread-1351-1-1.html> 例程里 enc28j60 bug
8. <https://www.rt-thread.org/qa/forum.php?mod=viewthread&tid=3614&highlight=enc28j60> enc28j60 驱动含 spi 驱动基于 RT-Thread 1.2.0 正式版 非常好用。
9. <https://github.com/RT-Thread-packages/ali-iotkit> 阿里云 iot kit
10. <https://ai.baidu.com/docs#/ASR-Android-SDK/top> 百度语音识别

问题总结：

1. 想要使用 SPI2,只能在代码中修改,无法通过配置解决。可以通过使用 HAL 库就能选择。
2. 发现 enc28j60 移植资料不多,可能这个芯片性能不好吧。
3. 串口出现乱码,先检查一下晶振是不是正确,比如之前改过 #define HSE_VALUE ((uint32_t)12000000) /*!< Value of the External oscillator in Hz */ 但新板子是 8M 的话,就需要重新改一下。这个在 stm32f10x.h 文件中。
4. 要使用 lwip 为什么会自动配置 SAL 呢?
5. 出现这个错误 No space in execution regions with .ANY selector matching mqtttdeserializepublish.o(.text)., 优化一下编译选项,比如使用 optimization Level 3; Use MicroLIB;
6. 网卡设备挂载成功以后,如果用 ifconfig 命令,没有 ip 地址的话,可能是网线的问题哦。昨天网线还可以用,不代表今天就还能用。可能你收拾网线时候,哪折了一下。2018.8.24 00:48 分。一个教训啊。
- 7.如果在串口输出模式,想切换回 msh 命令行模式,直接按回车就可以。

进展及结果：

1.先看一下对应的模块引脚。芯片简介。

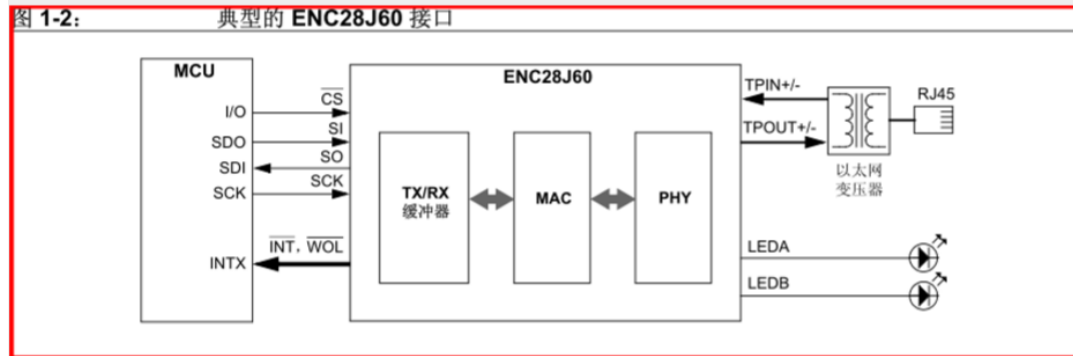


不用管 WOL 那个引脚，还是需要的。NSS 片选 是不是一直可以拉低。RESET 正常工作高电平。初始化时候可以先置低再置高。INT 中断引脚也可以不用。

7.2 ENC28J60 简介

ENC28J60 是带有行业标准串行外设接口（SerialPeripheral Interface, SPI）的独立以太网控制器。它可作为任何配备有 SPI 的控制器以太网接口。*ENC28J60* 符合 IEEE 802.3 的全部规范，采用了一系列包过滤机制以对传入数据包进行限制。它还提供了一个内部 DMA 模块，以实现快速数据吞吐和硬件支持的 IP 校验和计算。与主控制器的通信通过两个中断引脚和 SPI 实现，数据传输速率高达 10 Mb/s。两个专用的引脚用于连接 LED，进行网络活动状态指示。

下图所示为 ENC28J60 的简化框图。图 1-2 所示为使用该器件的典型应用电路。要将单片机连接到速率为 10 Mbps 的以太网，只需 ENC28J60、两个脉冲变压器和一些无源元件即可。本开发板中用的网络变压器的型号为 911105A。



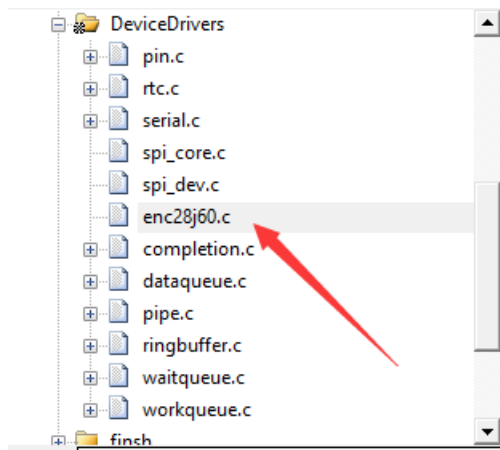
这里有提到 WOL 这个引脚。

2.配置 env spi 驱动中 enc28j60 lwip 文件系统 然后 scon -target=mdk5 -s

```

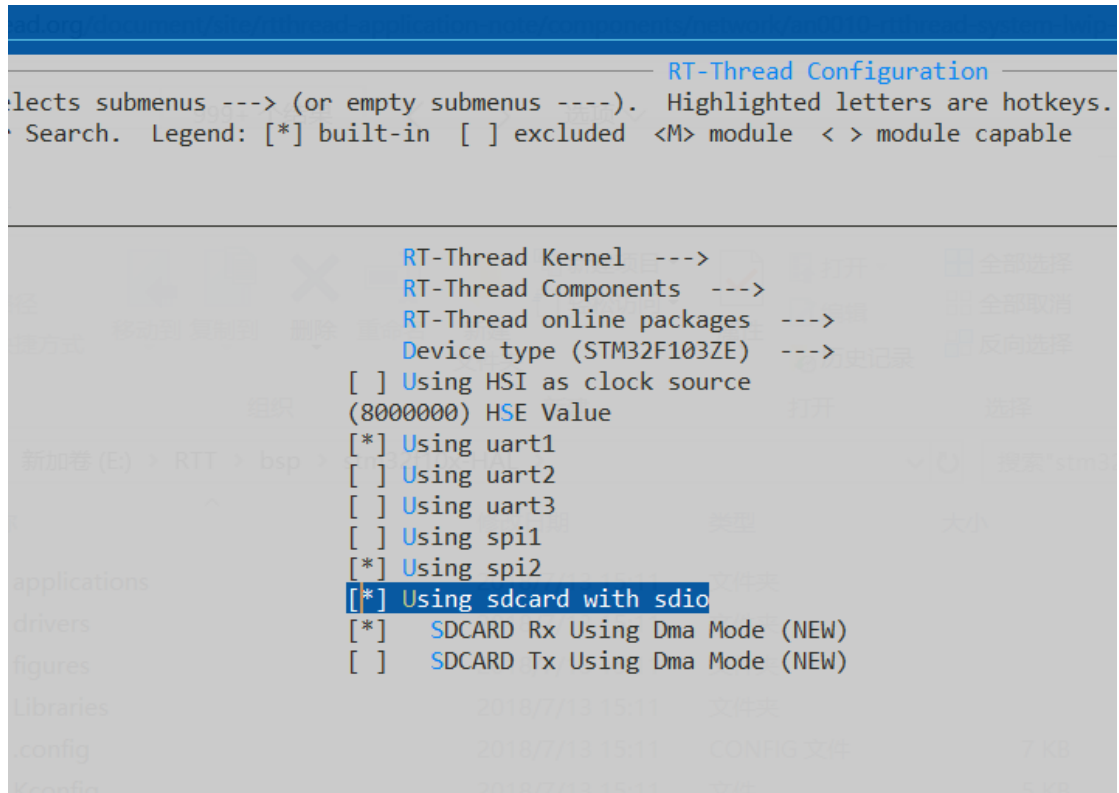
- *- Using device drivers IPC
(512) Set pipe buffer size
[*] Using serial device drivers
[ ] Using CAN device drivers
[ ] Using hardware timer device drivers
[ ] Enable CPU time for high resolution clock counter
[ ] Using I2C device drivers
[*] Using generic GPIO device drivers
[ ] Using PWM device drivers
[ ] Using MTD Nor Flash device drivers
[ ] Using MTD Nand Flash device drivers
- *- Using RTC device drivers
2 配 [ ] Using software simulation RTC device
[ ] Using NTP auto sync RTC time
[ ] Using SD/MMC device drivers
[*] Using SPI Bus/Device device drivers
[ ] Using SD/TF card driver with spi
[ ] Using Serial Flash Universal Driver
[ ] Using W25QXX SPI NorFlash
[ ] Using GD SPI NorFlash
[*] Using ENC28J60 SPI Ethernet network interface
[ ] Using RW009/007 SPI Wi-Fi wireless interface
[ ] Using Watch Dog device drivers
[ ] Using Wi-Fi network
[ ] Using Audio device drivers
Using USB --->

```



- 配置 shell 使用串口 1: 选中 Using UART1, 进入 RT-Thread Kernel ---> Kernel Device Object 菜单, 修改 the device name for console 为 uart1。
- 开启 SPI 总线及设备驱动并注册 SPI 总线到系统: 进入 RT-Thread Components ---> Device Drivers 菜单, 选中 Using SPI Bus/Device device drivers, RT-Thread Configuration 界面会默认选中 Using SPI1, spi1 总线设备会注册到操作系统。这里我们使用的是 SPI2.所以这里需要修改, 但再配置界面是无法修改的。

用 HAL 库就可以进行选择。但这次没有用。



- 开启 GPIO 驱动：进入 RT-Thread Components ---> Device Drivers 菜单，选中 Using generic GPIO device drivers。ENC28J60 模块 SPI 总线驱动也需要对片选管脚进行操作，都需要调用系统的 GPIO 驱动接口。

11. 引脚对应查看

```
/* SPI2 clock enable */
__HAL_RCC_SPI2_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();
/**SPI2 GPIO Configuration
PB13      -----> SPI2_SCK
PB14      -----> SPI2_MISO
PB15      -----> SPI2_MOSI
*/
GPIO_InitStruct.Pin = GPIO_PIN_13 | GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

GPIO_InitStruct.Pin = GPIO_PIN_14;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
```

不用 HAL 库。

9.RT-Thread 中 lwip, 网卡驱动, 如何联系起来的。

由此可知, RT-Thread 中将 lwIP 应用起来主要包括三个核心步骤: 1. 创建收发包线程, 调用接口 `eth_system_device_init()`。2. 提供网卡驱动, 调用网卡初始化函数, 注册网卡设备。(驱动不同相应的接口函数可能不同) 3. 初始化 lwIP, 创建 lwIP 线程, 调用接口 `lwip_sys_init()` (实际调用的 `lwip_system_init()`)。

至此, 三个步骤完成之后, 应用层便可以直接与外界通讯。

其中对第二步有疑问? 如何调用 `enc28j60` 的注册, 什么时候调用呢?

`eth_device_init()`调用 `eth_device_init_with_flag()`接口初始化网卡设备 (为网卡添加名称, IP、子网掩码、网关, 网卡设备使用的发包和收包接口函数等), 并向系统注册网卡设备。

到此, 解释了一个现象: 网卡驱动初始化和 lwIP 的初始化顺序互换并无影响。

就是说我没有发现调用 `rt_err_t enc28j60_attach(const char *spi_device_name)` 这个函数的地方。这个函数就是完成网卡设备注册的。

`eth_device_init(&(enc28j60_dev.parent), "e0");`

`git grep enc28j60_attach` 查看搜一下 `enc28j60_attach` 在别的 bsp 里的用法, 主要结构时 `enc28j60` 关联到 `spi device`, `spi device` 关联到 `spi bus`

```

dingo@DESKTOP-89J171S MINGW64 /e/RTT/bsp (master)
$ git grep enc28j60_attach
fh8620/platform/fh8620/iot_cam/board.c: enc28j60_attach(ENC28J60_SPI_DEV);
dingo@DESKTOP-89J171S MINGW64 /e/RTT/bsp (master)
$
dingo@DESKTOP-89J171S MINGW64 /e/RTT/bsp (master)
$ |

```



aozima 发表于 2018-3-15 15:30:07 | 只看该作者

沙发

要使用enc28j60芯片，还需要根据自己的板子，写一下SPI驱动，以及enc28j60芯片使用的中断引脚。

参考一下这个包看看是否有用

<https://www.rt-thread.org/qa/thread-3614-1-1.html>

12. 试着移植：

总结步骤就是：



改个名字@RT-Thread

- 1.配置CS引脚
- 2.将CS引脚挂载到spi总线设备，创建spi设备
- 3.配置 spi 设备的数据宽度，工作模式，频率
- 4.配置enc28j60的中断引脚的中断回调函数
- 5.挂载enc28j60设备enc28j60_attach

@改个名字@RT-Thread 解决了。参考论坛上一个帖子。对，就是这个思路，之前移植不知道，该做什么。



```

int rt_hw_enc28j60_init(void)
{
    enc28j60_attach("spi2");
    return 0;
}
INIT_DEVICE_EXPORT(rt_hw_enc28j60_init);

```

```

- RT - Thread Operating System
/ | \ 3.1.0 build Aug 21 2018
2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
(rt_object_get_type(&mutex->parent.parent) == RT_Object_Class_Mutex) assertion failed at function:rt_mutex_take, line number:6
\ | /
- RT - Thread Operating System
/ | \ 3.1.0 build Aug 21 2018
2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
(rt_object_get_type(&mutex->parent.parent) == RT_Object_Class_Mutex) assertion failed at function:rt_mutex_take, line number:6

```

这么做不成功。

<https://www.rt-thread.org/qa/thread-3614-1-1.html> 参考这个试一下。很有用。

这个很不错，把 spi 驱动都做好了。剩下的就是根据自己板子，换引脚。

注意：不要使用这个下载下来的 enc28j60.c 文件，直接使用 env 配置生成的文件。

enc28j60驱动含spi驱动基于RT-Thread 1.2.0正式版

2014-6-1 07:10[复制链接]

首先感谢anyang分享的enc28j60的新驱动

原帖地址: [分享一份enc28j60的新驱动](#)

此贴只是上贴的补充，以帮助对rt thread不熟的人

我下载后添加了spi的底层部分（从f107工程下复制的）

修改了platform.c下的自己spi引脚的定义，其他无变化，经测试可用，特分享下，以供需要的人！

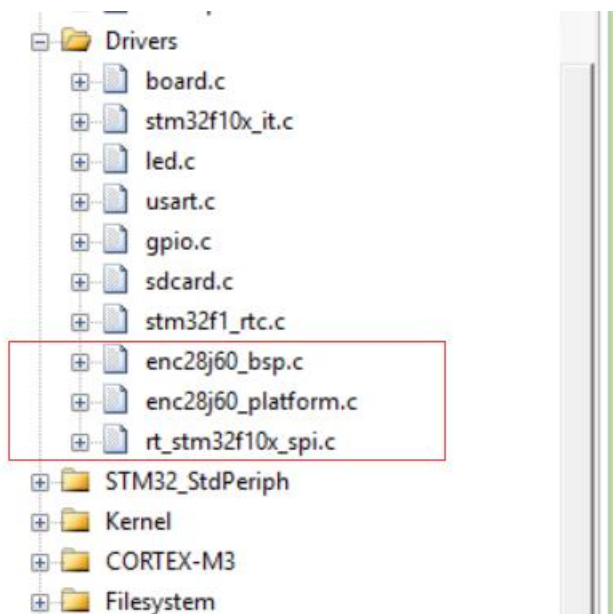
将文件替换1.2.0下的stm32f10x文件夹即可



[stm32f10x.rar](#)

1.51 MB, 下载次数: 739

把红框中的三个文件添加到 Drivers Group 下：



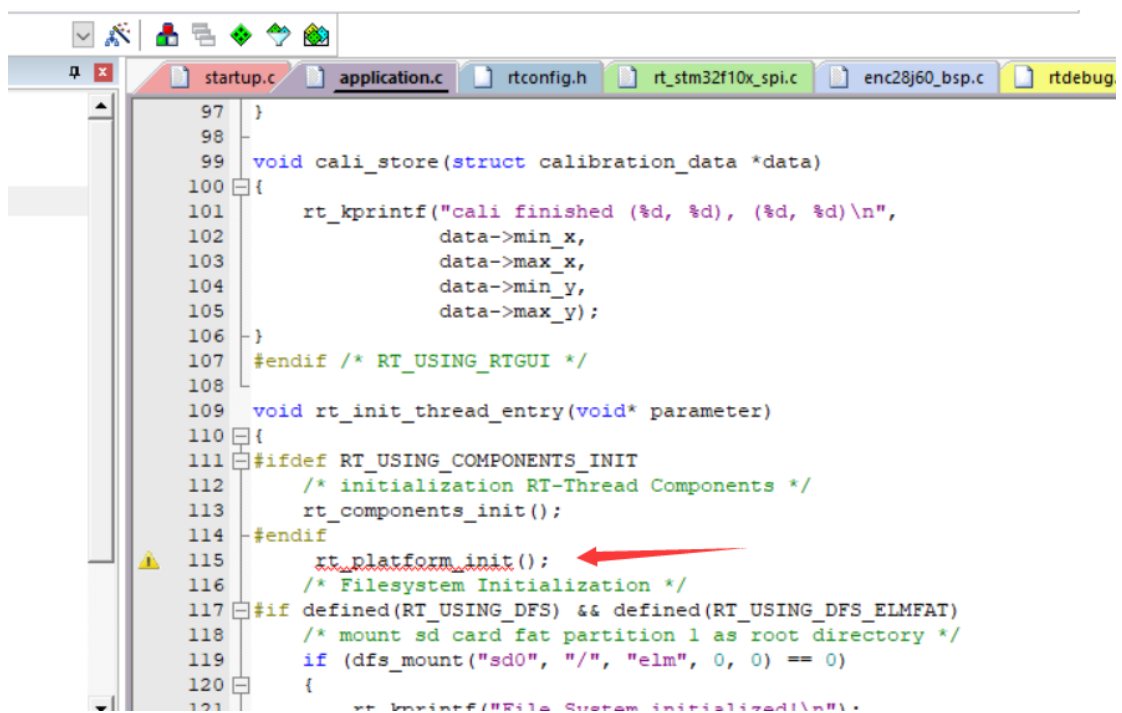
把 rt_platform_init()添加到 void rt_init_thread_entry(void* parameter) 中。

```

void rt_platform_init(void)
{
#ifdef RT_USING_SPI
    rt_hw_spi_init();
#endif // RT_USING_SPI

#ifdef RT_USING_LWIP
    /* initialize eth interface */
    //rt_hw_stm32_eth_init();
    rt_hw_enc28j60_init();
#endif /* RT_USING_LWIP */
}

```

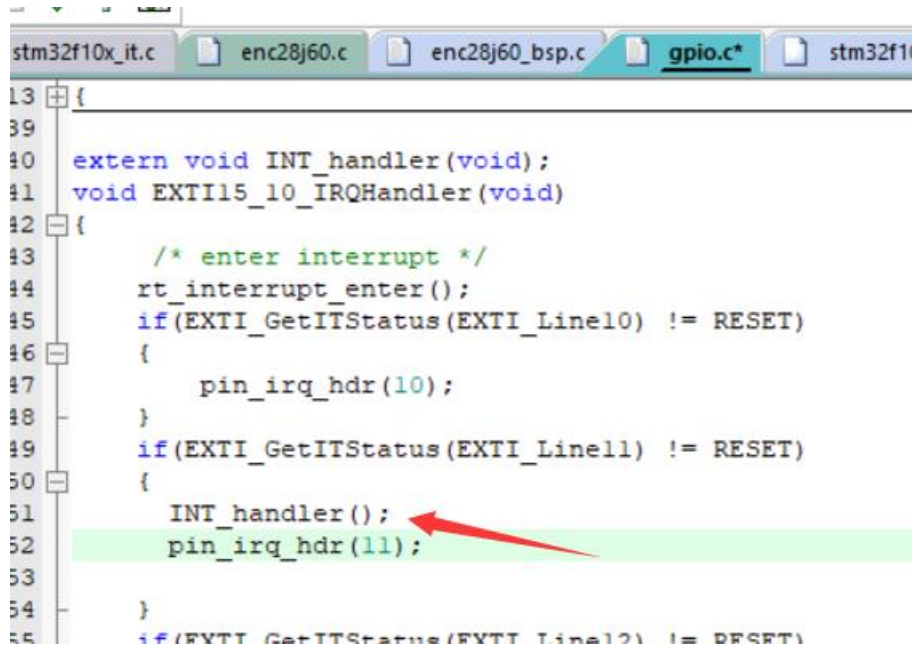


```

97 }
98
99 void cali_store(struct calibration_data *data)
100 {
101     rt_kprintf("cali finished (%d, %d), (%d, %d)\n",
102               data->min_x,
103               data->max_x,
104               data->min_y,
105               data->max_y);
106 }
107 #endif /* RT_USING_RTGUI */
108
109 void rt_init_thread_entry(void* parameter)
110 {
111 #ifdef RT_USING_COMPONENTS_INIT
112     /* initialization RT-Thread Components */
113     rt_components_init();
114 #endif
115     rt_platform_init();
116     /* Filesystem Initialization */
117 #if defined(RT_USING_DFS) && defined(RT_USING_DFS_ELMFAT)
118     /* mount sd card fat partition 1 as root directory */
119     if (dfs_mount("sd0", "/", "elm", 0, 0) == 0)
120     {
121         rt_kprintf("File System initialized\n");

```

在 gpio.c 中增加：
要注意这个。



```

stm32f10x_it.c  enc28j60.c  enc28j60_bsp.c  gpio.c*  stm32f1
13 {
39
40 extern void INT_handler(void);
41 void EXTI15_10_IRQHandler(void)
42 {
43     /* enter interrupt */
44     rt_interrupt_enter();
45     if(EXTI_GetITStatus(EXTI_Line10) != RESET)
46     {
47         pin_irq_hdr(10);
48     }
49     if(EXTI_GetITStatus(EXTI_Line11) != RESET)
50     {
51         INT_handler();
52         pin_irq_hdr(11);
53     }
54 }
55 if(EXTI_GetITStatus(EXTI_Line12) != RESET)

```

4 SPI设备驱动接口使用详解

按照前文的步骤，相信读者能很快的将RT-Thread SPI设备驱动运行起来，那么如何使用SPI设备驱动接口开发应用程序呢？

RT-Thread SPI设备驱动使用流程大致如下：

1. 定义SPI设备对象，调用 `rt_spi_bus_attach_device()` 挂载SPI设备到SPI总线。
2. 调用 `rt_spi_configure()` 配置SPI总线模式。
3. 使用 `rt_spi_send()` 等相关数据传输接口传输数据。

```

list_thread
thread  pri  status    sp      stack size max used left tick  error
-----
tshell   20  ready    0x0000013c 0x00001000    07%  0x00000008 000
tcpip    10  suspend  0x000000c0 0x00000400    59%  0x00000014 000
etx      12  suspend  0x00000090 0x00000400    22%  0x00000010 000
erx      12  suspend  0x00000090 0x00000400    52%  0x0000000f 000
mmcsd_de 22  suspend  0x00000098 0x00000400    14%  0x00000014 000
tidle    31  ready    0x00000044 0x00000100    35%  0x00000015 000
led      20  suspend  0x00000070 0x00000200    21%  0x00000005 000
msh />list_device
device      type          ref count
-----
e0          Network Interface  0
spi20       SPI Device      0
spi2        SPI Bus         0
sd0         Block Device     1
pin         Miscellaneous Device 0
uart3       Character Device  0
uart2       Character Device  0
uart1       Character Device  2
msh />

```

挂在上了。



12.试试跑跑例子：

Ifconfig

```
msh />ifconfig
network interface: e0 (Default)
MTU: 1500
MAC: 00 04 a3 12 34 56
FLAGS: UP LINK_UP ETHARP BROADCAST IGMP
ip address: 192.168.1.103
gw address: 192.168.1.1
net mask : 255.255.255.0
dns server #0: 211.100.225.34
dns server #1: 124.207.160.106
msh />
```

使用 tcp 客户端：

```
Configuration
+ miscellaneous packages + samples: RT-Thread kernel and components samples + network sample options ---
network sample options
menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Press:
, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] [network] tcp client
[*] [network] tcp server
[*] [network] udp client
[*] [network] udp server
[*] [network] tcp client by select api
[*] [network] http client
```

程序功能：接收并显示从服务端发送过来的信息，接收到开头是 'q' 或 'Q' 的信息退出程序

```
uart1 Character Device 2
msh />tcpclient 192.168.1.105 8080
[E/SAL_SOCKET] SAL socket protocol family input failed, return error -3.
Socket error
msh />help
RT-Thread shell commands:
tcpclient - a tcp client sample
```

出现这个错误，需要重新配置一下 SAL，选择其中支持 LWIP.

5.1 lwIP 协议栈注册

如果开启 lwIP 支持，需要在 SAL 组件中对 lwIP 协议栈的进行注册初始化，主要是在 `sal_socket/proto/lwip/af_inet_lwip.c` 文件中完成，如果文件中已经完成自动初始化则可忽略，如果没有则需要调用如下初始化函数：

```
int lwip_inet_init(void);
```

是不是这个意思，开启 lwip，就需要配置这个组件。

```

C:\Users\dingo>ping 192.168.1.103

正在 Ping 192.168.1.103 具有 32 字节的数据:
请求超时。
来自 192.168.1.103 的回复: 字节=32 时间=986ms TTL=255
来自 192.168.1.103 的回复: 字节=32 时间=3ms TTL=255
来自 192.168.1.103 的回复: 字节=32 时间=3ms TTL=255

192.168.1.103 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 3, 丢失 = 1 (25% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 3ms, 最长 = 986ms, 平均 = 330ms

C:\Users\dingo>ping 192.168.1.103

正在 Ping 192.168.1.103 具有 32 字节的数据:
来自 192.168.1.103 的回复: 字节=32 时间=4ms TTL=255
来自 192.168.1.103 的回复: 字节=32 时间=4ms TTL=255
来自 192.168.1.103 的回复: 字节=32 时间=7ms TTL=255
来自 192.168.1.103 的回复: 字节=32 时间=3ms TTL=255

192.168.1.103 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间(以毫秒为单位):
        最短 = 3ms, 最长 = 7ms, 平均 = 4ms

```

从电脑上可以 ping 通，但为什么没有反应呢在板子上。本来就没有反应啊，需要服务器端发送数据才显示。

```

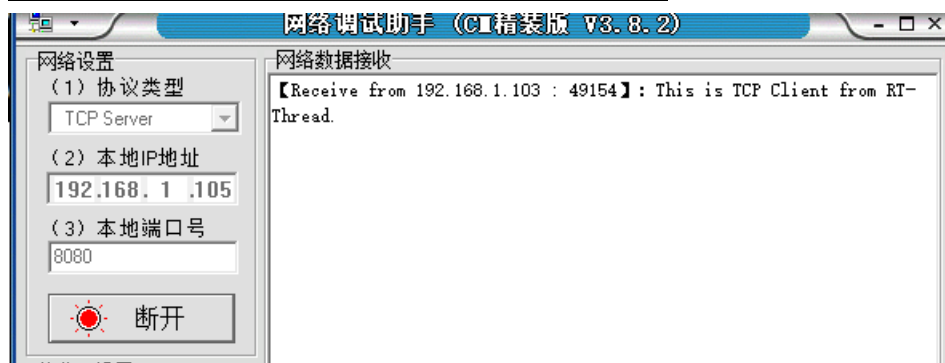
\ | /
- RT -   Thread Operating System
/ | \   3.1.0 build Aug 23 2018
2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
found part[0], begin: 32256, size: 940.992MB
[I/SAL_SOCKET] Socket Abstraction Layer initialize success.
msh />File System initialized!

msh />ifconfig
network interface: e0 (Default)
MTU: 1500
MAC: 00 04 a3 12 34 56
FLAGS: UP LINK UP ETHARP BROADCAST IGMP
ip address: 192.168.1.103
gw address: 192.168.1.1
net mask : 255.255.255.0
dns server #0: 211.100.225.34
dns server #1: 124.207.160.106
msh />tcpclient 192.168.1.105 8080

```



```
msh />tcpclient 192.168.1.105 8080
received error,close the socket.
msh />tcpclient 192.168.1.105 8080
Received data = http://www.cmsoft.cn QQ:10865600
```

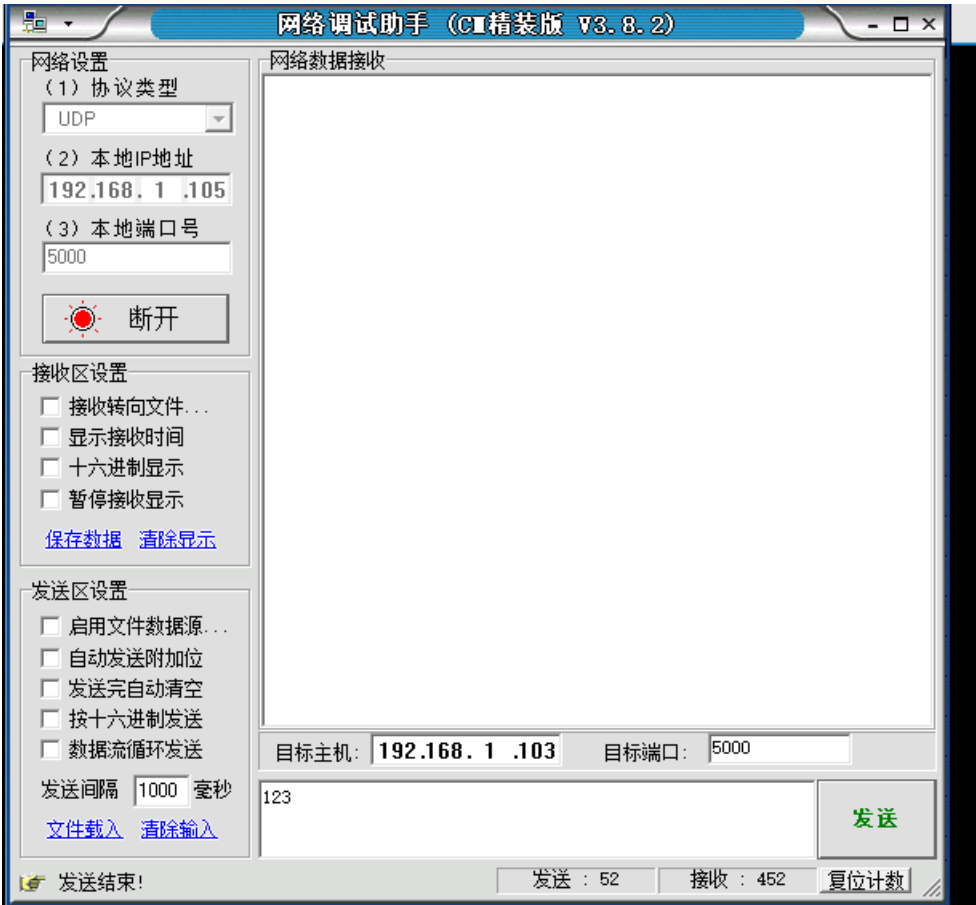


```
Received data = http://www.cmsoft.cn QQ:10865600
got a 'q' or 'Q',close the socket.
```

使用 tcp 服务器

程序功能：作为一个服务端，接收并显示客户端发来的数据，接收到 exit 退出程序

使用 udp 服务器



这个软件下面可以选目标主机。直接连接就可以。

```
msh />udpserv
UDPServer Waiting for client on port 5000...

(192.168.1.105 , 5000) said : 123
(192.168.1.105 , 5000) said : 123hello kkkd
```

13 连接阿里云

<https://github.com/RT-Thread-packages/ali-iotkit> 参考链接。

依赖：[MbedTLS 软件包](#)

三元组	<p>三元组指ProductKey、DeviceName、DeviceSecret。</p> <ul style="list-style-type: none">• ProductKey：是物联网平台为产品颁发的全局唯一标识。该参数很重要，在设备认证以及通信中都会用到，因此需要您保管好。• DeviceName：在注册设备时，自定义的或自动生成的设备名称，具备产品维度内的唯一性。该参数很重要，在设备认证以及通信中都会用到，因此需要您保管好。• DeviceSecret：物联网平台为设备颁发的设备密钥，和DeviceName成对出现。该参数很重要，在设备认证时会用到，因此需要您保管好并且不能泄露。
-----	---

RRPC 全称：Revert-RPC。可以实现由服务端请求设备端并能够使设备端响应的功能。

步骤：1.控制台添加信息。

https://help.aliyun.com/document_detail/68946.html?spm=a2c4g.11186623.6.553.3e143f69tHhH4s

RTT_ARM_DIY

基础版

ProductKey: a1OjDzGZFa 复制

ProductSecret: ***** 显示

设备数: 0 前往管理

产品信息消息通信服务订阅日志服务

产品信息

产品名称	RTT_ARM_DIY	节点类型	设备	设备数	0 前往管理
产品版本	基础版	创建时间	2018/08/23 22:23:56		
动态注册	<div>已关闭</div>	ProductKey	a1OjDzGZFa 复制	ProductSecret	QrKI7oAd3t0kZTP1 重置 隐藏
产品描述:	TEST ALIYUN				

RTT_ARM_DIY

基础版

ProductKey: a1OjDzGZFa 复制

ProductSecret: ***** 显示

设备数: 0 前往管理

产品信息消息通信服务端订阅日志服务

Topic列表

Topic类	操作权限	描述	操作
/a1OjDzGZFa/\${deviceName}/data	发布和订阅		编辑 删除
/a1OjDzGZFa/\${deviceName}/update	发布		编辑 删除
/a1OjDzGZFa/\${deviceName}/update/error	发布		编辑 删除
/a1OjDzGZFa/\${deviceName}/get	订阅		编辑 删除

查看设备证书

设备证书用于云端对接入的设备做鉴权认证，请妥善保管！

ProductKey	a1OjDzGZFa 复制
DeviceName	stm320001 复制
DeviceSecret	***** 显示

一键复制 关闭

ProductKey: a1OjDzGZFa
DeviceName: stm320001
DeviceSecret: aX3lQUfjumvmh46DWwGmRI7sKJJkfkJL

配置 ENV

```
things → IoT Cloud → Ali-iotkit: Ali Cloud SDK for IoT platform
-- Ali-iotkit: Ali Cloud SDK for IoT platform
s submenus ---> (or empty submenus ----). Highlighted letters are hotkeys.
for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

-- Ali-iotkit: Ali Cloud SDK for IoT platform
  Select Aliyun platform (LinkDevelop Platform) --->
(a1dSQSGZ77X) Config Product Key
(stm320001) Config Device Name
(aX3lQUfjumvmh46DwwGmRI7sKJJkfkJL) Config Device Secret
-- Enable MQTT
[*] Enable MQTT sample (NEW)
[*] Enable MQTT direct connect (NEW)
-- Enable SSL
[ ] Enable COAP (NEW)
[*] Enable OTA
  Select OTA channel (Use MQTT OTA channel) --->
  Version (latest) --->
```

增加 mbedTLS 帧大小 (OTA 的时候至少需要 8K 大小) 平台就选 Develop 就可以。

```
packages mbedtls:An open source, portable, easy to use, readable and flexible SSL library
mbedtls:An open source, portable, easy to use, readable and flexible SSL library
r> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> e
lp, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- mbedtls:An open source, portable, easy to use, readable and flexible SSL library
[*] Store the AES tables in ROM
(2) Maximum window size used
(8192) Maximum fragment length in bytes
[ ] Enable a mbedtls client example
[ ] Enable Debug log output
  version (latest) --->
```

开启 OTA 功能必须使能加密连接 (因为 OTA 升级必须使用 HTTPS 下载固件)

```
Enable the auto update option,env will auto update the packages you select.
Start to remove E:\RTT\bsp\stm32f10x\packages\webclient-v1.0.0, please wait...

Start to remove E:\RTT\bsp\stm32f10x\packages\samples-latest, please wait...

The folder is managed by git. Do you want to delete this folder?

Press the Y Key to delete the folder or just press Enter to keep them :Y
Folder has been removed.
Cloning into 'E:\RTT\bsp\stm32f10x\packages\ali-iotkit-latest'...
remote: Counting objects: 835, done.
remote: Compressing objects: 100% (667/667), done.
Receiving objects: 3% (33/835), 372.00 KiB | 49.00 KiB/s
```

配置完以后 `scons -target=mdk5 -s`

然后打开 mdk 编译。

```

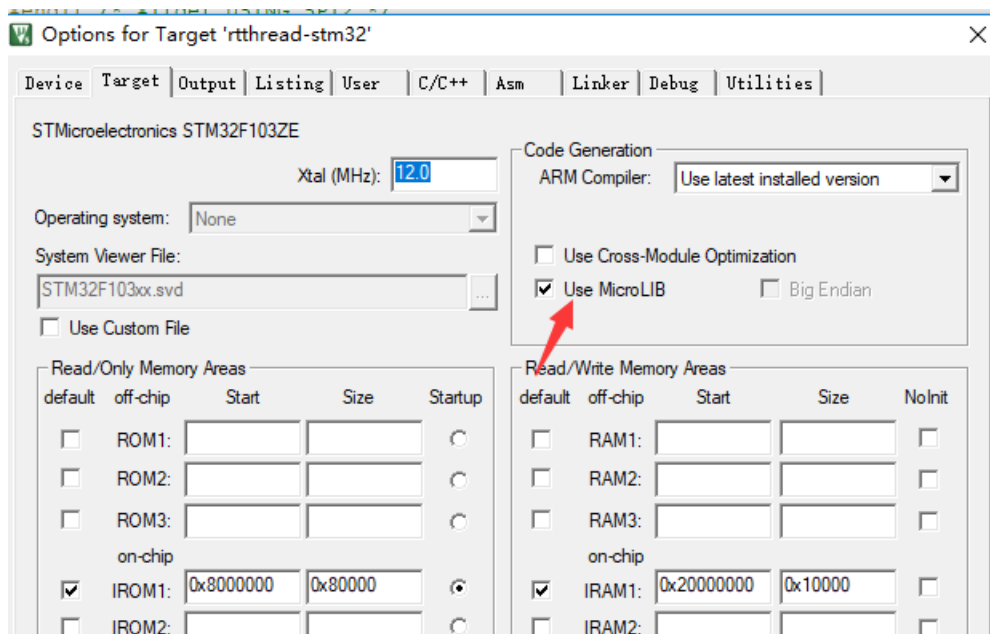
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching sal_socket.o(.rti_fn.4).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching af_inet_lwip.o(.rti_fn.4).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching hguard.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching getc.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching aeabi_ldiv0_sigfpe.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching printf1.o(x$fp1$printf1).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching printf2.o(x$fp1$printf2).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching libshutdn2.o(.ARM.Collect$S$1).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching use_no_semi.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching usenofp.o(x$fp1$usenofp).
.\build\rtthread-stm32.axf: Error: L6407E: Sections of aggregate size 0x6520 bytes could not fit into .ANY selector(s).
Not enough information to list image symbols.
Not enough information to list the image map.
Finished: 2 information, 0 warning and 282 error messages.
".\build\rtthread-stm32.axf" - 282 Error(s), 69 Warning(s).
Target not created.

.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching utils_http.o(.constdata).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching kservice.o(.constdata).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching sdio.o(.constdata).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching mem_stats.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching freadfast.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching mqtttdeserializepublish.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching lc_ctype_c.o(locale$S$data).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching cmd.o(F$SymTab).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching ripemd160.o(.conststring).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching mqtttunsubscribeclient.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching hmac_dbg.o(.constdata).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching netifapi.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching dhm.o(.constdata).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching fseek.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching enc28j60_platform.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching context_rvds.o(.text).
.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching finish_error.o(.conststring).

```

.\build\rtthread-stm32.axf: Error: L6406E: No space in execution regions with .ANY selector matching mqtttdeserializepublish.o(.text).

<http://www.openedv.com/posts/list/42541.htm> 解决办法
stm32zet6 512kB Flash, 64KB RAM 如何优化, 改红色标识部分。

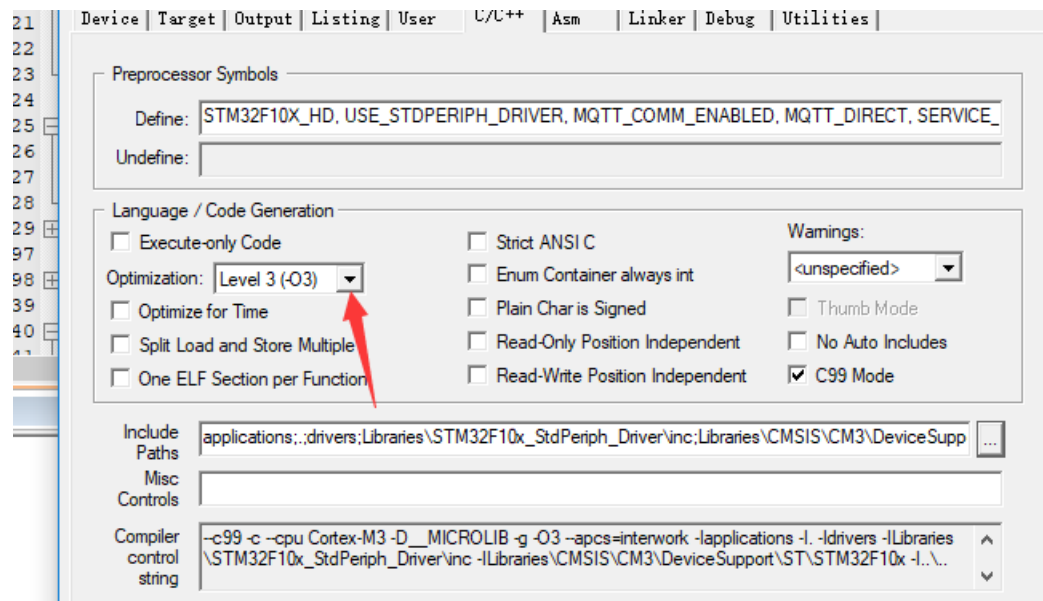


<https://blog.csdn.net/kelsey11/article/details/51246636>

选上“Use MicroLIB”这是 KEIL 自带的一个简易的库,例如你用 printf 函数的时候,就会从串口 1 输出字符串,直接默认定向到串口 1

2.microlib 是缺省 C 库的备选库。它用于必须在极少量内存环境下运行的深层嵌入式应用程序。这些应用程序不在操作系统中运行。microlib 不会尝试成为符合标准的 ISO C 库。

microlib 进行了高度优化以使代码变得很小。它的功能比缺省 C 库少，并且根本不具备某些 ISO C 特性。某些库函数的运行速度也比较慢，例如，memcpy()。



```
compiling net_sockets.c...
compiling af_inet_lwip.c...
compiling dfs_net.c...
linking...
Program Size: Code=360108 RO-data=78492 RW-data=1744 ZI-data=52536
After Build - User command #1: fromelf --bin .\build\rtthread-stm32.axf --output rtthread.bin
".\build\rtthread-stm32.axf" - 0 Error(s), 69 Warning(s).
```

再点 ONE ELF Section per Function 不用 MicroLIB 确实比之前少了。

```
compiling net_sockets.c...
compiling af_inet_lwip.c...
compiling dfs_net.c...
linking...
Program Size: Code=342552 RO-data=71736 RW-data=1692 ZI-data=53476
After Build - User command #1: fromelf --bin .\build\rtthread-stm32.axf --output rt
".\build\rtthread-stm32.axf" - 0 Error(s), 69 Warning(s).
```

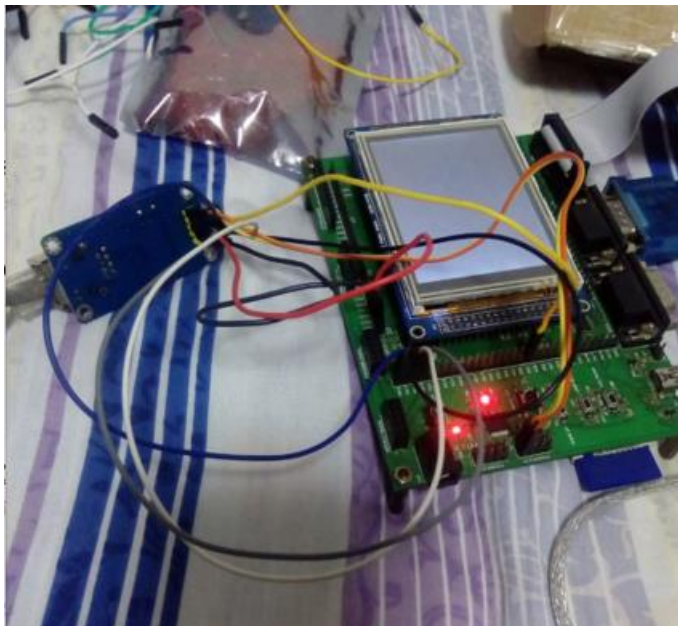
编译下载。

插上 enc28j60 以后，网口指示灯应该要亮的。不亮就是有问题。

```

\ | /
- RT -   Thread Operating System
/ | \   3.1.0 build Aug 23 2018
2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
found part[0], begin: 32256, size: 940.992MB
[I/SAL SOC] Socket Abstraction Layer initialize s
msh />File System initialized!
list_device
device      type      ref count
-----
e0          Network Interface  0
spi20       SPI Device    0
spi2        SPI Bus      0
sd0         Block Device   1
pin         Miscellaneous Device 0
uart2       Character Device 0
uart1       Character Device 2
msh />ifconfig
network interface: e0 (Default)
MTU: 1500
MAC: 00 04 a3 12 34 56
FLAGS: UP LINK_DOWN ETHARP BROADCAST IGMP
ip address: 0.0.0.0
gw address: 0.0.0.0
net mask : 0.0.0.0
dns server #0: 0.0.0.0
dns server #1: 0.0.0.0
msh />

```



重新接线也是不亮灯。

最后查明原因是网线的问题。坑人啊。

```

- RT - Thread Operating System
/ | \ 3.1.0 build Aug 23 2018
2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
found part[0], begin: 32256, size: 940.992MB
[1/SAL_S0C] Socket Abstraction Layer initialize
msh />File System initialized!
ifconfig
network interface: e0 (Default)
MTU: 1500
MAC: 00 04 a3 12 34 56
FLAGS: UP LINK UP ETHARP BROADCAST IGMP
ip address: 192.168.1.107
gw address: 192.168.1.1
net mask : 255.255.255.0
dns server #0: 211.100.225.34
dns server #1: 124.207.160.106
msh />

```

```

msh />ali_mqtt_test
Input param error! Example: ali_mqtt_test start/stop or ali_mqtt_test pub open/close
msh />ali_mqtt_test start
ali_mqtt_test: command not found.
msh />

```

stm320001 未激活

产品: RTT_ARM_DIY [查看](#) ProductKey: a1OjDzGZF_a [复制](#) DeviceSecret: ***** [显示](#)

[设备信息](#) [Topic列表](#) [设备影子](#)

设备信息					
产品名称	RTT_ARM_DIY	ProductKey	a1OjDzGZF _a 复制	区域	华东 2
节点类型	设备	DeviceName	stm320001 复制	DeviceSecret	***** 显示
当前状态	未激活	IP地址	-	固件版本	-
添加时间	2018/08/23 22:41:03	激活时间		最后上线时间	

一直未激活？什么意思啊？

```

msh />ali_mqtt_test start
msh />ali_mqtt_test pub
Input param error! Example: ali_mqtt_test start/stop or ali_mqtt_test pub open/close
msh />ali_mqtt_test pub open
param:open
ali_mqtt_test_pub|568 :: error occur when publish
msh />ali_mqtt_test stop
msh />ali_matt_test start
ali_matt_test: command not found.
msh />ali_mqtt_test start
msh />ali_mqtt_test pub open
param:open
ali_mqtt_test_pub|568 :: error occur when publish
msh />

```

第二天在公司 用 qemu 虚拟机来实现。配置还是之前的配置，可以运行。但发现一个问题，之前中是 Ping 不通，说是没有这个指令。是因为没有开启网络小工具。会不会是这个原因呢？晚上回去试试。

```
----- IoT - internet of things -----
menus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> module
Legend: [*] built-in [ ] excluded <M> module < > module capable

[ ] Paho MQTT: Eclipse Paho MQTT C/C++ client for Embedded platforms ----
[ ] WebClient: A HTTP/HTTPS Client for RT-Thread ----
[ ] mongoose: Embedded Web Server / Embedded Networking Library ----
[ ] WebTerminal: Terminal runs in a Web browser ----
[ ] cJSON: Ultralightweight JSON parser in ANSI C ----
[ ] jsmn: is a minimalistic JSON parser in C ----
[ ] ljson: JSON parser in ANSI C ----
[ ] ezXML is a C library for parsing XML documents. ----
[ ] nanopb: Protocol Buffers for Embedded Systems ----
[ ] Wi-Fi --->
[ ] CoAP: A C implementation of the Constrained Application Protocol ----
[ ] nopoll: A OpenSource WebSocket implementation (RFC 6455) in ansi C ----
[*] netutils: Networking utilities for RT-Thread --->
[ ] AT DEVICE: RT-Thread AT component porting or samples for different device ----
IoT Cloud --->
```

```
Specify the 'raw' format explicitly to remove the restrictions.

\ | /
- RT - Thread Operating System
/ | \ 3.1.0 build Aug 24 2018
2006 - 2018 Copyright by rt-thread team
lwIP-2.0.2 initialized!
[I/SAL_SOC] Socket Abstraction Layer initialize success.
SD card capacity 65536 KB
probe mmcblk0 block device!
found part[0], begin: 32256, size: 63.992MB
file system initialization done!
hello rt-thread
msh />list_thread
thread pri status sp stack size max used left tick error
-----
tshell 20 ready 0x000001b0 0x00001000 10% 0x00000003 000
aio 16 suspend 0x00000080 0x00000800 06% 0x0000000a 000
mmcsd_de 22 suspend 0x000000b0 0x00000400 53% 0x0000000c 000
tcpip 10 suspend 0x000000e8 0x00000800 30% 0x00000006 000
etx 12 suspend 0x000000a8 0x00000400 16% 0x00000010 000
erx 12 suspend 0x000000b0 0x00000400 57% 0x00000008 000
tidle 31 ready 0x0000005c 0x00000200 28% 0x00000011 000
timer 4 suspend 0x00000078 0x00000400 11% 0x00000009 000
msh />ali_mqtt_test start
iotkit-embedded sdk version: V2.10
[inf] iotx_device_info_init(40): device_info created successfully!
[dbg] iotx_device_info_set(50): start to set device info!
[dbg] iotx_device_info_set(64): device_info set successfully!
[dbg] guider_print_dev_guider_info(271): .....
[dbg] guider_print_dev_guider_info(272): ProductKey : a10j1DzGZFa
[dbg] guider_print_dev_guider_info(273): DeviceName : stm320001
[dbg] guider_print_dev_guider_info(274): DeviceID : a10j1DzGZFa.stm320001
[dbg] guider_print_dev_guider_info(276): .....
[dbg] guider_print_dev_guider_info(277): PartnerID Buf : ,partner_id=example.demo.partner-id
[dbg] guider_print_dev_guider_info(278): ModuleID Buf : ,module_id=example.demo.module-id
[dbg] guider_print_dev_guider_info(279): Guider URL :
[dbg] guider_print_dev_guider_info(281): Guider SecMode : 2 (TLS + Direct)
[dbg] guider_print_dev_guider_info(283): Guider Timestamp : 252460800000
[dbg] guider_print_dev_guider_info(284): .....
[dbg] guider_print_dev_guider_info(290): .....
[dbg] guider_print_conn_info(248): .....
[dbg] guider_print_conn_info(249): Host : a10j1DzGZFa.iot-as-mqtt.cn-shanghai.aliyuncs.com
[dbg] guider_print_conn_info(250): Port : 1883
[dbg] guider_print_conn_info(253): ClientID : a10j1DzGZFa.stm320001|securemode=2,timestamp=252460800000|example.module-id|
[dbg] guider_print_conn_info(255): TLS PubKey : 0x60127554 ('-----BEGIN CERTI ...')
```


[illegible]

设备管理 > 设备详情

stm320001 在线

产品: RTT ARM DIY [查看](#)

ProductKey: a1OjIDzGZFfa [复制](#)

设备信息 Topic列表 设备影子

在线了。

[illegible]

RTT ARM DIY

ProductKey: a1OjDzGZFa [复制](#)

ProductSecret: ***** 显示

设备数: 1 [前往管理](#)

产品信息

消息通信

服务端订阅

日志服务

日志服务 日志以英文展示，详细的中英文对照请参考[文档](#)

设备行为分析

上行消息分析

下行消息分析

1小时

2018-08-24 09:03:07

- 2018-08-24 10:03:07

收起

全部状态

0457

2

时间	MessageID	DeviceName	内容(全部)	状态以及原因分析
2018/08/24 09:56:23	1032808808734351361	stm320001	Publish message to topic/sys/a1Oj...	成功

MQTT Sample 单次发布订阅

想要单次发布订阅，按 `ctrl + C` 退出终端，重新进入，就卡在这里，网页上也不显示在线。网友说按回车就可以切换回 `msh` 命令行模式。

```
signed using      : RSA with SHA1
RSA key size     : 2048 bits
basic constraints : CA=true
key usage        : Key Cert Sign, CRL Sign
[inf] _ssl_parse crt(143): crt content:451
[inf] _ssl_client_init(183): ok (0 skipped)
[inf] _TLSConnectNetwork(345): Connecting to /a1ojldzgza.iot-as-mqtt.cn-shanghai.aliyuncs.com/1883...
msh />[inf] _TLSConnectNetwork(358): ok
[inf] _TLSConnectNetwork(363): . Setting up the SSL/TLS structure...
[inf] _TLSConnectNetwork(373): ok
[inf] _TLSConnectNetwork(408): Performing the SSL/TLS handshake...
```



```

msh />ali_mqtt_test start
iotkit-embedded sdk version: V2.10
[inf] iotx_device_info_init(40): device_info created successfully!
[dbg] iotx_device_info_set(50): start to set device info!
[dbg] iotx_device_info_set(64): device_info set successfully!
[dbg] guider_print_dev_guider_info(271): .....
[dbg] guider_print_dev_guider_info(272): ProductKey : a10jldzGZFa
[dbg] guider_print_dev_guider_info(273): DeviceName : stm320001
[dbg] guider_print_dev_guider_info(274): DeviceID : a10jldzGZFa.stm320001
[dbg] guider_print_dev_guider_info(276): .....
[dbg] guider_print_dev_guider_info(277): PartnerID Buf : ,partner_id=example.demo.partner-id
[dbg] guider_print_dev_guider_info(278): ModuleID Buf : ,module_id=example.demo.module-id
[dbg] guider_print_dev_guider_info(279): Guider URL :
[dbg] guider_print_dev_guider_info(281): Guider SecMode : 2 (TLS + Direct)
[dbg] guider_print_dev_guider_info(283): Guider Timestamp : 2524608000000
[dbg] guider_print_dev_guider_info(284): .....
[dbg] guider_print_dev_guider_info(290): .....
[dbg] guider_print_conn_info(248): .....
[dbg] guider_print_conn_info(249): Host : a10jldzGZFa.iot-as-mqtt.cn-shanghai.aliyuncs.com
[dbg] guider_print_conn_info(250): Port : 1883
[dbg] guider_print_conn_info(253): ClientID : a10jldzGZFa.stm320001|securemode=2,timestamp=2524608000000,sig
|
[dbg] guider_print_conn_info(255): TLS PubKey : 0x60127554 ('-----BEGIN CERTI ...')
[dbg] guider_print_conn_info(258): .....
host: a10jldzgza.iot-as-mqtt.cn-shanghai.aliyuncs.com
[inf] iotx_mc_init(1703): MQTT init success!
[inf] _ssl_client_init(175): Loading the CA root certificate ...
cert. version : 3
serial number : 04:00:00:00:00:01:15:4B:5A:C3:94
issuer name : C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
subject name : C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
issued on : 1998-09-01 12:00:00
expires on : 2028-01-28 12:00:00
signed using : RSA with SHA1
RSA key size : 2048 bits
basic constraints : CA=true
key usage : Key Cert Sign, CRL Sign
[inf] _ssl_parse_cert(143): crt content:451
[inf] _ssl_client_init(183): ok (0 skipped)
[inf] _TLSConnectNetwork(345): Connecting to /a10jldzgza.iot-as-mqtt.cn-shanghai.aliyuncs.com/1883...
msh />[inf] _TLSConnectNetwork(358): ok
[inf] _TLSConnectNetwork(363): . Setting up the SSL/TLS structure...
[inf] _TLSConnectNetwork(373): ok
[inf] _TLSConnectNetwork(408): Performing the SSL/TLS handshake...

```

再重启就是连接失败。

```

[inf] _TLSConnectNetwork(373): ok
[inf] _TLSConnectNetwork(408): Performing the SSL/TLS handshake...
[inf] _TLSConnectNetwork(412): failed ! mbedtls_ssl_handshake returned -0x7280
[inf] _network_ssl_disconnect(514): ssl_disconnect
[inf] HAL_SSL_Destroy(530): handle is NULL
[err] iotx_mc_connect(2001): TCP or TLS Connection failed
[inf] HAL_SSL_Destroy(530): handle is NULL
[inf] iotx_mc_disconnect(2129): mqtt disconnect!
[inf] iotx_mc_release(2175): mqtt release!
mqtt_client|275 :: MQTT construct failed
[err] LITE_dump_malloc_free_stats(594): WITH_MEM_STATS = 0
mqtt_client|346 :: out of sample!
|

```

之前正常的是：

```

[inf] _TLSConnectNetwork(373): ok
[inf] _TLSConnectNetwork(408): Performing the SSL/TLS handshake...
[inf] _TLSConnectNetwork(416): ok
[inf] _TLSConnectNetwork(420): . Verifying peer X.509 certificate..
[inf] _real_confirm(92): certificate verification result: 0x00
[inf] iotx_mc_connect(2035): mqtt connect success!
[dbg] iotx_mc_report_mid(2259): MID Report: started in MQTT
[dbg] iotx_mc_report_mid(2276): MID Report: json data = '{"id":"a10jldzGZFa_stm320001_mid","params":{"_sys_device_mid":"example.demo.module-id","_sys_device_pid":"e
[dbg] iotx_mc_report_mid(2292): MID Report: topic name = /sys/a10jldzGZFa/stm320001/thing/status/update'
[dbg] iotx_mc_report_mid(2309): MID Report: finished, IOT_MQTT_Publish() = 0
[inf] iotx_mc_subscribe(1388): mqtt subscribe success,topic = /sys/a10jldzGZFa/stm320001/thing/service/property/set!
[inf] iotx_mc_subscribe(1388): mqtt subscribe success,topic = /sys/a10jldzGZFa/stm320001/thing/event/property/post_reply!
[dbg] iotx_mc_cycle(1269): SUBACK
event_handle|141 :: subscribe success, packet-id=0
[dbg] iotx_mc_cycle(1269): SUBACK
event_handle|141 :: subscribe success, packet-id=0
[inf] iotx_mc_heartbeat_sub(2226): send MQTT ping...
[inf] iotx_mc_cycle(1295): receive ping response!
[inf] iotx_mc_heartbeat_sub(2226): send MQTT ping...
[inf] iotx_mc_cycle(1295): receive ping response!

```

多重启几次就好了。

```
msh />[inf] _TLSConnectNetwork(358): ok
[inf] _TLSConnectNetwork(363): . Setting up the SSL/TLS structure...
[inf] _TLSConnectNetwork(373): ok
[inf] _TLSConnectNetwork(408): Performing the SSL/TLS handshake...
[inf] _TLSConnectNetwork(416): ok
[inf] _TLSConnectNetwork(420): . Verifying peer X.509 certificate..
[inf] _real_confirm(92): certificate verification result: 0x00
[inf] iotx_mc_connect(2035): mqtt connect success!
[dbg] iotx_mc_report_mid(2259): MID Report: started in MQTT
[dbg] iotx_mc_report_mid(2276): MID Report: json data = '{"id":"a10jldzGZFa_stm320001_mid","params":{"_sys_device_mid":"example.demo.module
[dbg] iotx_mc_report_mid(2292): MID Report: topic name = '/sys/a10jldzGZFa/stm320001/thing/status/update'
[dbg] iotx_mc_report_mid(2309): MID Report: finished, IOT_MQTT_Publish() = 0
[inf] iotx_mc_subscribe(1388): mqtt subscribe success,topic = /sys/a10jldzGZFa/stm320001/thing/service/property/set!
[inf] iotx_mc_subscribe(1388): mqtt subscribe success,topic = /sys/a10jldzGZFa/stm320001/thing/event/property/post_reply!
[dbg] iotx_mc_cycle(1269): SUBACK
event_handle|141 :: subscribe success, packet-id=0
[dbg] iotx_mc_cycle(1269): SUBACK
event_handle|141 :: subscribe success, packet-id=0
[inf] iotx_mc_keepalive_sub(2226): send MQTT ping...
[inf] iotx_mc_cycle(1295): receive ping response!
[inf] iotx_mc_keepalive_sub(2226): send MQTT ping...
[inf] iotx_mc_cycle(1295): receive ping response!
```

ali_mqtt_test pub 123

```
msh />ali_mqtt_test pub 133
param:133
msh />[inf] iotx_mc_keepalive_sub(2226): send MQTT ping...
[inf] iotx_mc_cycle(1295): receive ping response!
```

命令不对，
应该是 ali_mqtt_test pub open

```
msh />ali_mqtt_test pub open
param:open
ali_mqtt_test pub|573 ::
publish message:
topic: /sys/a10jldzGZFa/stm320001/thing/event/property/post
payload: {"id": "1","version":"1.0","params
msh />[dbg] iotx_mc_cycle(1260): PUBACK
event_handle|165 :: publish success, packet-id=0
[dbg] iotx_mc_cycle(1277): PUBLISH
[dbg] iotx_mc_handle_recv_PUBLISH(1091): Packet Ident : 00000000
[dbg] iotx_mc_handle_recv_PUBLISH(1092): Topic Length : 58
[dbg] iotx_mc_handle_recv_PUBLISH(1096): Topic Name : /sys/a10jldzGZFa/stm320001/thing/event/property/post_reply
[dbg] iotx_mc_handle_recv_PUBLISH(1099): Payload Len/Room : 137 / 961
[dbg] iotx_mc_handle_recv_PUBLISH(1100): Receive BufLen : 1024
[dbg] iotx_mc_handle_recv_PUBLISH(1111): delivering msg ...
[dbg] iotx_mc_deliver_message(866): topic be matched
_demo_message_arrive|199 :: ----
_demo_message_arrive|200 :: packetId: 0
_demo_message_arrive|204 :: Topic: '/sys/a10jldzGZFa/stm320001/thing/event/property/post_reply' (Length: 58)
_demo_message_arrive|208 :: Payload: '{"code":6320,"data":{},"id":"1","message":"tsl parse: no property exist in product","method":"thing.event.property.p
_demo_message_arrive|209 :: ----
[inf] iotx_mc_keepalive_sub(2226): send MQTT ping...
[inf] iotx_mc_cycle(1295): receive ping response!
```

设备行为分析

上行消息分析

下行消息分析

请输入DeviceName

1小时

2018-08-24 09:40:22

-

2018-08-24 10:40:22

收起

请输入MessageID

全部状态

时间	MessageID	DeviceName	内容(全部)	状态以及原因分析
2018/08/24 10:39:47	1032819732060848128	stm320001	Publish message to topic:/sys/a...	成功
2018/08/24 10:32:46	1032817964656713728	stm320001	Publish message to topic:/sys/a...	成功
2018/08/24 09:56:23	1032808808734351361	stm320001	Publish message to topic:/sys/a...	成功

日志服务 日志以英文展示，详细的中英文对照请参考[文档](#)

设备行为分析

上行消息分析

下行消息分析

请输入DeviceName

1小时

2018-08-24 09:40:22

-

2018-08-24 10:40:22

收起

请输入MessageID

全部状态

时间	MessageID	DeviceName	内容(全部)	状态以及原因分析
2018/08/24 10:39:47	1032819732111179776	stm320001	Publish message to topic:/sys/a...	成功

```
msh />
msh />ali_mqtt_test stop
msh />[inf] iotx_mc_unsubscribe(1423): mqtt unsubscribe success,topic = /sys/a10jldzGZFfa/stm320001/thing/event/property/post_reply!
[inf] iotx_mc_unsubscribe(1423): mqtt unsubscribe success,topic = /sys/a10jldzGZFfa/stm320001/thing/service/property/set!
event_handle[153 :: unsubscribe success, packet-id=0
[dbg] iotx_mc_disconnect(2121): rc = MQTTDisconnect() = 0
[inf] network_ssl_disconnect(514): ssl_disconnect
[inf] iotx_mc_disconnect(2129): mqtt disconnect!
[inf] iotx_mc_release(2175): mqtt release!
[err] LITE_dump_malloc_free_stats(594): WITH_MEM_STATS = 0
mqtt_client[346 :: out of sample!
```

```
L
]#ifndef PKG_USING_ALI_IOTKIT_IS_LINKDEVELOP
/* ALINK TSL Device attribute report */
#define ALINK_PROPERTY_POST_PUB | "/sys/"PRODUCT_KEY"/"DEVICE_NAME"/thing/event/property/post"
#define ALINK_PROPERTY_POST_REPLY_SUB "/sys/"PRODUCT_KEY"/"DEVICE_NAME"/thing/event/property/post_reply"
#define ALINK_PROPERTY_SET_REPLY_SUB "/sys/"PRODUCT_KEY"/"DEVICE_NAME"/thing/event/property/set_reply"
#define ALINK_SERVICE_SET_SUB "/sys/"PRODUCT_KEY"/"DEVICE_NAME"/thing/service/property/set"
#else
#define TOPIC_UPDATE "/PRODUCT_KEY"/"DEVICE_NAME"/update"
#define TOPIC_ERROR "/PRODUCT_KEY"/"DEVICE_NAME"/update/error"
#define TOPIC_GET "/PRODUCT_KEY"/"DEVICE_NAME"/get"
#define TOPIC_DATA "/PRODUCT_KEY"/"DEVICE_NAME"/data"
#endif
```

程序中订阅的是这几个主题，所以不应该选 DEVELOP 平台。试试。

设备信息

Topic列表

设备影子

设备的Topic列表

设备的Topic	设备具有的权限	发布消息数	操作
/a1OjIDzGZFfa/stm320001/data	发布和订阅	0	发布消息
/a1OjIDzGZFfa/stm320001/update	发布	0	发布消息
/a1OjIDzGZFfa/stm320001/update/error	发布	0	发布消息
/a1OjIDzGZFfa/stm320001/get	订阅	0	发布消息

点发布消息。

stm320001 在线

产品: RTT_ARM_DIY 查看

ProductKey: a10jldzGZFfa 复制

DeviceSecret: ***** 显示

设备信息

Topic列表

设备影子

设备的Topic列表

设备的Topic	设备具有的权限	发布消息数	操作
/a10jldzGZFfa/stm320001/data	发布和订阅	1	发布消息
/a10jldzGZFfa/stm320001/update	发布	0	发布消息
/a10jldzGZFfa/stm320001/update/error	发布	0	发布消息
/a10jldzGZFfa/stm320001/get	订阅	0	发布消息



注意：如果该Topic正在被应用使用，请谨慎操作，以防应用出现异常。

Topic:

/a10jldzGZFa/stm320001/data

* 消息内容:

dddddddd

8/1000

* Qos:

☒ 0 ☐ 1

确认

取消

```
[inf] iotx_mc_keepalive_sub(2226): send MQTT ping...
[inf] iotx_mc_cycle(1295): receive ping response!
[dbg] iotx_mc_cycle(1277): PUBLISH
[dbg] iotx_mc_handle_rcv_PUBLISH(1091):      Packet Ident : 00000000
[dbg] iotx_mc_handle_rcv_PUBLISH(1092):      Topic Length : 27
[dbg] iotx_mc_handle_rcv_PUBLISH(1096):      Topic Name : /a10jldzGZFa/stm320001/data
[dbg] iotx_mc_handle_rcv_PUBLISH(1099):      Payload Len/Room : 8 / 993
[dbg] iotx_mc_handle_rcv_PUBLISH(1100):      Receive Buflen : 1024
[dbg] iotx_mc_handle_rcv_PUBLISH(1111): delivering msg ...
[dbg] iotx_mc_deliver_message(866): topic be matched
_demo_message_arrive|199 :: ----
_demo_message_arrive|200 :: packetId: 0
_demo_message_arrive|204 :: Topic: '/a10jldzGZFa/stm320001/data' (Length: 27)
_demo_message_arrive|208 :: Payload: 'dddddddd' (Length: 8)
_demo_message_arrive|209 :: ----
```

行吧。先做到这里，可以正常订阅 发布了。

14.