



北京联盛德微电子有限责任公司

W60X_SDK DEMO 运行指南

V0.5

北京联盛德微电子有限责任公司 (winner micro)

地址：北京市海淀区阜成路 67 号银都大厦 18 层

电话：+86-10-62161900

公司网址：www.winnermicro.com

文档修改记录

版本	修订时间	修订记录	作者	审核
V0.1	20180830	初稿	Chenzx	
V0.2	20180921	重新整理	Ligh	
V0.3	20181012	更新截图	Ligh	
V0.4	20181101	增加 slave spi demo, 修改两处错误	Zhangwl	
V0.5	20181213	该 DEMO 支持 W600/W601, 文件 更名为 W60X	Cuiych	

目录

文档修改记录.....	1
1 引言.....	4
1.1 编写目的.....	4
1.2 预期读者.....	4
1.3 术语定义.....	4
2 DEMO 概要.....	4
3 DEMO 功能描述.....	4
3.1 DEMO_CONNECT_NET 操作步骤	4
3.1.1 t-connect 加网.....	4
3.1.2 t-oneshot (app 配网, 需要下载安卓 OneShotActivity).....	4
3.1.3 t-oneshot (airkiss 配网).....	5
3.1.4 t-webcfg 配网.....	5
3.2 DEMO_STD_SOCKET_CLIENT 操作步骤	5
3.3 DEMO_STD_SOCKET_SERVER 操作步骤	5
3.4 DEMO_UDP 操作步骤	6
3.4.1 UDP 广播.....	6
3.4.2 UDP 单播.....	6
3.4.3 UDP 组播.....	7
3.5 DEMO_APSTA 操作步骤	7
3.6 DEMO_UARTx 操作步骤	7
3.7 DEMO_SOFT_AP 操作步骤	8
3.8 DEMO_WPS 操作步骤	8
3.8.1 t-wps-start-pbc.....	8
3.8.2 t-wps-start-pin.....	8
3.9 DEMO_GPIO 操作步骤	9
3.9.1 t-gpio (使用 PB13、PB14 演示).....	9
3.9.2 t-gpioirq (使用 PA1 演示).....	9
3.10 DEMO_FLASH 操作步骤	9
3.11 DEMO_ENCRYPT 操作步骤	9
3.12 DEMO_RSA 操作步骤	10
3.13 DEMO_RTC 操作步骤	11

3.14	DEMO_TIMER 操作步骤.....	11
3.15	DEMO_PWM 操作步骤.....	11
3.16	DEMO_PMU 操作步骤.....	11
3.16.1	t-pmuT0.....	11
3.16.2	t-pmuT1.....	12
3.17	DEMO_NTP 操作步骤.....	12
3.17.1	t-ntp.....	12
3.17.2	t-setntp.....	12
3.18	DEMO_HTTP 操作步骤.....	12
3.18.1	t-httpget.....	14
3.18.2	t-httpput.....	14
3.18.3	t-httppost.....	15
3.18.4	t-httpwup.....	15
3.19	DEMO_I2C 操作步骤.....	16
3.20	DEMO_SSL_SERVER 操作步骤.....	16
3.21	DEMO_WEBSOCKETS 操作步骤.....	17
3.21.1	websocket 不加密.....	17
3.21.2	websocket 加密.....	18
3.22	DEMO_I2S 操作步骤.....	18
3.23	DEMO_MASTER_SPI 操作步骤.....	19
3.24	DEMO_SLAVE_SPI 操作步骤.....	20

1 引言

1.1 编写目的

为基于 W60X 芯片 WMSDK 进行二次开发的软件开发工程师提供相关功能的代码示例。

1.2 预期读者

FAE, 客户方软件开发工程师。

1.3 术语定义

2 DEMO 概要

该文档中用到的所有 DEMO 相关的宏定义都在 `wm_demo.h` 中。运行 DEMO 时必须打开该 DEMO 对应的宏定义, 建议关闭不相关宏定义。DEMO 演示需要在控制台下进行, 打开 `DEMO_CONSOLE` 编译选项, 即打开了控制台。

3 DEMO 功能描述

3.1 DEMO_CONNECT_NET 操作步骤

注: 此 DEMO 下有四个演示 DEMO

3.1.1 t-connect 加网

1. 打开宏定义 `DEMO_CONNECT_NET`;
2. 编译, 升级成功后, 在 `uart0` 打印的控制台信息中能看到对应命令;
3. 通过 `uart0` 发送 `t-connect("TEST_N40_6","1234567890")`所有命令需要带回车换行, 命令中使用英文符号;
4. 加网成功后 `uart0` 会打印模块 ip。

3.1.2 t-oneshot (app 配网, 需要下载安卓 OneShotActivity)

1. 打开宏定义 `DEMO_CONNECT_NET`;
2. 编译, 升级成功后, 在 `uart0` 打印的控制台信息中能看到对应命令;
3. 通过 `uart0` 发送 `t-oneshot`;
4. 手机加入目标网络, 安装 `OneShotActivity (SDK ver2.0.0)`, 在 app 界面输入正确 `ssid` 和 `password`, 点 `Start Configuration`;
5. 模块加网成功后 `uart0` 会打印 ip。

3.1.3 t-oneshot (airkiss 配网)

1. 打开宏定义 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-oneshot;
4. 手机加入目标网络 (需要外网), 打开微信, 关注公众号【联盛德微电子】, 进入公众号后点击产品应用下的 AirKiss 配网, 进入配置设备上网页面, 设置正确 Wi-Fi 密码, 点击连接按钮;
5. 模块加网成功后 uart0 会打印 ip。

3.1.4 t-webcfg 配网

1. 打开宏定义 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-webcfg;
4. 手机加入"softap_XXXX" (模块 mac 地址的后 4 位), 用浏览器访问 192.168.1.1, 在页面 List 中选择目标网络 (如果找不到目标网络, 尝试刷新页面), 在 pwd 输入正确密码, 点击 save 按钮;
5. 模块加网成功后 uart0 会打印模块 ip, 同网络设备可以 ping 通模块 ip。

3.2 DEMO_STD_SOCKET_CLIENT 操作步骤

1. 打开宏定义 DEMO_STD_SOCKET_CLIENT 和 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网;
4. 在与模块同网络的 PC(ip 为 192.168.1.100)上打开调试助手 tcp server 端口号 1000;
5. 通过 uart0 发送 t-sockc(1000,192.168.1.100)让模块创建 tcp client 连接对端 server, 连接成功后 uart0 会打印 socket num;
6. Server 发数据, 模块收到数据后 uart0 会打印收到的数据长度, 每次累加;
7. 通过 uart0 发送 t-skcsnd(0,1)设置使用 uart1 透传;
8. 串口工具设置波特率 115200、校验位 NONE、数据位 8、停止位 1 打开 uart1, 通过 uart1 与 server 双向透传;

注: 通过 uart0 发送 demohelp 模块 uart0 会返回控制台信息。

3.3 DEMO_STD_SOCKET_SERVER 操作步骤

1. 打开宏定义 DEMO_STD_SOCKET_SERVER 和 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-connect("HUAWEI-6SEWE5","123456789")或 t-oneshot 让模块加网;

4. 通过 `uart0` 发送 `t-socks(2000)`让模块创建 `tcp server`, `uart0` 会打印监听的端口;
5. 在与模块同网络的 PC 上打开调试助手, 创建 `tcp client` (设置模块的 `ip` 和端口号) 连接模块 `server`, 连接成功后 `uart0` 会打印 `client` 信息 (模块 `server` 最多连接 7 个 `client`);
6. `client` 发数据, 模块收到数据后 `uart0` 会打印收到对应连接的数据长度, 每次累加;
7. 通过 `uart0` 发送 `t-skssnd(1,16,0)`使用 1 号连接发送长度 16 的固定数据, `client` 能收到数据;
8. 通过 `uart0` 发送 `t-skssnd(1,0,1)`设置 1 号连接在 `uart1` 透传;
9. 串口工具设置波特率 115200、校验位 NONE、数据位 8、停止位 1 打开 `uart1`, 通过 `uart1` 与 `client` 双向透传。

3.4 DEMO_UDP 操作步骤

注: 此 DEMO 下有四个演示 DEMO, 需要使用抓包网卡

3.4.1 UDP 广播

1. 打开宏定义 `DEMO_UDP` 和 `DEMO_CONNECT_NET`;
2. 编译, 升级成功后, 在 `uart0` 打印的控制台信息中能看到对应命令;
3. 通过 `uart0` 发送 `t-connect("TEST_N40_6","1234567890")`或 `t-oneshot` 让模块加网;
4. 通过 `uart0` 发送 `t-udp(0,1000,0)``uart0` 打印

```
udp demo,cast:0, port:1000
localip : 192.168.1.104
local port :3000
```
5. 在与模块同网络的 PC 上打开调试助手 `udp` 端口 1000;
6. 通过 `uart0` 发送 `t-sndudp(10)`, 抓包网卡可以抓到模块到路由器的 `Destination` 为 `Ethernet Broadcast` 的包, 同时调试助手收到了 10 个数据;
7. 调试助手发数据, 模块收到数据后 `uart0` 会打印地址和数据长度。

3.4.2 UDP 单播

1. 打开宏定义 `DEMO_UDP` 和 `DEMO_CONNECT_NET`;
2. 编译, 升级成功后, 在 `uart0` 打印的控制台信息中能看到对应命令;
3. 通过 `uart0` 发送 `t-connect("TEST_N40_6","1234567890")`或 `t-oneshot` 让模块加网;
4. 通过 `uart0` 发送 `t-udp(1,1001,192.168.1.100)``uart0` 会打印

```
udp demo,cast:1, port:1001
localip : 192.168.1.104
local port :3000
```
5. 在与模块同网络的 PC (`ip` 为 192.168.1.100) 上打开调试助手连接 `udp` 端口 1001;

6. 通过 `uart0` 发送 `t-sndudp(10)` 抓包网卡可以抓到模块到路由器的 Destination 为 PC 网卡的包，同时调试助手收到了 10 个数据；
7. 调试助手发数据，模块收到数据后 `uart0` 会打印地址和数据长度。

3.4.3 UDP 组播

1. 打开宏定义 `DEMO_UDP` 和 `DEMO_CONNECT_NET`；
2. 编译，升级成功后，在 `uart0` 打印的控制台信息中能看到对应命令；
3. 通过 `uart0` 发送 `t-connect("TEST_N40_6","1234567890")` 或 `t-oneshot` 让模块加网；
4. 通过 `uart0` 发送 `t-udp(2,5100,224.1.2.1)` `uart0` 会打印：

```
udp demo,cast:2, port:5100
localip : 192.168.1.104
local port :3000
setmuticast
```
5. 在与模块同网络的 PC 上打开组播工具，在接收测试中添加地址(组播地址为 224.1.2.1，端口为 5100)，选择地址，点击接收按钮；
6. 通过 `uart0` 发送 `t-sndudp(1024)`，组播工具中显示未丢包；
7. 在 PC 打开调试助手，设置目标组播地址 224.1.2.1 目标端口 3000，发送数据，模块收到数据后 `uart0` 打印地址和数据长度。

3.5 DEMO_APSTA 操作步骤

1. 打开宏定义 `DEMO_APSTA`；
2. 编译，升级成功后，在 `uart0` 打印的控制台信息中能看到对应命令；
3. 在与模块同网络的 PC 上打开调试助手 UDP 的 65530 端口，设置十六进制显示；
4. 通过 `uart0` 发送 `t-apsta("TEST_N40_6","1234567890","softapssid")`；
5. `uart0` 会打印 `softap` 的 ip 与模块 `sta` 的 ip，此时调试助手会收到 `sta` 重复发的 mac 地址；尽快用其他设备加入软 ap，在 `udp` 的 65530 端口可以收到软 ap 重复发的 mac 地址；
6. 创建的 `softap` 是不加密的，手机加入 `softap` 后，`uart0` 会打印设备上线，手机可以 ping 通路由器下的设备。

3.6 DEMO_UARTx 操作步骤

1. 打开宏定义 `DEMO_UARTx`；
2. 编译，升级成功后，在 `uart0` 打印的控制台信息中能看到对应命令；
3. 通过 `uart0` 发送 `t-uart=(9600,0,0)` 修改 `uart1` 的参数；
4. 串口工具设置波特率 9600、校验位 NONE、数据位 8、停止位 1，打开 `uart1` 发数

据，模块会把收到的数据从 uart1 打印出来。

3.7 DEMO_SOFT_AP 操作步骤

1. 打开宏定义 DEMO_SOFT_AP;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-softap("softap1s","1234567890",6,4,1);
4. 手机可以扫描到"softap1s"网络，加入 softap 后，uart0 会打印手机 mac。

3.8 DEMO_WPS 操作步骤

注：此 DEMO 下有两个演示 DEMO，需要路由器支持 wps



3.8.1 t-wps-start-pbc

1. 打开宏定义 DEMO_WPS;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-wps-start-pbc，并在路由器上按 wps 按钮，稍候 uart0 打印
[CMD]t-wps-start-pbcStart WPS pbc mode ...
WiFi JOIN SUCCESS
NET UP OK,Local IP:192.168.1.101

3.8.2 t-wps-start-pin

1. 打开宏定义 DEMO_WPS;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-wps-get-pin，uart0 打印 pin 码并自动给模块设置;
4. 在路由器中输入 pin 码，启动连接;
5. 通过 uart0 发送 t-wps-start-pin，稍候 uart0 打印
[CMD]t-wps-start-pinStart WPS pin mode ...
WiFi JOIN SUCCESS
NET UP OK,Local IP:192.168.1.101

3.9 DEMO_GPIO 操作步骤

注：此 DEMO 下有两个演示 DEMO

3.9.1 t-gpio（使用 PB13、PB14 演示）

1. 打开宏定义 DEMO_GPIO;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-gpio，uart0 会打印测试结果

```
gpioB[13] default value==[0]
gpioB[13] floating high value==[1]
gpioB[13] floating low value==[0]
gpioB[13] pulllow high value==[1]
gpioB[13] pulllow low value==[0]
gpioB[14] default value==[0]
gpioB[14]floating high value==[1]
gpioB[14]floating low value==[0]
gpioB[14] pulllow high value==[1]
gpioB[14] pulllow low value==[0]
```

3.9.2 t-gpioirq（使用 PA1 演示）

1. 打开宏定义 DEMO_GPIO;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-gpioirq，把 PA1 拉低 uart0 打印

```
int flag =1
after int io =0
```

4. 把 PA1 拉高，uart0 打印

```
int flag =1
after int io =1
```

3.10 DEMO_FLASH 操作步骤

1. 打开宏定义 DEMO_FLASH;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-flash，uart0 会打印 success。

3.11 DEMO_ENCRYPT 操作步骤

1. 打开宏定义 DEMO_ENCRYPT;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；

3. 通过 uart0 发送 t-crypt, uart0 会打印

[CMD]t-cryptRNG out:

C4 5F 5E FE A7 E 99 CC 1E 7

RNG out:

14 8F 2E 44 B7 3E 29 29 EE 7C 58 48 E9 C6 6B CC 18 97 52 37

rc4 test success

aes ecb test success

aes cbc test success

aes ctr test success

des ecb test success

des cbc test success

3des ecb test success

3des cbc test success

CRYPTO_CRC_TYPE_8 normal value:0x000000B7

CRYPTO_CRC_TYPE_8 INPUT_REFLECT value:0x000000E2

CRYPTO_CRC_TYPE_8 OUTPUT_REFLECT value:0x000000ED

CRYPTO_CRC_TYPE_8 INPUT_REFLECT | OUTPUT_REFLECT value:0x00000047

CRYPTO_CRC_TYPE_16_MODBUS normal value:0x0000B69B

CRYPTO_CRC_TYPE_16_MODBUS INPUT_REFLECT value:0x0000642A

CRYPTO_CRC_TYPE_16_MODBUS OUTPUT_REFLECT value:0x0000D96D

CRYPTO_CRC_TYPE_16_MODBUS INPUT_REFLECT | OUTPUT_REFLECT value:0x00005426

CRYPTO_CRC_TYPE_16_CCITT normal value:0x00008CC2

CRYPTO_CRC_TYPE_16_CCITT INPUT_REFLECT value:0x00005CA9

CRYPTO_CRC_TYPE_16_CCITT OUTPUT_REFLECT value:0x00004331

CRYPTO_CRC_TYPE_16_CCITT INPUT_REFLECT | OUTPUT_REFLECT value:0x0000953A

CRYPTO_CRC_TYPE_32 normal value:0x3C514E40

CRYPTO_CRC_TYPE_32 INPUT_REFLECT value:0x715268A8

CRYPTO_CRC_TYPE_32 OUTPUT_REFLECT value:0x02728A3C

CRYPTO_CRC_TYPE_32 INPUT_REFLECT | OUTPUT_REFLECT value:0x15164A8E

md5 test success

sha1 test success

3.12 DEMO_RSA 操作步骤

1. 打开宏定义 DEMO_RSA;

2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-rsa，uart0 会打印

```
[CMD]t-rsarsa test start
rsa128 test sucess
rsa256 test sucess
rsa512 test sucess
rsa1024 test sucess
rsa2048 test sucess
rsa test end
```

3.13 DEMO_RTC 操作步骤

1. 打开宏定义 DEMO_RTC；
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-rtc 开启 rtc clock, 20 秒时 uart0 会打印 rtc clock 表示进入 rtc 中断。

3.14 DEMO_TIMER 操作步骤

1. 打开宏定义 DEMO_TIMER；
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-timer 开启 timer, uart0 每 2 秒打印 timer irq 表示进入 timer 中断。

3.15 DEMO_PWM 操作步骤

1. 打开宏定义 DEMO_PWM；
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-pwm=(1,10000,99,4,0), 示波器量 PB17 可以测到 10KHz, 占空比约为 39%(99/255)的波形。
4. 括号中的第一个参数为通道序号, 序号 0-4 分别对应 demo 中的 PB18-PB14 共五路。

3.16 DEMO_PMU 操作步骤

注：此 DEMO 下有两个演示 DEMO

3.16.1 t-pmuT0

1. 打开宏定义 DEMO_PMU；
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-pmuT0 模块启动 timer0 进入 standby, 10 秒左右 uart0 打印模块复位，表示 timer0 中断唤醒；

3.16.2 t-pmuT1

1. 打开宏定义 DEMO_PMU;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-pmuT1 模块启动 timer1 进入 standby, 5 秒左右 uart0 打印模块复位, 表示 timer1 中断唤醒。

3.17 DEMO_NTP 操作步骤

注: 此 DEMO 下有两个演示 DEMO

3.17.1 t-ntp

1. 打开宏定义 DEMO_NTP 和 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网(有外网);
4. 通过 uart0 发送 t-ntp, uart0 会打印当前时间。

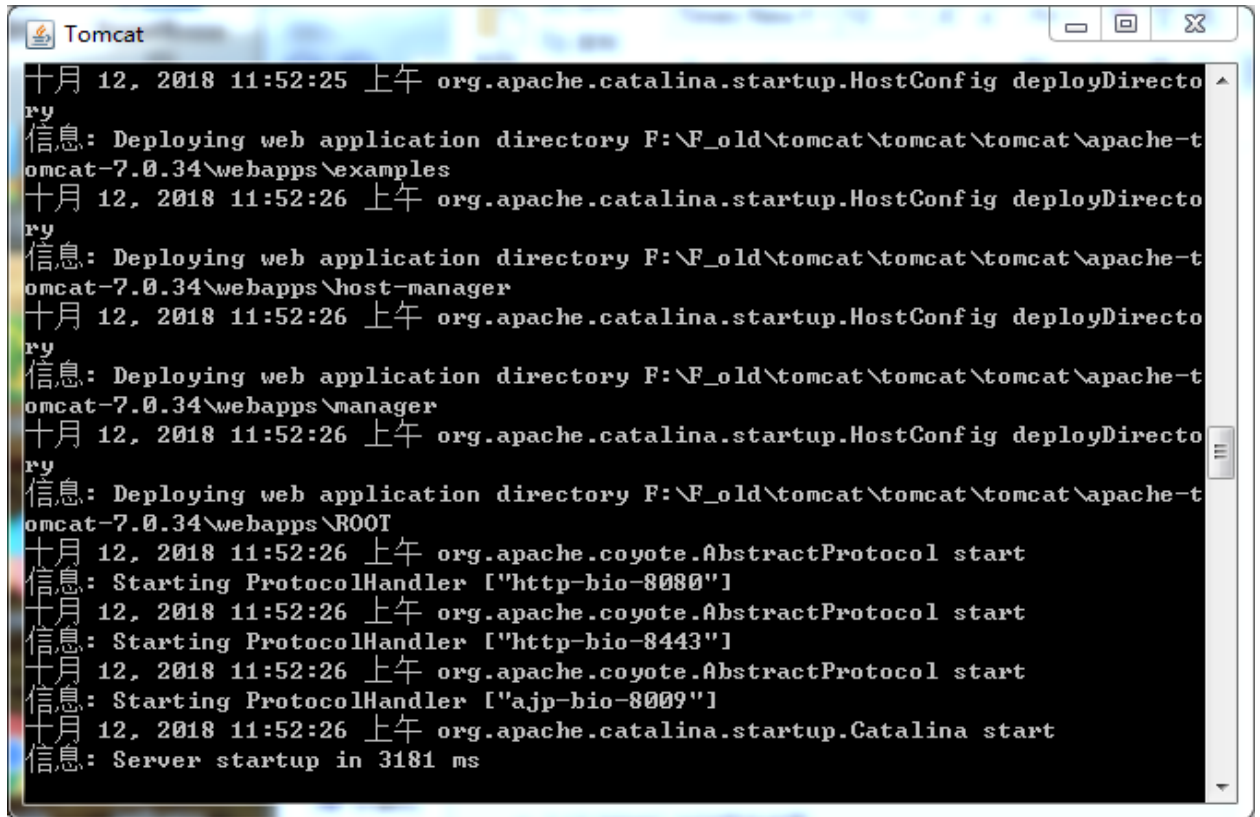
3.17.2 t-setntps

1. 打开宏定义 DEMO_NTP 和 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-setntps("120.25.108.11", "ntp.sjtu.edu.cn", "us.pool.ntp.org")手动设置 ntp 服务器;
4. 复位模块后, 通过 uart0 发送 t-queryntps 返回
[CMD]t-queryntps"120.25.108.11","ntp.sjtu.edu.cn","us.pool.ntp.org"
5. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网(有外网);
6. 通过 uart0 发送 t-ntp, uart0 会打印当前时间。

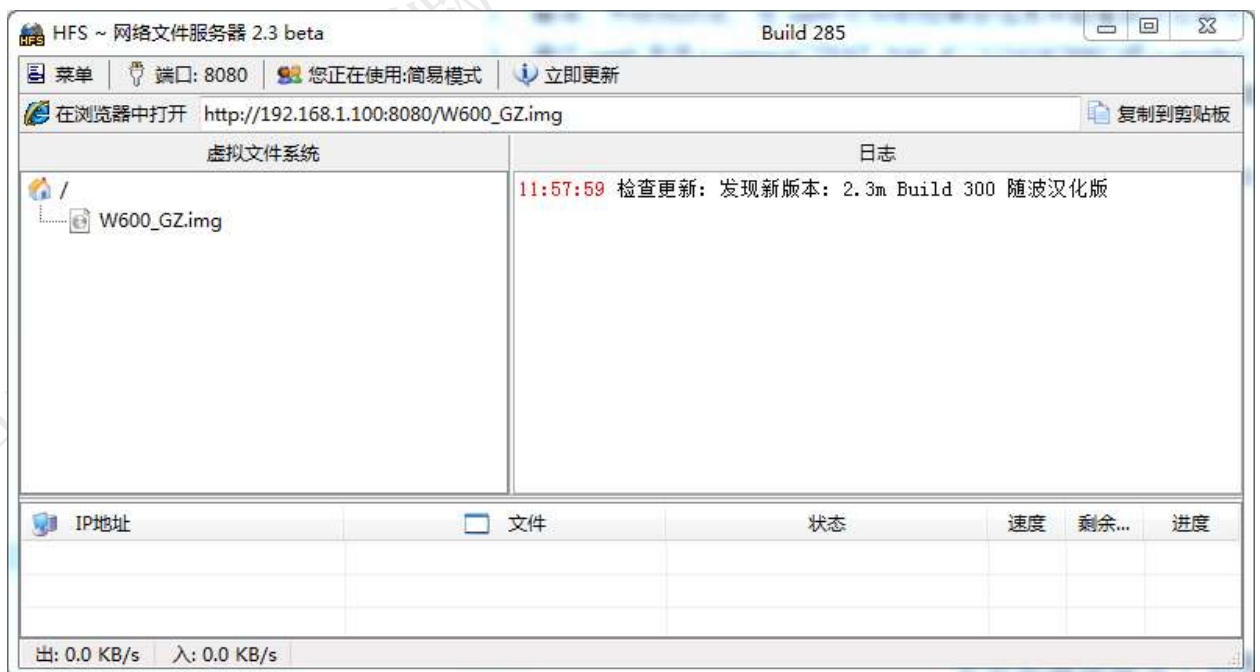
3.18 DEMO_HTTP 操作步骤

注: 此 DEMO 下有四个演示 DEMO, 需要下载 tomcat 服务器(需要放置所需脚本文件)和 hfs 服务器。

下图分别为 tomcat 服务器启动后的页面：



以及 http 服务器添加固件就绪后的页面：



其中 hfs 服务器及 tomcat 服务器可以从网上下载，hfs 下载后直接可用，tomcat(已测试过 7.0.34 及 8.5.23 版本)服务器下载下来后需要在里面修改添加一些脚本文件。具体为将 tomcat 根目录下的 webapps 文件夹下的 TestWeb 文件夹替换为官方提供的 TestWeb 文件夹(已在里面添加了测试 httpget httpput httppost 所需要的相应脚本文件)。

3.18.1 t-httpget

1. 打开宏定义 DEMO_HTTP 和 DEMO_CONNECT_NET;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网;
4. 在与模块同网络的 PC (ip 为 192.168.1.100) 上打开 tomcat 服务器并放置文件;
5. 通过 uart0 发送 t-httpget, uart0 返回

```
[CMD]t-httpgetLocation: http://192.168.1.100:8080/TestWeb/
HTTP Client v1.0
Start to receive data from remote server...
<html>
<body>
<h2>Hello World!</h2>
<form method="POST" action="/TestWeb/login.do">
  userd: <input id="user" type="text" name="user"/>
  <input type="submit" value="Submit" />
  <div> </div>
</form>
</body>
</html>
```

HTTP Client terminated 1000 (got 213 b)

3.18.2 t-httpput

1. 打开宏定义 DEMO_HTTP 和 DEMO_CONNECT_NET;
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网;
4. 在与模块同网络的 PC (ip 为 192.168.1.100) 上打开 tomcat 服务器并放置文件;
5. 通过 uart0 发送 t-httpput=(user=winnermicroput), uart0 返回

```
Location: http://192.168.1.100:8080/TestWeb/login_put.do
HTTP Client v1.0
Start to receive data from remote server...
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=GBK">
<title>Insert title here</title>
</head>
<body>
:winnermicroput
</body>
</html>
```

HTTP Client terminated 1000 (got 277 b)

3.18.3 t-httpst

1. 打开宏定义 DEMO_HTTP 和 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网;
4. 在与模块同网络的 PC (ip 为 192.168.1.100) 上打开 tomcat 服务器并放置文件;
5. 通过 uart0 发送 t-httpst=(user=winnermicropost), uart0 返回

Location: http://192.168.1.100:8080/TestWeb/login.do

HTTP Client v1.0

Start to receive data from remote server...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=GBK">
<title>Insert title here</title>
</head>
<body>
:winnermicropost
</body>
</html>
```

HTTP Client terminated 1000 (got 278 b)

3.18.4 t-httpfwup

1. 打开宏定义 DEMO_HTTP 和 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;

3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网;
4. 在与模块同网络的 PC (ip 为 192.168.1.100) 上打开 hfs 服务器, 端口 8080, 并放置名称为 WM_W600_SEC.img 的固件;
5. 通过 uart0 发送 t-httpfwup=(http://192.168.1.100:8080/WM_W600_SEC.img), uart0 打印升级进度, 模块升级成功后复位。升级压缩的 img。

3.19 DEMO_I2C 操作步骤

注: 此 DEMO 需要 AT24CXX 芯片



1. 打开宏定义 DEMO_I2C;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 模块 PIN 连接 AT24CXX 芯片:
4. PB11 接 SCL, PB12 接 SDA, GND 接 GND, VCC 接 3.3v
5. 通过 uart0 发送 t-i2c, uart0 返回

```
[CMD]t-i2c
```

```
AT24CXX check success
```

```
read data is:AT24CXX I2C TEST OK
```

3.20 DEMO_SSL_SERVER 操作步骤

注: 此 DEMO 需要打开 TLS_CONFIG_SERVER_SIDE_SSL, 演示其他 DEMO 时关闭, 需要下 openssl 或其他可以连接 ssl server 的工具。具体 demo 测试步骤如下:

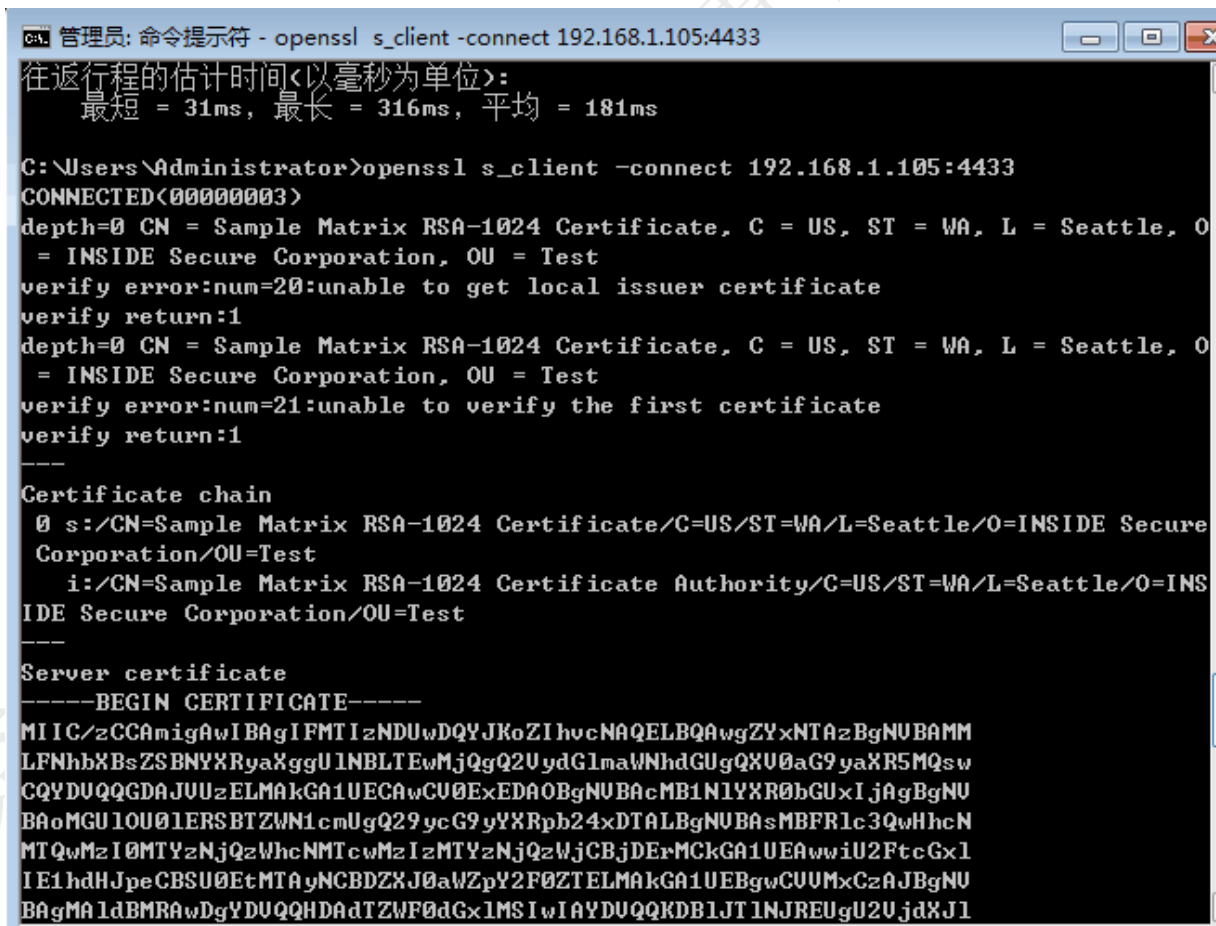
1. 打开宏定义 DEMO_SSL_SERVER 和 DEMO_CONNECT_NET;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网 (ip 为 192.168.1.104);
4. 通过 uart0 发送 t-ssl-server, uart0 返回

```
[CMD]t-ssl-server
ssl server task
Listening on port 4433
```

5. 在与模块同网络的 PC 上打开 openssl s_client -connect 192.168.1.104:4433
6. 此时模块的 uart0 打印

```
accept fd 1
tls_mem_alloc cp 2001ef88
tls_ssl_server_handshake rc 0
cp->time.tv_sec 0
```

下图为使用 openssl(需要用户自己安装)工具连接 ssl server 成功后的命令行页面信息。



```
ca. 管理员: 命令提示符 - openssl s_client -connect 192.168.1.105:4433
往返行程的估计时间<以毫秒为单位>:
  最短 = 31ms, 最长 = 316ms, 平均 = 181ms

C:\Users\Administrator>openssl s_client -connect 192.168.1.105:4433
CONNECTED(00000003)
depth=0 CN = Sample Matrix RSA-1024 Certificate, C = US, ST = WA, L = Seattle, O
= INSIDE Secure Corporation, OU = Test
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 CN = Sample Matrix RSA-1024 Certificate, C = US, ST = WA, L = Seattle, O
= INSIDE Secure Corporation, OU = Test
verify error:num=21:unable to verify the first certificate
verify return:1
---
Certificate chain
 0 s:/CN=Sample Matrix RSA-1024 Certificate/C=US/ST=WA/L=Seattle/O=INSIDE Secure
Corporation/OU=Test
  i:/CN=Sample Matrix RSA-1024 Certificate Authority/C=US/ST=WA/L=Seattle/O=INS
IDE Secure Corporation/OU=Test
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIC/zCCAmigAwIBAgIFMTIzNDUwDQYJKoZIhvcNAQELBQAwgZYxNTAzBgNVBAMM
LFNhbXBsZSBSBnYXRyaXggU1NBLTEwMjQyUydG1maWNhdGUgQXU0aG9yaXR5MQsw
CQYDUQGDAAJUuzELMAkGA1UECAwCU0ExEDAOBgNVBAcMB1N1YXR0bGUxIjAgBgNV
BAoMGU1OU01ERSBTZWN1cmUgQ29ycG9yYXRpb24xDTALBgNVBAcMBFRlc3QwHhcN
MTQwMzI0MTYzNjQzWjcNMTcwMzIzMTYzNjQzWjCBjDERMCAkGA1UEAwwiU2FtcGxl
IE1hdHJpeCBSU0EtMTA5NCBDZXJ0aWZpY2F0ZTElMAkGA1UEBwwCUUMxOzA1BgNV
BAgMA1dBMRAdgYDUQHQDAATZWFOdGx1MSIwIAAYDUQKDB1JT1NjREUgU2UjdXJl
```

3.21 DEMO_WEBSOCKETS 操作步骤

注：此 DEMO 下有两个演示 DEMO，需要下载 WEBSOCKET_SERVER 测试服务器

3.21.1 websocket 不加密

1. 打开宏定义 DEMO_WEBSOCKETS 和 DEMO_CONNECT_NET，关闭

LWS_USE_SSL;

2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网；
4. 如果使用 WEBSOCKET_SERVER 测试服务器，在与模块同网络的 PC（ip 为 192.168.1.100）上命令行运行 websocketd --port=8080 echo_client.bat；
5. 通过 uart0 发送 t-websockets，uart0 返回

```
[CMD]t-websocketsCLIENT_ESTABLISHED
```

```
send {"msg_type":"keepalive"} 2
```

```
recv:websocket server send
```

```
recv:{"msg_type":"keepalive"} 2
```

3.21.2 websocket 加密

1. 打开宏定义 DEMO_WEBSOCKETS、DEMO_CONNECT_NET、LWS_USE_SSL，如果使用测试服务器，请按 wm_websockets_demo.c 中 Notice 步骤修改代码（正规服务器无需 Notice 步骤 3）；
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 通过 uart0 发送 t-connect("TEST_N40_6","1234567890")或 t-oneshot 让模块加网；
4. 如果使用 WEBSOCKET_SERVER 测试服务器，在与模块同网络的 PC（ip 为 192.168.1.100）上命令行运行 websocketd --port=8080 --ssl --sslcert="certificate.pem" --sslkey="key.pem" echo_client.bat；
5. 通过 uart0 发送 t-websockets，uart0 返回

```
[CMD]t-websocketsCLIENT_ESTABLISHED
```

```
send {"msg_type":"keepalive"} 1
```

```
recv:websocket server send
```

```
recv:{"msg_type":"keepalive"} 1
```

3.22 DEMO_I2S 操作步骤

注：此 DEMO 需要两个模块演示，一个作 Master，一个作 Slave

1. 打开宏定义 DEMO_I2S；
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 模块 Master pin 接 Slave 对应 pin:

PB8_M_SCL 接 PB15_S_SCL、PB9_M_SDA 接 PB14_S_SDA、PB10_M_RL 接 PB16_S_RL、GND 接 GND

4. Slave 模块通过 uart0 发送 t-i2s=(0,2,44100,16,0,0)使用中断方式;
5. Master 模块通过 uart0 发送 t-i2s=(0,1,44100,16,0,0), uart0 打印
send 1024
6. Slave 模块 uart0 打印接收到的 1024 个数据
7. Slave 模块通过 uart0 发送 t-i2s=(0,2,44100,16,0,1)使用 DMA 方式;
8. Master 模块通过 uart0 发送 t-i2s=(0,1,44100,16,0,1), uart0 打印
send 1024
9. Slave 模块 uart0 打印接收到的 1024 个数据

3.23 DEMO_MASTER_SPI 操作步骤

注: 此 DEMO 需要下载对端代码

1. 打开宏定义 DEMO_MASTER_SPI;
2. 编译, 升级成功后, 在 uart0 打印的控制台信息中能看到对应命令;
3. 用 keil 打开

STM32_SOC_TEST_SLAVE_SPI\Project\STM32F10x_StdPeriph_Template\MDK-ARM\Project 编译后通过 jlink 给 stm32 升级;

注: STM32 开发板型号: **STM32_Mini_V2.0**

STM32 开发板软件版本: STM32F10x_StdPeriph_Template V3.5.0

STM32 测试代码:

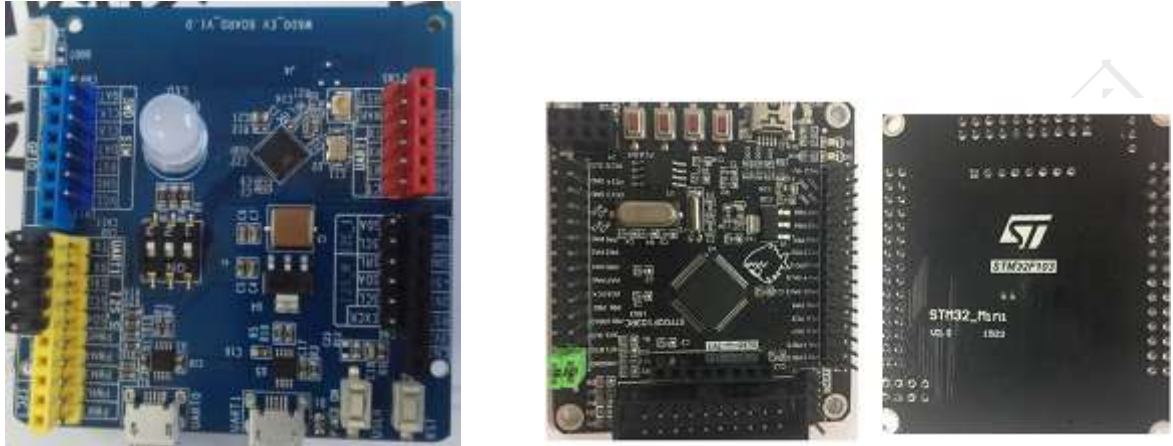


4. 模块 PIN 连接对端 stm32(PA9tx, PA10rx 作为打印口):
PB15 接 PB12(cs), PB16 接 PB13(ck), PB17 接 PB14(so), PB18 接 PB15(si), GND 接 GND;
5. 通过 uart0 发送 t-mspi-s(1000000,0)发送 1500 数据, stm32 的 uart0 打印
down data len: 1500;
6. 通过 uart0 发送 t-mspi-r, 模块 uart0 打印
[CMD]t-mspi-rSPI Master receive 1500 byte, modeA, little endian
rcv data len: 1500。

3.24 DEMO_SLAVE_SPI 操作步骤

注：此 DEMO 需要下载对端代码，STM32 开发板型号：STM32_Mini_V2.0；使用 W600_EVM_V10 开发板或在所用 pin 上串几十欧电阻。

下图中左图为 W600 arduino 开发板，右图为 stm32 开发板；



1. 打开宏定义 DEMO_SLAVE_SPI
2. 编译，升级成功后，在 uart0 打印的控制台信息中能看到对应命令；
3. 用 keil 打开 stm32_ucos_ri\uCOSDemo 编译后通过 jlink 给 stm32 升级；
4. 模块 PIN 连接对端 stm32(PA9tx, PA10rx 作为打印口):
PB15 接 PA4(cs), PB16 接 PA5(ck), PB18 接 PA6(mi), B17 接 PA7(mo), PB14 接 PA0(cts), GND 接 GND
5. 通过 uart0 发送 t-sspi=(0)
6. 复位 stm32, 模块 uart0 打印:
HspiRxCmdCb
rx[5] :5a 00 05 01 60
RX ok 100
RX ok 200
RX ok 300
7. Stm32 打印:
###kevin debug ...
tx start cmd
kevin debug TX_BUFFER_AVAIL = 3, cmdlen=8
RX ok 100
RX ok 200
RX ok 300