SCM1612
Wi-Fi 6 和 BLE 5 低功耗 SoC

MQTT 开发指南

文档版本 0.1
发布日期 2024-3-11

联系方式
速通半导体科技有限公司 (www.senscomm.com)
江苏省苏州市工业园区苏州大道西 2 号国际大厦 303 室
销售或技术支持，请发送电子邮件至
support@senscomm.com

免责声明和注意事项

本文档仅按"现状"提供。速通半导体有限公司保留在无需另行通知的情况下对其或本文档中包含的任何规格进行更正、改进和其他变更的权利。

与使用本文档中的信息有关的一切责任，包括侵犯任何专有权利的责任，均不予承认。此处不授予任何明示或暗示、通过禁止或其他方式对任何知识产权的许可。

本文档中的所有第三方信息均按"现状"提供，不对其真实性和准确性提供任何保证。

本文档中提及的所有商标、商号和注册商标均为其各自所有者的财产，特此确认。

版本历史

版本	日期	描述
1.0	2024-10-08	更新了构建过程和命令行界面命令
0.1	2024-3-11	初稿

目录

1 开发指南

本文档旨在帮助实现需要 MQTT 客户端功能的应用程序。

1.1 概述

SCM1612 SDK 使用 coreMQTT-Agent 和底层 coreMQTT：

- API 位于: `lib/mqtt/coreMQTT-Agent`, `lib/mqtt/coreMQTT`
- 示例位于: `api/examples/protocols/mqtt`

1.2 示例操作指南

按照以下步骤在 SCM1612 平台上运行 MQTT 示例。

1.2.1 配置构建选项

1. 选择 MQTT 示例作为主应用程序：
   ```
   $ make scm1612s_defconfig
   $ make menuconfig
   ```
2. 进入以下路径：
   `Applications -> Protocols Demo`
3. 选择：
   `Protocols Demo -> MQTT Demo`
4. 退出并保存配置。

1.2.2 设置 Wi-Fi 参数

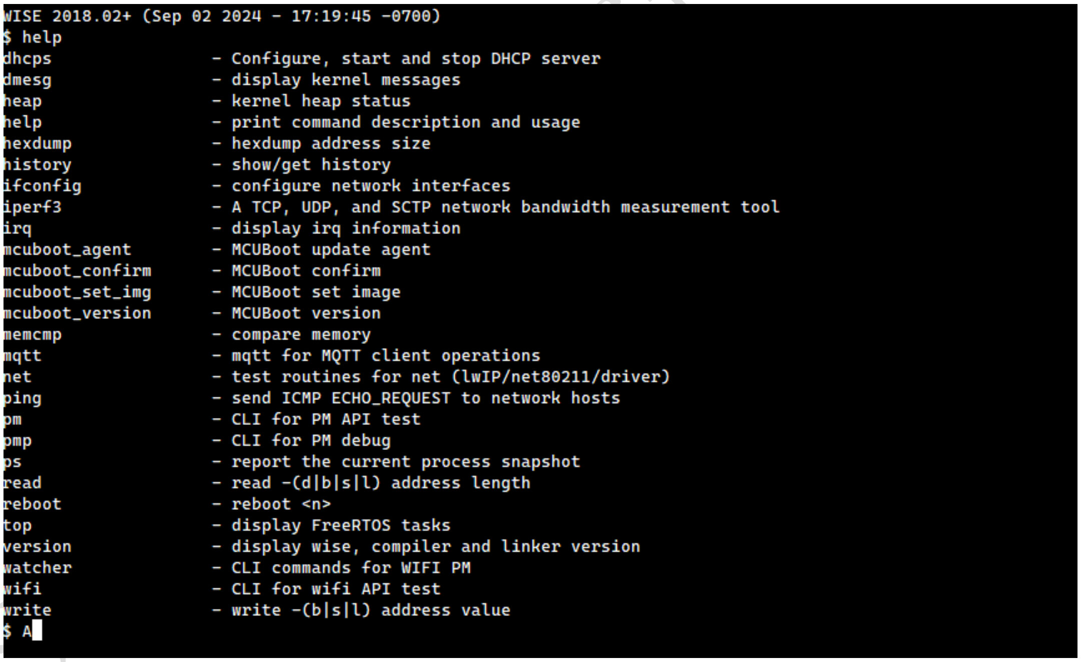
1. 打开配置菜单：
   ```
   $ make menuconfig
   ```
2. 进入以下路径：
   `Applications -> Common -> include WI-FI Configuration`
3. 在以下路径中输入所需的 Wi-Fi 参数 :
   `DEMO WI-FI Configuration`
   (如果需要，可以使用帮助菜单查看每个选项的说明。)

4. 退出并保存配置。

1.2.3 配置 MQTT 客户端参数

1. 再次打开配置菜单：
   $ make menuconfig
2. 进入以下路径：
   `Applications -> MQTT demo`
3. 根据需要修改 MQTT 客户端参数。

4. 构建固件映像文件：wise-mcuboot.bin.
   $ make
5. 请参考 《SCM1612_SDK 入门指南.pdf》获取如何下载生成的 wise-mcuboot.bin 映像文件并将其烧录至 SCM1612 开发板上的操作说明。



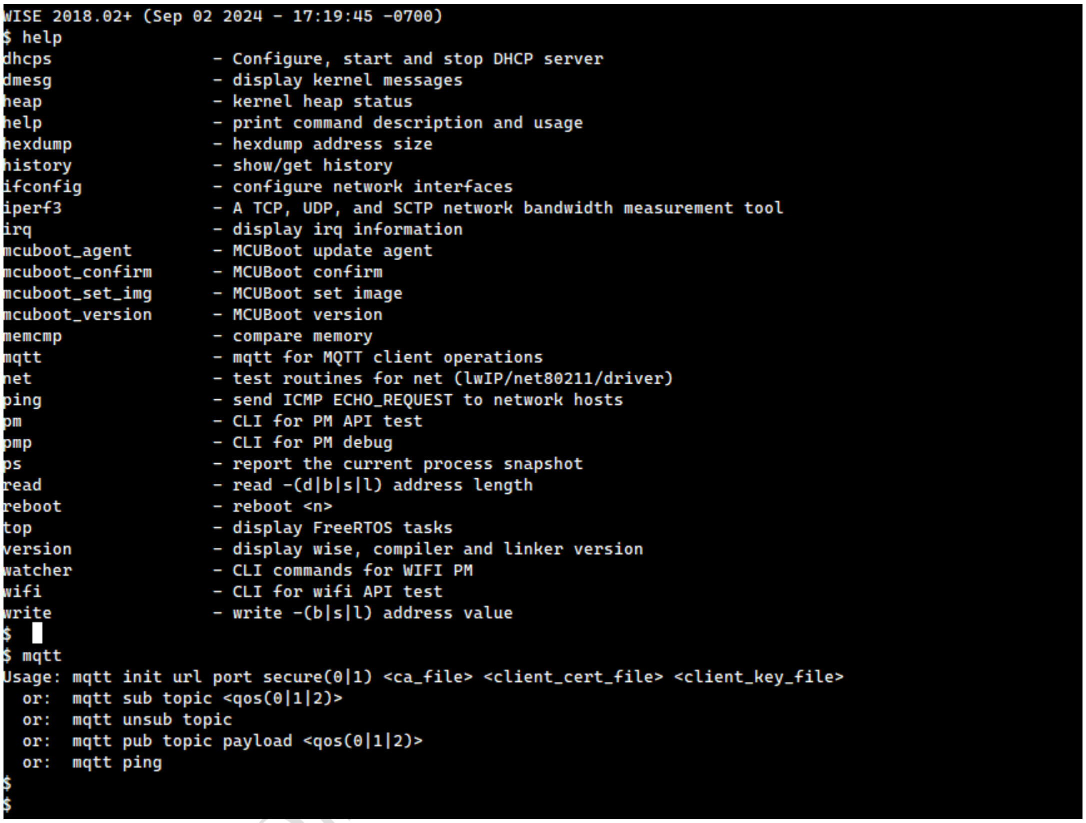
```
WISE 2018.02+ (Sep 02 2024 - 17:19:45 -0700)
$ help
dhcps                - Configure, start and stop DHCP server
dmesg                - display kernel messages
heap                 - kernel heap status
help                 - print command description and usage
hexdump              - hexdump address size
history              - show/get history
ifconfig             - configure network interfaces
iperf3               - A TCP, UDP, and SCTP network bandwidth measurement tool
irq                  - display irq information
mcuboot_agent        - MCUBoot update agent
mcuboot_confirm      - MCUBoot confirm
mcuboot_set_img      - MCUBoot set image
mcuboot_version      - MCUBoot version
memcmp               - compare memory
mqtt                 - mqtt for MQTT client operations
net                  - test routines for net (lwIP/net80211/driver)
ping                 - send ICMP ECHO_REQUEST to network hosts
pm                   - CLI for PM API test
pmp                  - CLI for PM debug
ps                   - report the current process snapshot
read                 - read -(d|b|s|l) address length
reboot               - reboot <n>
top                  - display FreeRTOS tasks
version              - display wise, compiler and linker version
watcher              - CLI commands for WIFI PM
wifi                 - CLI for wifi API test
write                - write -(b|s|l) address value
$ A
```

1.2.4 运行示例

要运行 MQTT 示例，你需要一个独立的 MQTT 客户端，它将与同一个 MQTT 服务器 test.mosquitto.org 交互，可以作为发布者（Publisher）或订阅者

（Subscriber）。本示例中使用了 Eclipse mosquito 的 PC 版本作为测试客户端。

- 该示例允许使用 CLI 命令与 MQTT 客户端进行交互式测试。



```
WISE 2018.02+ (Sep 02 2024 - 17:19:45 -0700)
$ help
dhcps                    - Configure, start and stop DHCP server
dmesg                    - display kernel messages
heap                     - kernel heap status
help                     - print command description and usage
hexdump                  - hexdump address size
history                  - show/get history
ifconfig                 - configure network interfaces
iperf3                   - A TCP, UDP, and SCTP network bandwidth measurement tool
irq                      - display irq information
mcuboot_agent            - MCUBoot update agent
mcuboot_confirm          - MCUBoot confirm
mcuboot_set_img          - MCUBoot set image
mcuboot_version          - MCUBoot version
memcmp                   - compare memory
mqtt                     - mqtt for MQTT client operations
net                      - test routines for net (lwIP/net80211/driver)
ping                     - send ICMP ECHO_REQUEST to network hosts
pm                       - CLI for PM API test
pmp                      - CLI for PM debug
ps                       - report the current process snapshot
read                     - read -(d|b|s|l) address length
reboot                   - reboot <n>
top                      - display FreeRTOS tasks
version                  - display wise, compiler and linker version
watcher                  - CLI commands for WIFI PM
wifi                     - CLI for wifi API test
write                    - write -(b|s|l) address value
$ 
$ mqtt
Usage: mqtt init url port secure(0|1) <ca_file> <client_cert_file> <client_key_file>
  or:  mqtt sub topic <qos(0|1|2)>
  or:  mqtt unsub topic
  or:  mqtt pub topic payload <qos(0|1|2)>
  or:  mqtt ping
$
$
```

1.2.5 初始化 MQTT 客户端

MQTT 客户端可以通过 mqtt init CLI 命令进行初始化和启动。该命令需要以下几个参数：

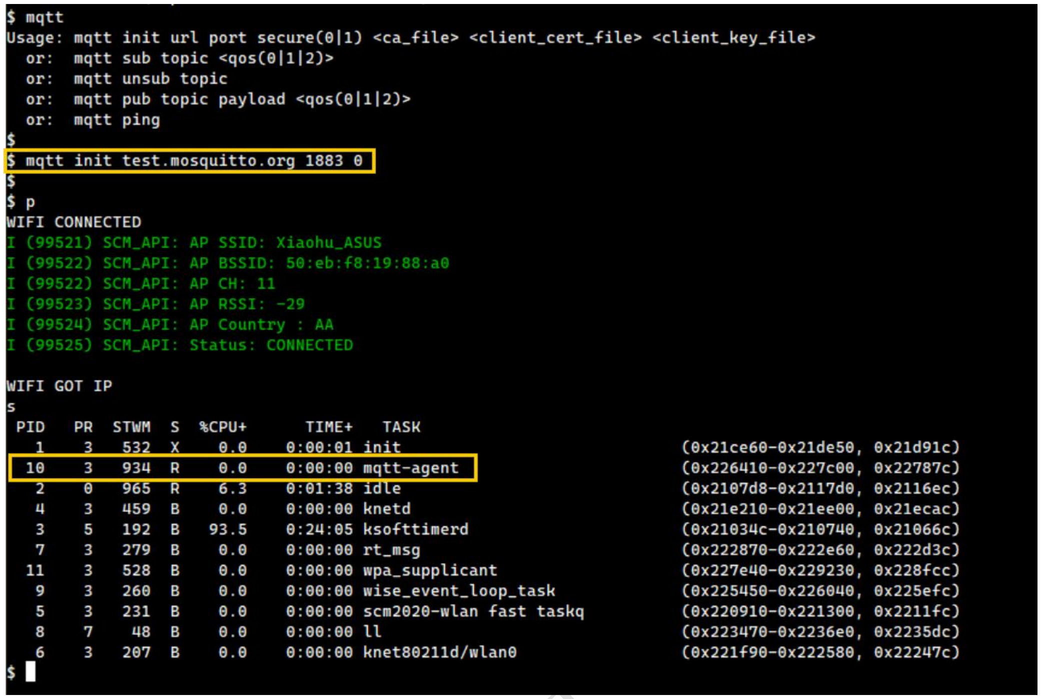
参数	含义	是否必填	示例
url	MQTT 服务器的 URL	M (必填)	test.mosquitto.org
port	MQTT 服务器的端口号	M (必填)	1883 (明文传输) 8883 (加密传输，无认证)

			8884 (加密传输，有认证)
secure	0：明文 TCP 传输 1：TLS 传输	M (必填)	
ca_file	CA 证书文件的完整路径	secure 为 1 时必填	/path/to/ca.crt
client_cert_file	客户端证书文件的完整路径	secure 为 1 时可选	/path/to/client.crt
client_key_file	客户端密钥文件的完整路径	secure 为 1 时可选	/path/to/client.key

1.2.5.1 使用明文 TCP 传输

以下是使用明文 TCP 传输初始化并启动 MQTT 客户端的 CLI 命令示例。该客户端将连接到已配置的 MQTT 服务器。

Wi-Fi 参数应在构建过程中已配置完成。当 mqtt init 命令运行时，SCM1612 设备将自动连接到已配置的接入点（AP）。

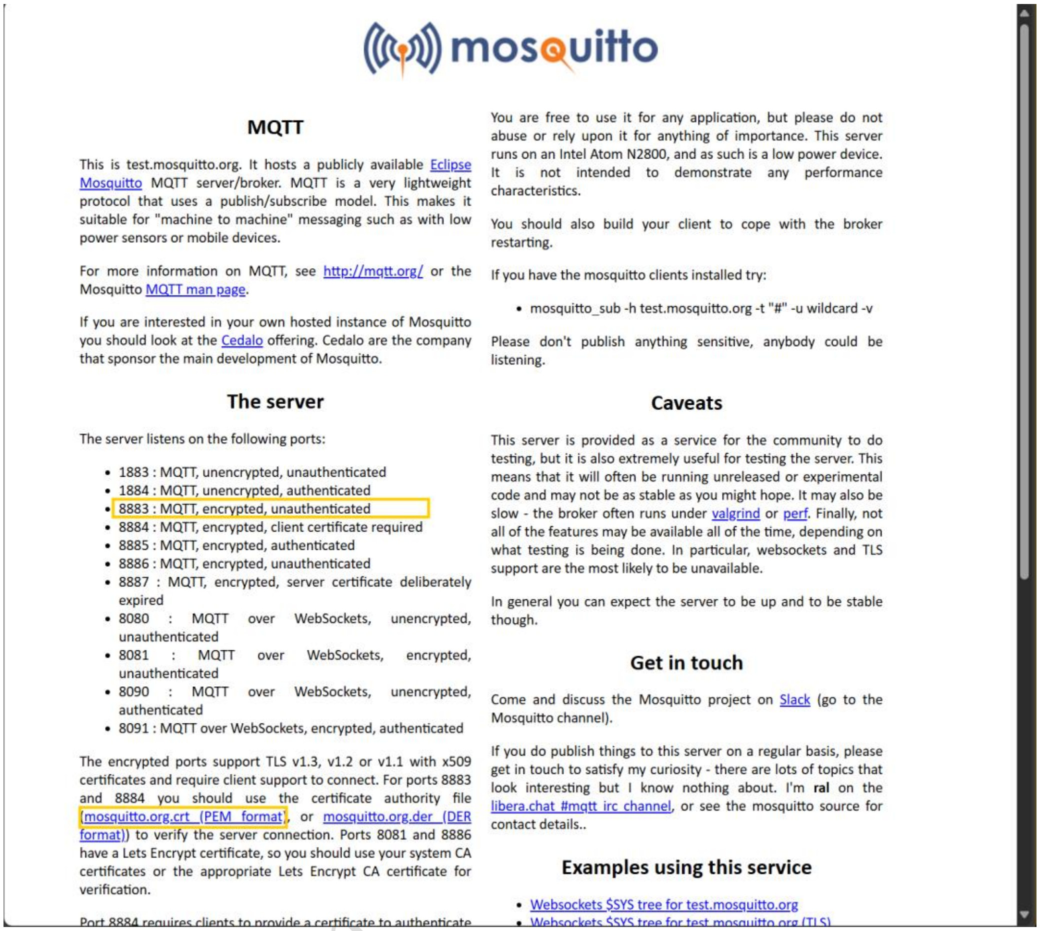


```
$ mqtt
Usage: mqtt init url port secure(0|1) <ca_file> <client_cert_file> <client_key_file>
  or:  mqtt sub topic <qos(0|1|2)>
  or:  mqtt unsub topic
  or:  mqtt pub topic payload <qos(0|1|2)>
  or:  mqtt ping
$
$ mqtt init test.mosquitto.org 1883 0
$
$ p
WIFI CONNECTED
I (99521) SCM_API: AP SSID: Xiaohu_ASUS
I (99522) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (99522) SCM_API: AP CH: 11
I (99523) SCM_API: AP RSSI: -29
I (99524) SCM_API: AP Country : AA
I (99525) SCM_API: Status: CONNECTED

WIFI GOT IP
s
 PID   PR  STWM  S  %CPU+      TIME+   TASK
   1    3   532  X    0.0    0:00:01 init                           (0x21ce60-0x21de50, 0x21d91c)
  10    3   934  R    0.0    0:00:00 mqtt-agent                     (0x226410-0x227c00, 0x22787c)
   2    0   965  R    6.3    0:01:38 idle                           (0x2107d8-0x2117d0, 0x2116ec)
   4    3   459  B    0.0    0:00:00 knetd                          (0x21e210-0x21ee00, 0x21ecac)
   3    5   192  B   93.5    0:24:05 ksofttimerd                    (0x21034c-0x210740, 0x21066c)
   7    3   279  B    0.0    0:00:00 rt_msg                         (0x222870-0x222e60, 0x222d3c)
  11    3   528  B    0.0    0:00:00 wpa_supplicant                 (0x227e40-0x229230, 0x228fcc)
   9    3   260  B    0.0    0:00:00 wise_event_loop_task           (0x225450-0x226040, 0x225efc)
   5    3   231  B    0.0    0:00:00 scm2020-wlan fast taskq        (0x220910-0x221300, 0x2211fc)
   8    7    48  B    0.0    0:00:00 ll                             (0x223470-0x2236e0, 0x2235dc)
   6    3   207  B    0.0    0:00:00 knet80211d/wlan0               (0x221f90-0x222580, 0x22247c)
$ 
```

1.2.5.2 使用 TLS 加密传输（仅加密）

要通过 TLS 连接，需要在文件系统中存储相应的 CA 证书。例如，SCM1612 MQTT 客户端可以使用 CA 证书（mosquitto.org.crt）连接到 test.mosquitto.org 的 8883 端口。



MQTT

This is test.mosquitto.org. It hosts a publicly available Eclipse Mosquitto MQTT server/broker. MQTT is a very lightweight protocol that uses a publish/subscribe model. This makes it suitable for "machine to machine" messaging such as with low power sensors or mobile devices.

For more information on MQTT, see http://mqtt.org/ or the Mosquitto MQTT man page.

If you are interested in your own hosted instance of Mosquitto you should look at the Cedalo offering. Cedalo are the company that sponsor the main development of Mosquitto.

The server

The server listens on the following ports:

- 1883 : MQTT, unencrypted, unauthenticated
- 1884 : MQTT, unencrypted, authenticated
- 8883 : MQTT, encrypted, unauthenticated
- 8884 : MQTT, encrypted, client certificate required
- 8885 : MQTT, encrypted, authenticated
- 8886 : MQTT, encrypted, unauthenticated
- 8887 : MQTT, encrypted, server certificate deliberately expired
- 8080 : MQTT over WebSockets, unencrypted, unauthenticated
- 8081 : MQTT over WebSockets, encrypted, unauthenticated
- 8090 : MQTT over WebSockets, unencrypted, authenticated
- 8091 : MQTT over WebSockets, encrypted, authenticated

The encrypted ports support TLS v1.3, v1.2 or v1.1 with x509 certificates and require client support to connect. For ports 8883 and 8884 you should use the certificate authority file (mosquitto.org.crt (PEM format), or mosquitto.org.der (DER format)) to verify the server connection. Ports 8081 and 8886 have a Lets Encrypt certificate, so you should use your system CA certificates or the appropriate Lets Encrypt CA certificate for verification.

Port 8884 requires clients to provide a certificate to authenticate

You are free to use it for any application, but please do not abuse or rely upon it for anything of importance. This server runs on an Intel Atom N2800, and as such is a low power device. It is not intended to demonstrate any performance characteristics.

You should also build your client to cope with the broker restarting.

If you have the mosquitto clients installed try:

- mosquitto_sub -h test.mosquitto.org -t "#" -u wildcard -v

Please don't publish anything sensitive, anybody could be listening.

Caveats

This server is provided as a service for the community to do testing, but it is also extremely useful for testing the server. This means that it will often be running unreleased or experimental code and may not be as stable as you might hope. It may also be slow - the broker often runs under valgrind or perf. Finally, not all of the features may be available all of the time, depending on what testing is being done. In particular, websockets and TLS support are the most likely to be unavailable.

In general you can expect the server to be up and to be stable though.

Get in touch

Come and discuss the Mosquitto project on Slack (go to the Mosquitto channel).

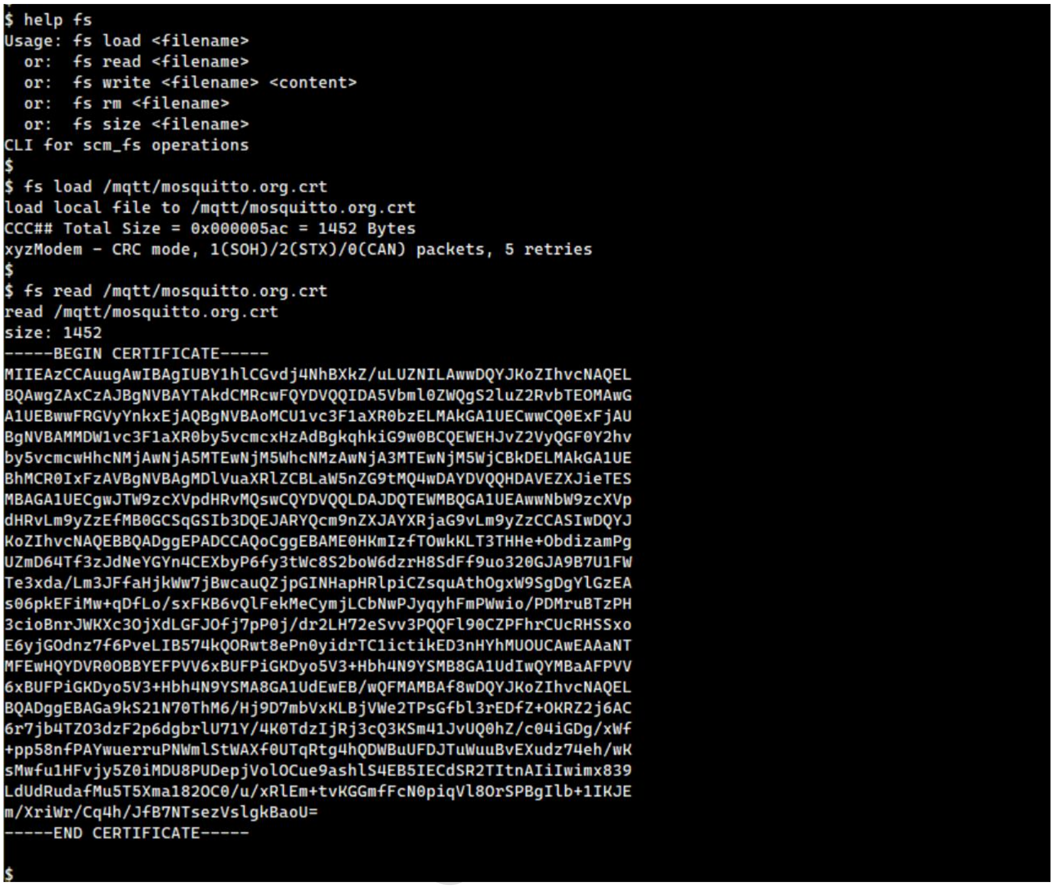
If you do publish things to this server on a regular basis, please get in touch to satisfy my curiosity - there are lots of topics that look interesting but I know nothing about. I'm **ral** on the libera.chat #mqtt irc channel, or see the mosquitto source for contact details..

Examples using this service

- Websockets $SYS tree for test.mosquitto.org
- Websockets $SYS tree for test.mosquitto.org (TLS)

步骤：

1. 从 test.mosquitto.org 下载 CA 证书到你的 PC 上。
2. 使用 fs load CLI 命令将证书文件加载到目标设备的文件系统中。



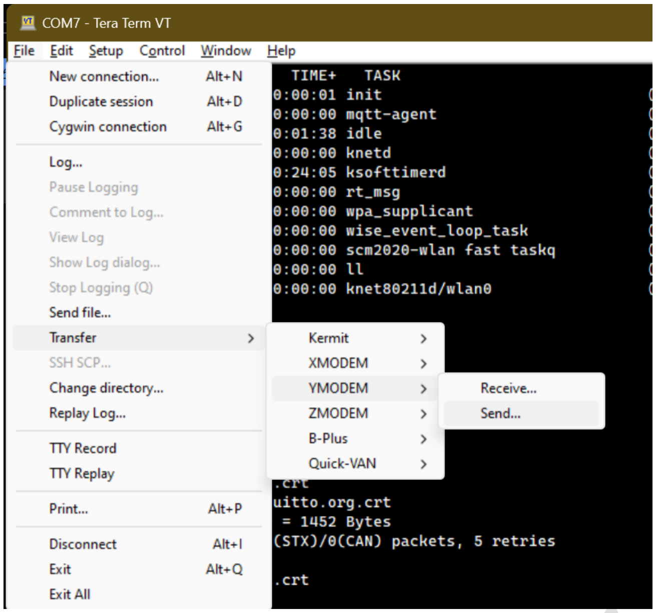
```
$ help fs
Usage: fs load <filename>
  or:  fs read <filename>
  or:  fs write <filename> <content>
  or:  fs rm <filename>
  or:  fs size <filename>
CLI for scm_fs operations
$
$ fs load /mqtt/mosquitto.org.crt
load local file to /mqtt/mosquitto.org.crt
CCC## Total Size = 0x000005ac = 1452 Bytes
xyzModem - CRC mode, 1(SOH)/2(STX)/0(CAN) packets, 5 retries
$
$ fs read /mqtt/mosquitto.org.crt
read /mqtt/mosquitto.org.crt
size: 1452
-----BEGIN CERTIFICATE-----
MIIEAzCCAuugAwIBAgIUBY1hlCGvdj4NhBXkZ/uLUZNILAwwDQYJKoZIhvcNAQEL
BQAwgZAxCzAJBgNVBAYTAkdCMRcwFQYDVQQIDA5Vbml0ZWQgS2luZ2RvbTEOMAwG
A1UEBwwFRGVyYnkxEjAQBgNVBAoMCU1vc3F1aXR0bzELMAkGA1UECwwCQ0ExFjAU
BgNVBAMMDW1vc3F1aXR0by5vcmcxHzAdBgkqhkiG9w0BCQEWEHJvZ2VyQGF0Y2hv
by5vcmcwHhcNMjAwNjA5MTEwNjM5WhcNMzAwNjA3MTEwNjM5WjCBkDELMAkGA1UE
BhMCR0IxFzAVBgNVBAgMDlVuaXRlZCBLaW5nZG9tMQ4wDAYDVQQHDAVEZXJieTES
MBAGA1UECgwJTW9zcXVpdHRvMQswCQYDVQQLDAJDQTEWMBQGA1UEAwwNbW9zcXVp
dHRvLm9yZzEfMB0GCSqGSIb3DQEJARYQcm9nZXJAYXRjaG9vLm9yZzCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAME0HKmIzfTOwkKLT3THHe+ObdizamPg
UZmD64Tf3zJdNeYGYn4CEXbyP6fy3tWc8S2boW6dzrH8SdFf9uo320GJA9B7U1FW
Te3xda/Lm3JFfaHjkWw7jBwcauQZjpGINHapHRlpiCZsquAthOgxW9SgDgYlGzEA
s06pkEFiMw+qDfLo/sxFKB6vQlFekMeCymjLCbNwPJyqyhFmPWwio/PDMruBTzPH
3cioBnrJWKXc3OjXdLGFJOfj7pP0j/dr2LH72eSvv3PQQFl90CZPFhrCUcRHSSxo
E6yjGOdnz7f6PveLIB574kQORwt8ePn0yidrTC1ictikED3nHYhMUOUCAwEAAaNT
MFEwHQYDVR0OBBYEFPVV6xBUFPiGKDyo5V3+Hbh4N9YSMB8GA1UdIwQYMBaAFPVV
6xBUFPiGKDyo5V3+Hbh4N9YSMA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEL
BQADggEBAGa9kS21N70ThM6/Hj9D7mbVxKLBjVWe2TPsGfbl3rEDfZ+OKRZ2j6AC
6r7jb4TZO3dzF2p6dgbrlU71Y/4K0TdzIjRj3cQ3KSm41JvUQ0hZ/c04iGDg/xWf
+pp58nfPAYwuerruPNWmlStWAXf0UTqRtg4hQDWBuUFDJTuWuuBvEXudz74eh/wK
sMwfu1HFvjy5Z0iMDU8PUDepjVolOCue9ashlS4EB5IECdSR2TItnAIiIwimx839
LdUdRudafMu5T5Xma182OC0/u/xRlEm+tvKGGmfFcN0piqVl8OrSPBgIlb+1IKJE
m/XriWr/Cq4h/JfB7NTsezVslgkBaoU=
-----END CERTIFICATE-----

$
```

3. 文件传输可以使用 YMODEM 协议，具体取决于使用的终端程序。对于 Tera Term：

   - 进入 Transfer -> YMODEM ->

     Send，然后选择证书文件进行传输。

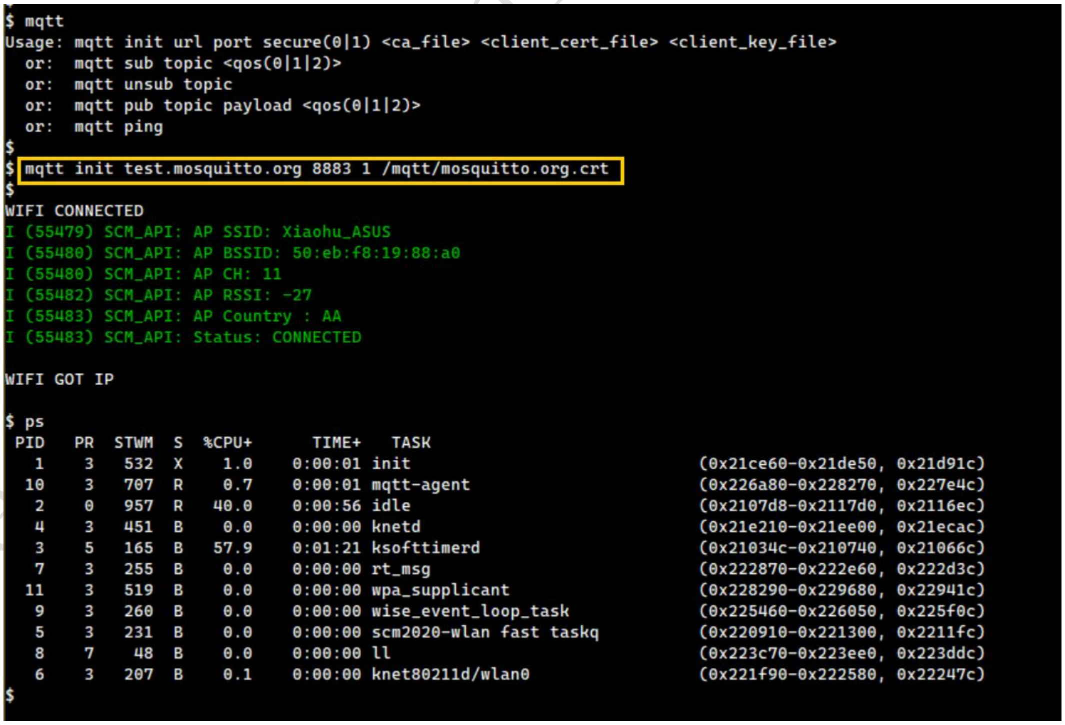


```
 TIME+   TASK
0:00:01 init
0:00:00 mqtt-agent
0:01:38 idle
0:00:00 knetd
0:24:05 ksofttimerd
0:00:00 rt_msg
0:00:00 wpa_supplicant
0:00:00 wise_event_loop_task
0:00:00 scm2020-wlan fast taskq
0:00:00 ll
0:00:00 knet80211d/wlan0

.crt
uitto.org.crt
 = 1452 Bytes
(STX)/0(CAN) packets, 5 retries

.crt
```

4. 传输完成后，启用 TLS 初始化并启动 MQTT 客户端。



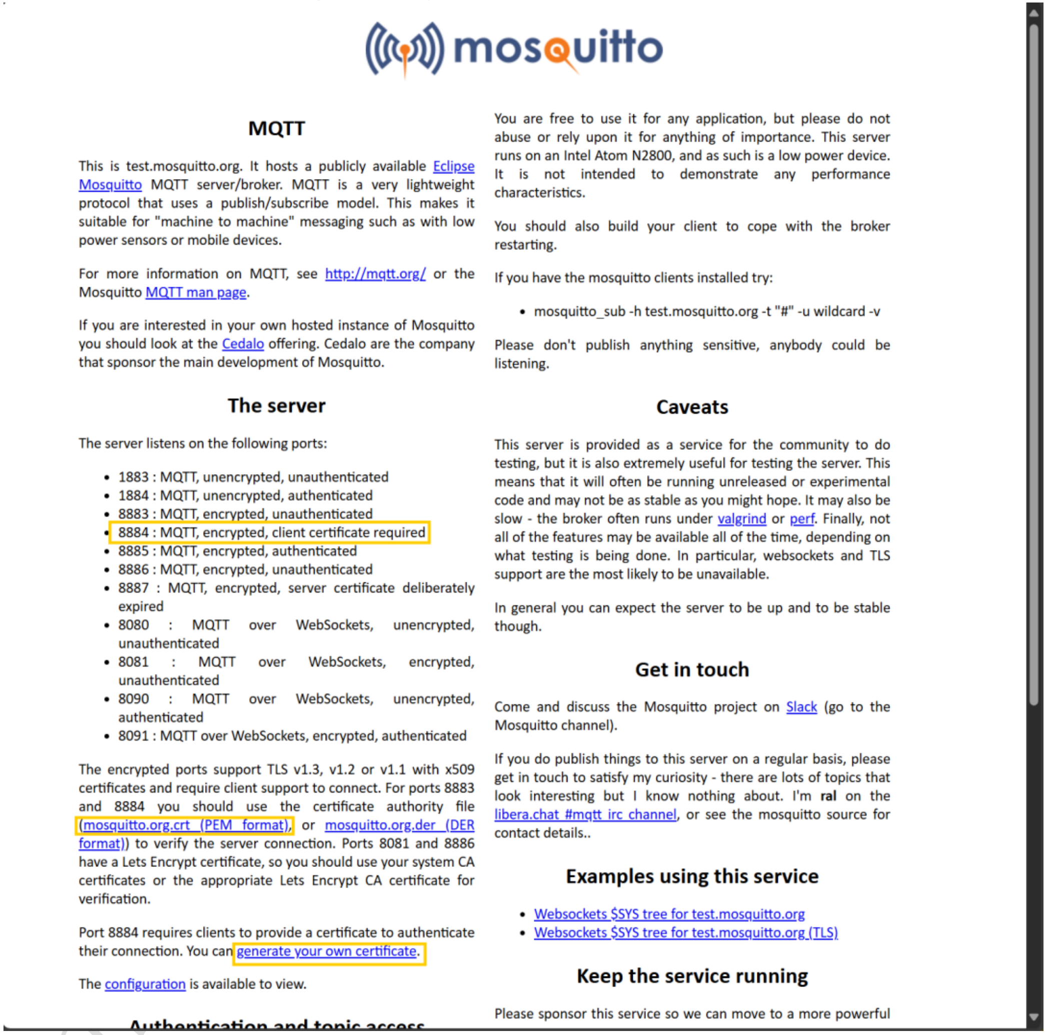
```
$ mqtt
Usage: mqtt init url port secure(0|1) <ca_file> <client_cert_file> <client_key_file>
  or:  mqtt sub topic <qos(0|1|2)>
  or:  mqtt unsub topic
  or:  mqtt pub topic payload <qos(0|1|2)>
  or:  mqtt ping
$
$ mqtt init test.mosquitto.org 8883 1 /mqtt/mosquitto.org.crt
$
WIFI CONNECTED
I (55479) SCM_API: AP SSID: Xiaohu_ASUS
I (55480) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (55480) SCM_API: AP CH: 11
I (55482) SCM_API: AP RSSI: -27
I (55483) SCM_API: AP Country : AA
I (55483) SCM_API: Status: CONNECTED

WIFI GOT IP

$ ps
 PID   PR  STWM  S  %CPU+     TIME+   TASK
   1    3   532  X    1.0   0:00:01 init                             (0x21ce60-0x21de50, 0x21d91c)
  10    3   707  R    0.7   0:00:01 mqtt-agent                       (0x226a80-0x228270, 0x227e4c)
   2    0   957  R   40.0   0:00:56 idle                             (0x2107d8-0x2117d0, 0x2116ec)
   4    3   451  B    0.0   0:00:00 knetd                            (0x21e210-0x21ee00, 0x21ecac)
   3    5   165  B   57.9   0:01:21 ksofttimerd                      (0x21034c-0x210740, 0x21066c)
   7    3   255  B    0.0   0:00:00 rt_msg                           (0x222870-0x222e60, 0x222d3c)
  11    3   519  B    0.0   0:00:00 wpa_supplicant                   (0x228290-0x229680, 0x22941c)
   9    3   260  B    0.0   0:00:00 wise_event_loop_task             (0x225460-0x226050, 0x225f0c)
   5    3   231  B    0.0   0:00:00 scm2020-wlan fast taskq          (0x220910-0x221300, 0x2211fc)
   8    7    48  B    0.0   0:00:00 ll                               (0x223c70-0x223ee0, 0x223ddc)
   6    3   207  B    0.1   0:00:00 knet80211d/wlan0                 (0x221f90-0x222580, 0x22247c)
$
```

1.2.5.3 使用 TLS 加密传输和客户端认证

要进行客户端认证，你需要准备客户端证书和客户端密钥。这些文件可以使用 openssl 工具生成，test.mosquitto.org 网站上有生成它们的详细指南。



mosquitto

MQTT

This is test.mosquitto.org. It hosts a publicly available Eclipse Mosquitto MQTT server/broker. MQTT is a very lightweight protocol that uses a publish/subscribe model. This makes it suitable for "machine to machine" messaging such as with low power sensors or mobile devices.

For more information on MQTT, see http://mqtt.org/ or the Mosquitto MQTT man page.

If you are interested in your own hosted instance of Mosquitto you should look at the Cedalo offering. Cedalo are the company that sponsor the main development of Mosquitto.

The server

The server listens on the following ports:

- 1883 : MQTT, unencrypted, unauthenticated
- 1884 : MQTT, unencrypted, authenticated
- 8883 : MQTT, encrypted, unauthenticated
- 8884 : MQTT, encrypted, client certificate required
- 8885 : MQTT, encrypted, authenticated
- 8886 : MQTT, encrypted, unauthenticated
- 8887 : MQTT, encrypted, server certificate deliberately expired
- 8080 : MQTT over WebSockets, unencrypted, unauthenticated
- 8081 : MQTT over WebSockets, encrypted, unauthenticated
- 8090 : MQTT over WebSockets, unencrypted, authenticated
- 8091 : MQTT over WebSockets, encrypted, authenticated

The encrypted ports support TLS v1.3, v1.2 or v1.1 with x509 certificates and require client support to connect. For ports 8883 and 8884 you should use the certificate authority file (mosquitto.org.crt (PEM format), or mosquitto.org.der (DER format)) to verify the server connection. Ports 8081 and 8886 have a Lets Encrypt certificate, so you should use your system CA certificates or the appropriate Lets Encrypt CA certificate for verification.

Port 8884 requires clients to provide a certificate to authenticate their connection. You can generate your own certificate.

The configuration is available to view.

Authentication and topic access

You are free to use it for any application, but please do not abuse or rely upon it for anything of importance. This server runs on an Intel Atom N2800, and as such is a low power device. It is not intended to demonstrate any performance characteristics.

You should also build your client to cope with the broker restarting.

If you have the mosquitto clients installed try:

- mosquitto_sub -h test.mosquitto.org -t "#" -u wildcard -v

Please don't publish anything sensitive, anybody could be listening.

Caveats

This server is provided as a service for the community to do testing, but it is also extremely useful for testing the server. This means that it will often be running unreleased or experimental code and may not be as stable as you might hope. It may also be slow - the broker often runs under valgrind or perf. Finally, not all of the features may be available all of the time, depending on what testing is being done. In particular, websockets and TLS support are the most likely to be unavailable.

In general you can expect the server to be up and to be stable though.

Get in touch

Come and discuss the Mosquitto project on Slack (go to the Mosquitto channel).

If you do publish things to this server on a regular basis, please get in touch to satisfy my curiosity - there are lots of topics that look interesting but I know nothing about. I'm **ral** on the libera.chat #mqtt irc channel, or see the mosquitto source for contact details..

Examples using this service

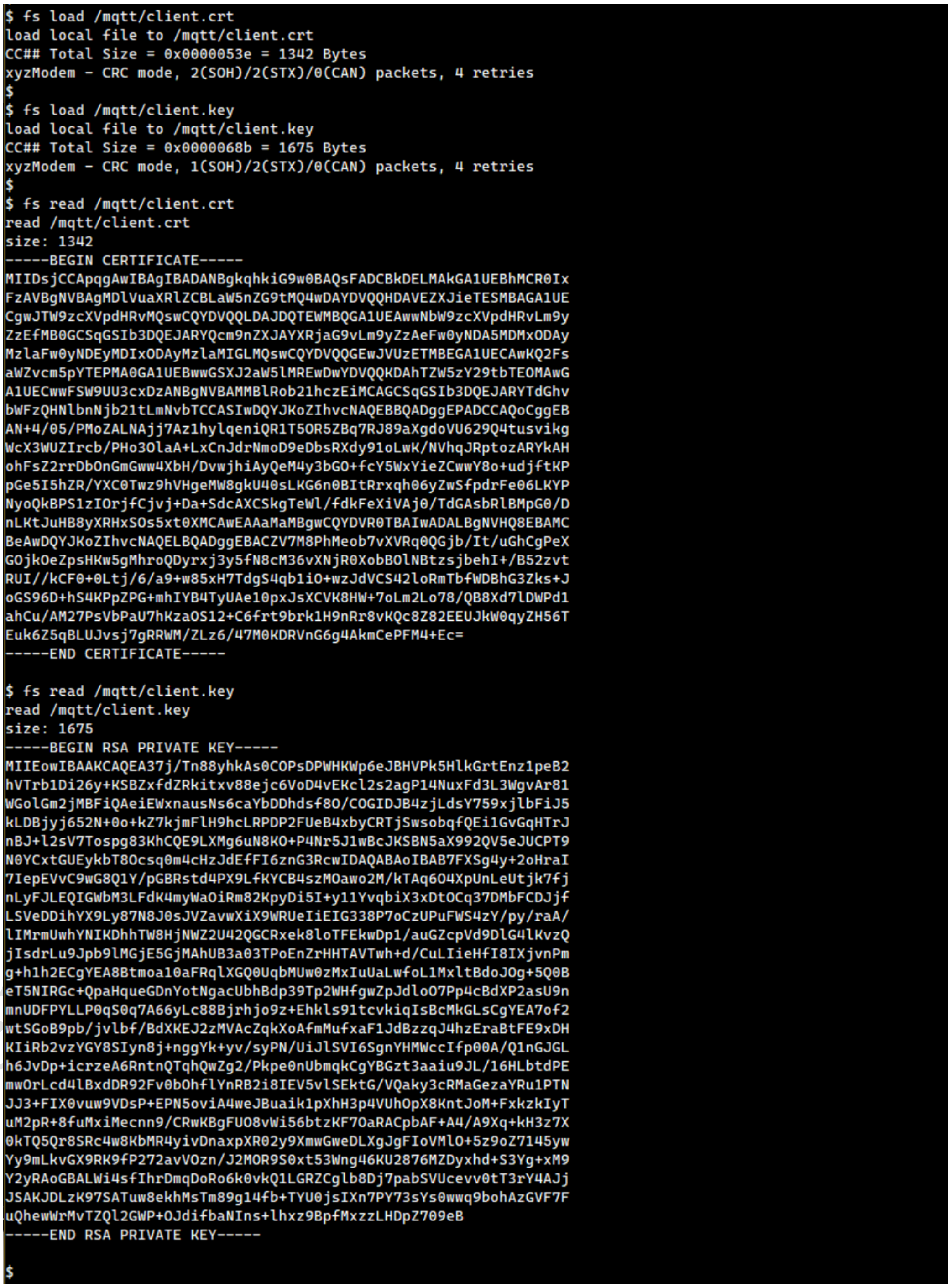
- Websockets $SYS tree for test.mosquitto.org
- Websockets $SYS tree for test.mosquitto.org (TLS)

Keep the service running

Please sponsor this service so we can move to a more powerful

1. 生成客户端证书和密钥。

2. 使用 fs load CLI 命令将客户端证书和密钥文件加载到目标设备中（与 CA 证书的加载方式相同）。

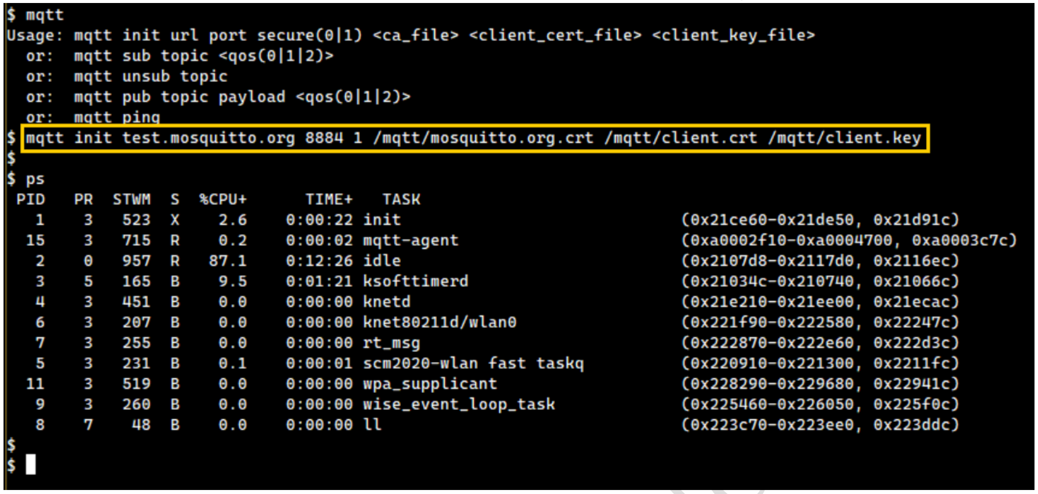


```
$ fs load /mqtt/client.crt
load local file to /mqtt/client.crt
CC## Total Size = 0x0000053e = 1342 Bytes
xyzModem - CRC mode, 2(SOH)/2(STX)/0(CAN) packets, 4 retries
$
$ fs load /mqtt/client.key
load local file to /mqtt/client.key
CC## Total Size = 0x0000068b = 1675 Bytes
xyzModem - CRC mode, 1(SOH)/2(STX)/0(CAN) packets, 4 retries
$
$ fs read /mqtt/client.crt
read /mqtt/client.crt
size: 1342
-----BEGIN CERTIFICATE-----
MIIDsjCCApqgAwIBAgIBADANBgkqhkiG9w0BAQsFADCBkDELMAkGA1UEBhMCR0Ix
FzAVBgNVBAgMDlVuaXRlZCBLaW5nZG9tMQ4wDAYDVQQHDAVEZXJieTESMBAGA1UE
CgwJTW9zcXVpdHRvMQswCQYDVQQLDAJDQTEWMBQGA1UEAwwNbW9zcXVpdHRvLm9y
ZzEfMB0GCSqGSIb3DQEJARYQcm9nZXJAYXRjaG9vLm9yZzAeFw0yNDA5MDMxODAy
MzlaFw0yNDEyMDIxODAyMzlaMIGLMQswCQYDVQQGEwJVUzETMBEGA1UECAwKQ2Fs
aWZvcm5pYTEPMA0GA1UEBwwGSXJ2aW5lMREwDwYDVQQKDAhTZW5zY29tbTEOMAwG
A1UECwwFSW9UU3cxDzANBgNVBAMMBlRob21hczEiMCAGCSqGSIb3DQEJARYTdGhv
bWFzQHNlbnNjb21tLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AN+4/05/PMoZALNAjj7Az1hylqeniQR1T5OR5ZBq7RJ89aXgdoVU629Q4tusvikg
WcX3WUZIrcb/PHo3OlaA+LxCnJdrNmoD9eDbsRXdy91oLwK/NVhqJRptozARYkAH
ohFsZ2rrDbOnGmGww4XbH/DvwjhiAyQeM4y3bGO+fcY5WxYieZCwwY8o+udjftKP
pGe5I5hZR/YXC0Twz9hVHgeMW8gkU40sLKG6n0BItRrxqh06yZwSfpdrFe06LKYP
NyoQkBPS1zIOrjfCjvj+Da+SdcAXCSkgTeWl/fdkFeXiVAj0/TdGAsbRlBMpG0/D
nLKtJuHB8yXRHxSOs5xt0XMCAwEAAaMaMBgwCQYDVR0TBAIwADALBgNVHQ8EBAMC
BeAwDQYJKoZIhvcNAQELBQADggEBACZV7M8PhMeob7vXVRq0QGjb/It/uGhCgPeX
GOjkOeZpsHKw5gMhroQDyrxj3y5fN8cM36vXNjR0XobBOlNBtzsjbehI+/B52zvt
RUI//kCF0+0Ltj/6/a9+w85xH7TdgS4qb1iO+wzJdVCS42loRmTbfWDBhG3Zks+J
oGS96D+hS4KPpZPG+mhIYB4TyUAe10pxJsXCVK8HW+7oLm2Lo78/QB8Xd7lDWPd1
ahCu/AM27PsVbPaU7hKzaOS12+C6frt9brk1H9nRr8vKQc8Z82EEUJkW0qyZH56T
Euk6Z5qBLUJvsj7gRRWM/ZLz6/47M0KDRVnG6g4AkmCePFM4+Ec=
-----END CERTIFICATE-----

$ fs read /mqtt/client.key
read /mqtt/client.key
size: 1675
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA37j/Tn88yhkAs0COPsDPWHKWp6eJBHVPk5HlkGrtEnz1peB2
hVTrb1Di26y+KSBZxfdZRkitxv88ejc6VoD4vEKcl2s2agP14NuxFd3L3WgvAr81
WGolGm2jMBFiQAeiEWxnausNs6caYbDDhdsf8O/COGIDJB4zjLdsY759xjlbFiJ5
kLDBjyj652N+0o+kZ7kjmFlH9hcLRPDP2FUeB4xbyCRTjSwsobqfQEi1GvGqHTrJ
nBJ+l2sV7Tospg83KhCQE9LXMg6uN8KO+P4Nr5J1wBcJKSBN5aX992QV5eJUCPT9
N0YCxtGUEykbT8Ocsq0m4cHzJdEfFI6znG3RcwIDAQABAoIBAB7FXSg4y+2oHraI
7IepEVvC9wG8Q1Y/pGBRstd4PX9LfKYCB4szMOawo2M/kTAq6O4XpUnLeUtjk7fj
nLyFJLEQIGWbM3LFdK4myWaOiRm82KpyDi5I+y11YvqbiX3xDtOCq37DMbFCDJjf
LSVeDDihYX9Ly87N8J0sJVZavwXiX9WRUeIiEIG338P7oCzUPuFWS4zY/py/raA/
lIMrmUwhYNIKDhhTW8HjNWZ2U42QGCRxek8loTFEkwDp1/auGZcpVd9DlG4lKvzQ
jIsdrLu9Jpb9lMGjE5GjMAhUB3a03TPoEnZrHHTAVTwh+d/CuLIieHfI8IXjvnPm
g+h1h2ECgYEA8Btmoa10aFRqlXGQ0UqbMUw0zMxIuUaLwfoL1MxltBdoJOg+5Q0B
eT5NIRGc+QpaHqueGDnYotNgacUbhBdp39Tp2WHfgwZpJdloO7Pp4cBdXP2asU9n
mnUDFPYLLP0qS0q7A66yLc88Bjrhjo9z+Ehkls91tcvkiqIsBcMkGLsCgYEA7of2
wtSGoB9pb/jvlbf/BdXKEJ2zMVAcZqkXoAfmMufxaF1JdBzzqJ4hzEraBtFE9xDH
KIiRb2vzYGY8SIyn8j+nggYk+yv/syPN/UiJlSVI6SgnYHMWccIfp00A/Q1nGJGL
h6JvDp+icrzeA6RntnQTqhQwZg2/Pkpe0nUbmqkCgYBGzt3aaiu9JL/16HLbtdPE
mwOrLcd4lBxdDR92Fv0bOhflYnRB2i8IEV5vlSEktG/VQaky3cRMaGezaYRu1PTN
JJ3+FIX0vuw9VDsP+EPN5oviA4weJBuaik1pXhH3p4VUhOpX8KntJoM+FxkzkIyT
uM2pR+8fuMxiMecnn9/CRwKBgFUO8vWi56btzKF7OaRACpbAF+A4/A9Xq+kH3z7X
0kTQ5Qr8SRc4w8KbMR4yivDnaxpXR02y9XmwGweDLXgJgFIoVMlO+5z9oZ7145yw
Yy9mLkvGX9RK9fP272avVOzn/J2MOR9S0xt53Wng46KU2876MZDyxhd+S3Yg+xM9
Y2yRAoGBALWi4sfIhrDmqDoRo6k0vkQ1LGRZCglb8Dj7pabSVUcevv0tT3rY4AJj
JSAKJDLzK97SATuw8ekhMsTm89g14fb+TYU0jsIXn7PY73sYs0wwq9bohAzGVF7F
uQhewWrMvTZQl2GWP+OJdifbaNIns+lhxz9BpfMxzzLHDpZ709eB
-----END RSA PRIVATE KEY-----

$
```

3. 启用 TLS 并进行客户端认证，初始化并启动 MQTT 客户端。

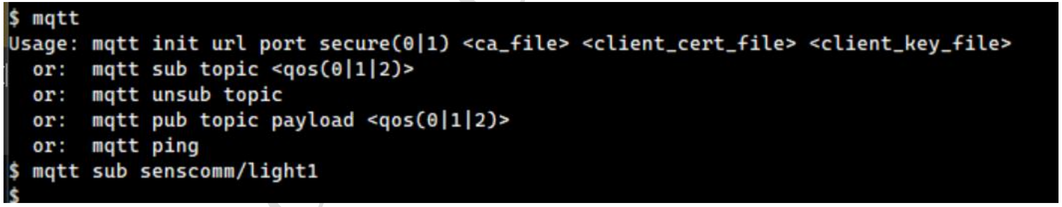


```
$ mqtt
Usage: mqtt init url port secure(0|1) <ca_file> <client_cert_file> <client_key_file>
  or:  mqtt sub topic <qos(0|1|2)>
  or:  mqtt unsub topic
  or:  mqtt pub topic payload <qos(0|1|2)>
  or:  mqtt ping
$ mqtt init test.mosquitto.org 8884 1 /mqtt/mosquitto.org.crt /mqtt/client.crt /mqtt/client.key
$
$ ps
 PID   PR  STWM  S  %CPU+      TIME+   TASK
   1    3   523  X    2.6    0:00:22 init                          (0x21ce60-0x21de50, 0x21d91c)
  15    3   715  R    0.2    0:00:02 mqtt-agent                    (0xa0002f10-0xa0004700, 0xa0003c7c)
   2    0   957  R   87.1    0:12:26 idle                          (0x2107d8-0x2117d0, 0x2116ec)
   3    5   165  B    9.5    0:01:21 ksofttimerd                   (0x21034c-0x210740, 0x21066c)
   4    3   451  B    0.0    0:00:00 knetd                         (0x21e210-0x21ee00, 0x21ecac)
   6    3   207  B    0.0    0:00:00 knet80211d/wlan0              (0x221f90-0x222580, 0x22247c)
   7    3   255  B    0.0    0:00:00 rt_msg                        (0x222870-0x222e60, 0x222d3c)
   5    3   231  B    0.1    0:00:01 scm2020-wlan fast taskq       (0x220910-0x221300, 0x2211fc)
  11    3   519  B    0.0    0:00:00 wpa_supplicant                (0x228290-0x229680, 0x22941c)
   9    3   260  B    0.0    0:00:00 wise_event_loop_task          (0x225460-0x226050, 0x225f0c)
   8    7    48  B    0.0    0:00:00 ll                            (0x223c70-0x223ee0, 0x223ddc)
$
$ █
```

1.2.6 测试订阅特定主题

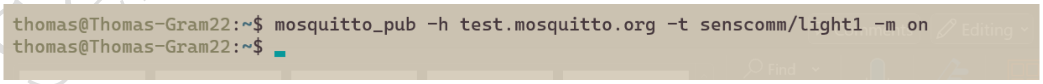
要测试订阅特定主题，请按以下步骤操作：

1. **使用 MQTT 客户端的 CLI 命令订阅主题：**



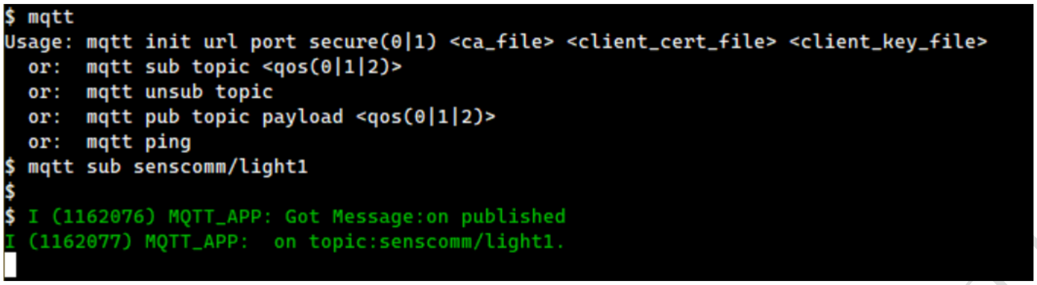
```
$ mqtt
Usage: mqtt init url port secure(0|1) <ca_file> <client_cert_file> <client_key_file>
  or:  mqtt sub topic <qos(0|1|2)>
  or:  mqtt unsub topic
  or:  mqtt pub topic payload <qos(0|1|2)>
  or:  mqtt ping
$ mqtt sub senscomm/light1
$
```

2. **在 PC 上使用另一个 MQTT 客户端（如 Eclipse Mosquitto）向相同的主题发布消息。**



```
thomas@Thomas-Gram22:~$ mosquitto_pub -h test.mosquitto.org -t senscomm/light1 -m on
thomas@Thomas-Gram22:~$ _
```

3. **检查 SCM1612 MQTT 客户端收到的消息。消息应显示在终端中。**

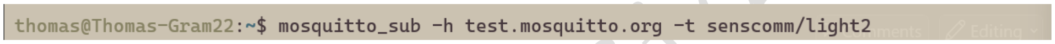


```
$ mqtt
Usage: mqtt init url port secure(0|1) <ca_file> <client_cert_file> <client_key_file>
  or:  mqtt sub topic <qos(0|1|2)>
  or:  mqtt unsub topic
  or:  mqtt pub topic payload <qos(0|1|2)>
  or:  mqtt ping
$ mqtt sub senscomm/light1
$
$ I (1162076) MQTT_APP: Got Message:on published
I (1162077) MQTT_APP:  on topic:senscomm/light1.
```

1.2.7 测试向特定主题发布消息

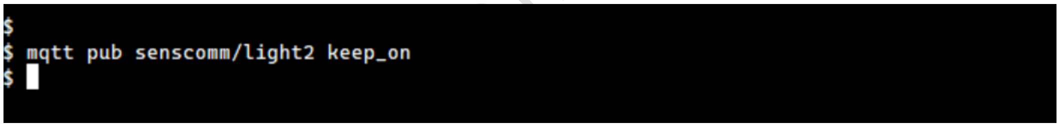
要测试向特定主题发布消息，请按以下步骤操作：

1. **在 PC 的 MQTT 客户端中订阅主题。**



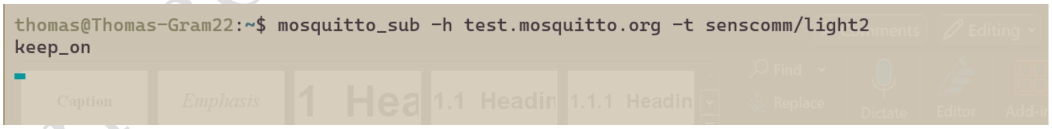
```
thomas@Thomas-Gram22:~$ mosquitto_sub -h test.mosquitto.org -t senscomm/light2
```

2. **使用 SCM1612 MQTT 客户端的 CLI 命令向相同主题发布消息。**



```
$
$ mqtt pub senscomm/light2 keep_on
$
```

3. **检查 PC 客户端收到的消息，消息应显示 SCM1612 客户端发布的内容。**



```
thomas@Thomas-Gram22:~$ mosquitto_sub -h test.mosquitto.org -t senscomm/light2
keep_on
```