
STM32F417xx 微控制器的安全套接字层 (SSL)

1 前言

STM32F417xx 微控制器的特点是具有完整的 10/100 Mb/s 以太网外设，可支持介质独立接口 (MII) 和精简的介质独立接口 (RMII)，通过这些接口可以连接物理层 (PHY)；此外还支持 IP、UDP、TCP 及 ICMP 协议的硬件校验。

STM32F417xx 的高级特性之一是采用硬件加密处理器，用于实现 AES/128/192/256、三重 DES、DES、SHA-1、MD5 和 RNG。

它支持安全套接字层 (SSL) 与传输层安全性 (TLS) 密码协议，可确保网络（如 Internet）通信安全，使得客户端和服务端应用程序能够以私密、安全的方式通信。

本应用笔记的目的在于提供以免费 SSL/TLS 库（即 PolarSSL 库）为基础创建的演示包。

本应用笔记的结构如下：

- 第 2 节提供缩略语表。
- 第 3 节对 SSL/TLS 进行了综合介绍。
- 第 4 节介绍了 PolarSSL 库。
- 第 5 节介绍了 STM32F417xx 硬件加密处理器。
- 最后，第 6 节介绍 STM32F417xx 的演示包。

注：本应用笔记仅适用于 STM32F417xx 器件并使用 STM3241G-EVAL 板作为硬件平台，因为在 STM32F407xx 器件中并未嵌入加速加密功能。

目录

1	前言	1
2	缩写和定义	6
3	SSL/TLS 协议概述	7
3.1	SSL 应用层	7
3.2	SSL/TLS 协议的历史	8
3.3	SSL/TLS 子协议	8
3.3.1	SSL 握手协议	9
3.3.2	SSL 记录协议	11
3.3.3	SSL 报警协议	12
3.3.4	密码规格变更协议	12
4	PolarSSL 库	13
4.1	许可证	13
5	STM32F417xx 硬件密码技术	14
5.1	加密处理器	14
5.2	随机数发生器	14
5.3	散列处理器	14
6	演示包说明	15
6.1	演示包目录和固件组成	15
6.1.1	演示包目录	15
6.1.2	固件组成	16
6.2	演示设置	17
6.2.1	PHY 接口配置	17
6.2.2	MAC 地址和 IP 地址设置	17
6.2.3	STM3241G-EVAL 设置	18
6.3	如何使用演示程序	18
6.3.1	SSL 客户端演示程序	18
6.3.2	SSL 服务器演示程序	21
6.4	SSL 演示程序的存储器占用情况	24
6.4.1	SSL 客户端演示程序	24

	6.4.2	SSL 服务器演示程序	25
7		结论	26
8		参考资料	27
附录 A		补充信息.....	28
	A.1	流程图	28
	A.2	项目配置	30
	A.2.1	LwIP 配置	30
	A.2.2	PolarSSL 配置.....	31
	A.2.3	FreeRTOS 配置.....	32
	A.3	使用 Mozilla 7.0.1 运行 SSL 服务器演示程序	33
	A.4	使用 IE8 运行 SSL 服务器演示程序	35
9		版本历史	37

表格索引

表 1. 缩写和定义 6

表 2. STM3241G-EVAL 跳线配置 18

表 3. SSL 客户端演示程序的存储器占用情况 25

表 4. SSL 服务器演示程序存储器占用情况 25

表 5. SSL 服务器演示程序的 lwIP 选项 30

表 6. SSL 客户端演示程序的 lwIP 选项 30

表 7. PolarSSL 选项: SSL 服务器演示程序的 config.h 文件 31

表 8. PolarSSL 选项: SSL 客户端演示程序的 config.h 文件 32

表 9. SSL 客户端演示程序的 FreeRTOS 配置 32

表 10. SSL 服务器演示程序的 FreeRTOS 配置 32

表 11. 文档版本历史 37



图片索引

图 1.	SSL 应用架构	8
图 2.	SSL 子协议	8
图 3.	SSL 握手协议	9
图 4.	恢复 SSL 会话的握手协议	11
图 5.	SSL 记录协议	11
图 6.	演示包结构	15
图 7.	PolarSSL 与 LwIP 连接	17
图 8.	SSL 客户端演示程序架构	19
图 9.	SSL 客户端演示程序	20
图 10.	ssl_server 应用程序窗口	20
图 11.	超级终端窗口	21
图 12.	SSL 服务器演示程序架构	22
图 13.	SSL 服务器演示程序	23
图 14.	成功连接后显示的 HTML 页面	23
图 15.	超级终端 SSL 服务器连接状态	24
图 16.	SSL 客户端任务流程图	28
图 17.	SSL 服务器任务流程图	29
图 18.	不可信连接对话框 1	33
图 19.	不可信连接对话框 2	33
图 20.	Add Security Exception (添加安全例外) 对话框	34
图 21.	任务状态页面	34
图 22.	“无法显示网页”错误消息	35
图 23.	证书错误消息	36
图 24.	任务状态页面	36

2 缩写和定义

表 1. 缩写和定义

缩写	定义
AES	高级加密标准
ANSI	美国国家标准协会
API	应用程序编程接口
ARC4	所谓的 Rivest 密码 4
ARP	地址解析协议
CA	证书颁发机构
CBC	密码块链接
CTR	计数器
DES	数据加密标准
DHCP	动态主机配置协议
DHM	Diffie–Hellman 密钥交换
ECB	电子密码本
FIPS	联邦信息处理标准
HAVEGE	硬件易失性熵收集与扩展
HMAC	散列消息验证码
HTTP	超文本传输协议
HTTPS	安全超文本传输协议
ICMP	Internet 控制消息协议
IETF	Internet 工程任务组
IGMP	Internet 组管理协议
LwIP	轻量级 TCP/IP 协议栈
MAC	消息验证码
MAC 地址	介质访问控制地址
MCO	微控制器时钟输出
MD2	消息摘要算法 2
MII	介质独立接口
PPP	点对点协议
RMII	精简的介质独立接口
RNG	随机数发生器
RSA	Rivest 、 Shamir 和 Adleman
SHA-1	安全散列算法 1
SNMP	简单网络管理协议

表 1. 缩写和定义 (续)

缩写	定义
SSL	安全套接字层
TCP/IP	传输控制协议/Internet 协议
TLS	传输层安全性
UDP	用户数据报协议
URL	统一资源定位器
USART	通用同步和异步收发器

3 SSL/TLS 协议概述

安全套接字层 (SSL) 协议与传输层安全性 (TLS) 协议可确保 Internet 通信安全，使得客户端和服务端应用程序能够以私密、安全的方式通信。这些协议层位于传输协议（如 TCP/IP）之上。

SSL 是用于在服务器与客户端间创建加密链路的标准安全技术。此链路可确保所有通信数据的私密与安全。

SSL/TLS 主要有如下目标：

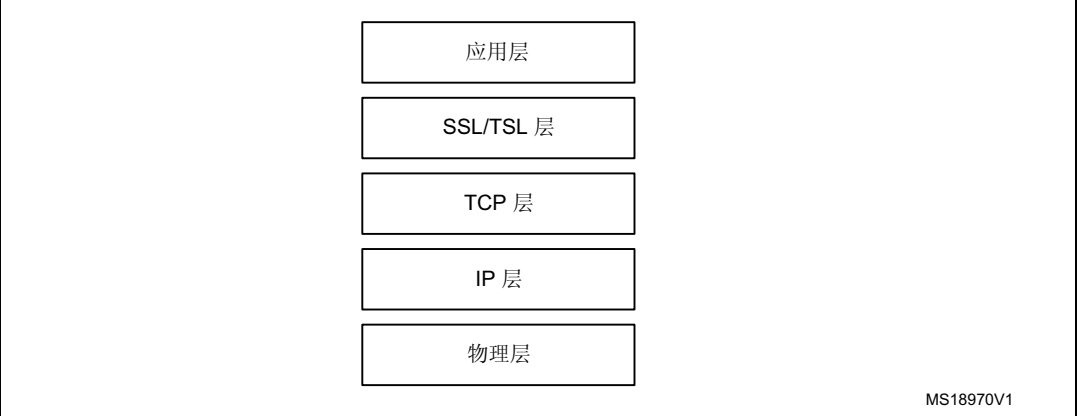
- 保持数据在通信应用程序双方之间的完整性。
- 保护服务器与客户端之间传送的信息安全。
- 验证与客户端通信的服务器。
- 允许客户端与服务器选择双方都支持的加密算法。
- 验证（可选）与服务器通信的客户端。
- 使用公钥加密技术生成共享密钥。
- 建立加密的 SSL 连接。

3.1 SSL 应用层

SSL/TLS 应用包括以下五层：

- 应用层：应用层指大多数应用程序进行网络通信所使用的高层级协议。
- SSL/TLS 层：SSL/TLS 层确保 Internet 通信安全。
- TCP 层：传输层负责独立于底层网络的端到端消息传输以及差错控制、分段、流控制、拥塞控制和应用程序寻址。
- IP 层：Internet 协议层负责主机寻址以及将数据包从源主机路由到目的主机。
- 物理层：物理层包含网络的基本硬件传输技术。

图 1. SSL 应用架构



3.2 SSL/TLS 协议的历史

1994 年，Netscape 开发了 SSL 用于确保 Internet 通信安全。不久之后，Internet 工程任务组 (IETF) 开始开发具有相同功能的标准协议。

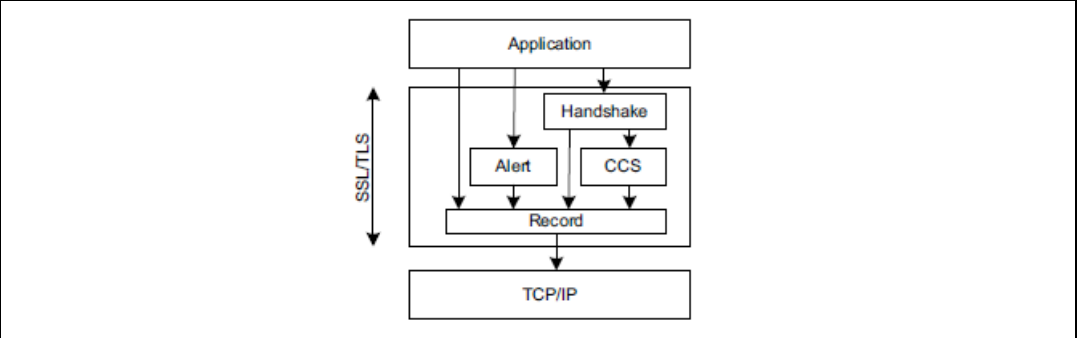
- SSL 1.0 (Netscape, 1993)：Netscape 内部设计。
- SSL 2.0 (Netscape, 1994)：该版本存在大量安全性缺陷。
- SSL 3.0 (Netscape, 1996)：所有 Internet 浏览器都支持该版本的协议。
- TLS 1.0 (IETF, 1999)：该版本在 RFC 2246 中被定义为 SSL 3.0 的升级版。“尽管 TLS 1.0 与 SSL 3.0 区别不大，但它们显然不支持相互操作”：[\[1\]: RFC 2246: TLS 协议 1.0 版](#)

注： 本文档中将 “SSL/TLS” 协议称为 “SSL”。

3.3 SSL/TLS 子协议

SSL 协议包括四个子协议：SSL 记录协议、SSL 握手协议、SSL 报警协议和 SSL 密码规格变更协议。

图 2. SSL 子协议

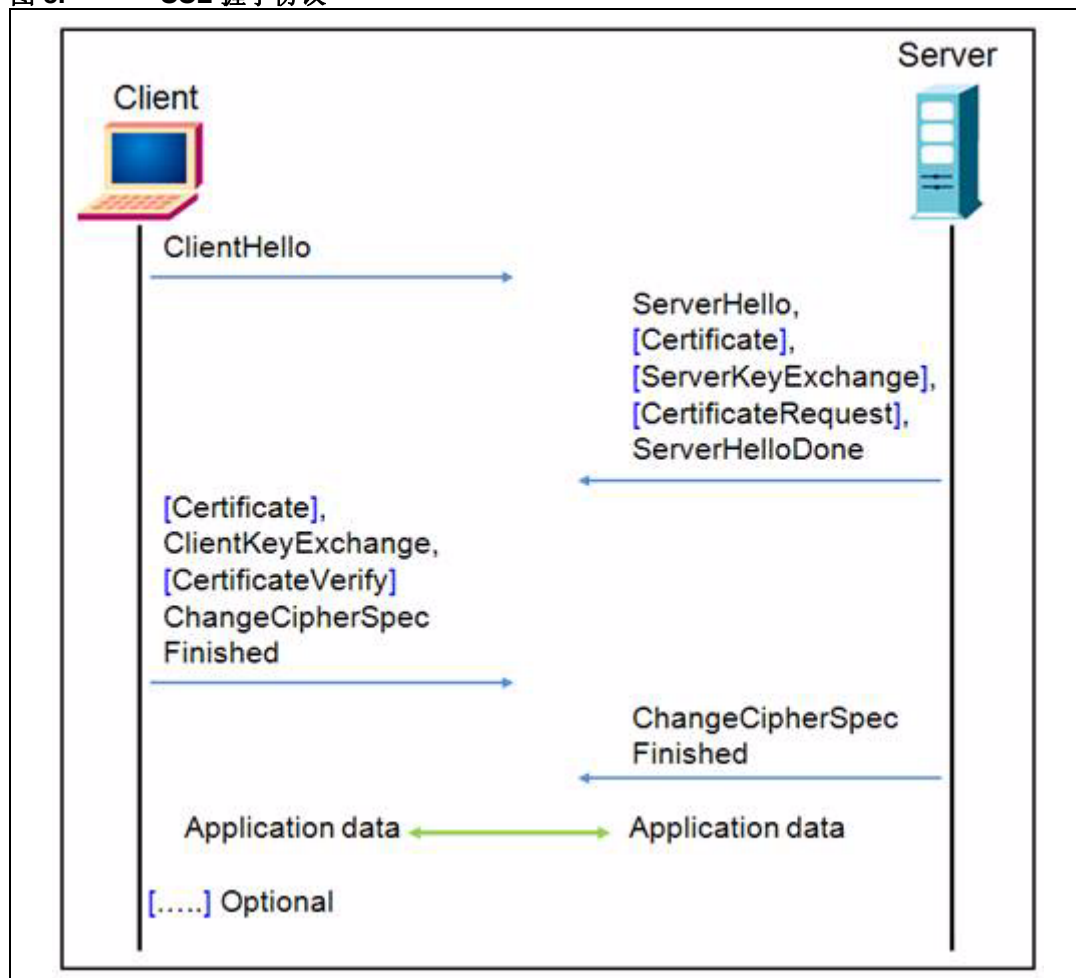


3.3.1 SSL 握手协议

SSL 会话状态由 SSL 握手协议控制。当 SSL 服务器与 SSL 客户端首次进行通信时，此协议即使用 SSL 记录协议在它们之间交换一系列消息。这种信息交换的目的是为了推进以下动作：

- 确定协议版本是 SSL 3.0 还是 TLS 1.0
- 允许客户端与服务器选择双方均支持的加密算法或密码文件
- 验证与客户端通信的服务器
- 验证（可选）与服务器通信的客户端
- 使用公钥加密技术生成共享密钥
- 建立加密的 SSL 连接

图 3. SSL 握手协议



如下是 SSL 握手协议的操作步骤：

1. 首先，客户端发送一条 **ClientHello** 消息，其中包含支持的 SSL 协议最高版本（SSL 3.0 版或 TLS 1.0 版）、一个随机数、支持的密码组合和压缩方法列表。
2. 服务器以一条 **ServerHello** 消息响应，其中包含所选的协议版本、另一个随机数、从客户端提供的列表中选择密码组合和压缩方法以及会话 ID。

注：客户端与服务器必须支持至少一个公用密码组合，否则握手协议失败。服务器通常选择双方都支持的最强公用密码组合。

3. 服务器以可选证书消息形式发送数字证书，例如，服务器使用 X.509 数字证书。
4. 如果未发送证书，服务器将发送一条可选的 **ServerKeyExchange** 消息，其中包含服务器的公用信息。
5. 如果服务器需要数字证书对客户端进行认证，则附加一条可选的 **CertificateRequest** 消息。
6. 服务器发送一条 **ServerHelloDone** 消息表示这一阶段的协商结束。
7. 如果服务器发送了 **CertificateRequest** 消息，则客户端必须在 **Certificate** 消息中发送其 X.509 客户端证书。
8. 客户端发送一条 **ClientKeyExchange** 消息。此消息包含用于生成对称加密密钥和消息认证码 (MAC) 密钥的预备主密码随机数。客户端使用服务器公钥加密预备主密码随机数。

注：公钥由服务器在数字证书或 **ServerKeyExchange** 消息中发送。

9. 如果客户端已将数字证书发送到服务器，客户端还将发送一条签有客户端私钥的 **CertificateVerify** 消息。通过验证此消息的签名，服务器可以明确验证客户端数字证书的所有关系。
10. 客户端发送一条 **ChangeCipherSpec** 消息说明已加载新参数（加密方法和密钥）。
11. 客户端发送一条 **Finished** 消息，此消息是以新加密方法和密钥加密的第一条消息。
12. 服务器端以 **ChangeCipherSpec** 和 **Finished** 消息进行响应。
13. SSL 握手协议结束，可以开始加密交换应用数据。

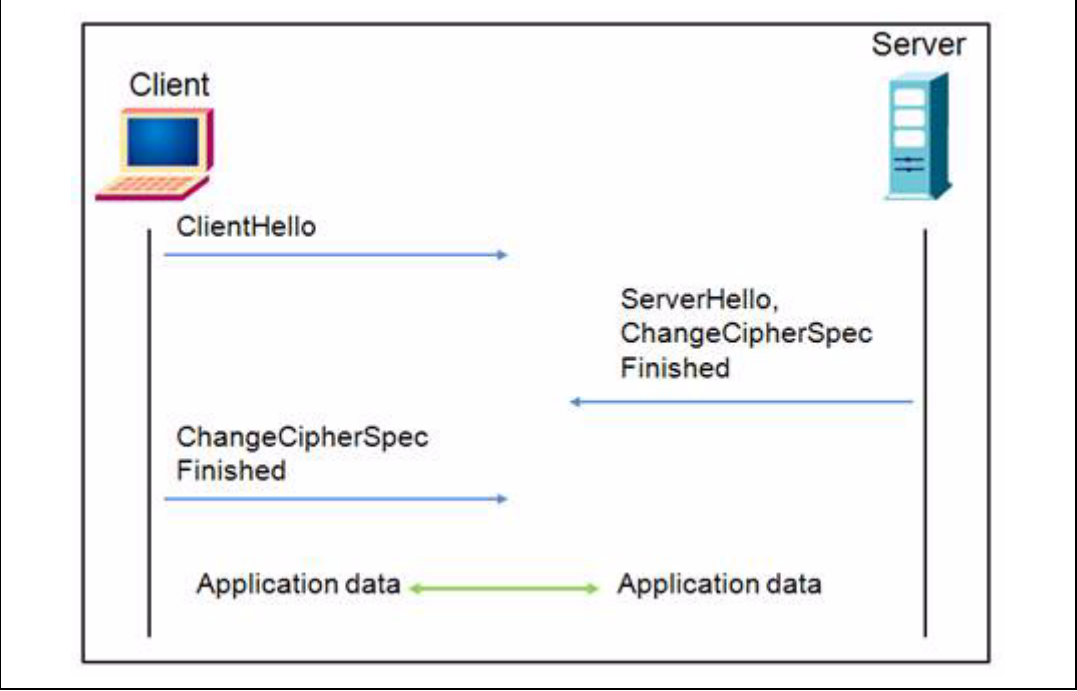
恢复 SSL 会话

当客户端与服务器决定恢复之前的会话或重复现有会话（取代协商新安全参数）时，其消息发送流程如下：

1. 客户端以要恢复会话的会话 ID 发送一条 **ClientHello** 消息。
2. 服务器检查其会话缓存以进行匹配。如果存在匹配的会话 ID，并且服务器希望在指定的会话状态下重新建立连接，则以相同的会话 ID 值发送一条 **ServerHello** 消息。
3. 客户端与服务器都必须发送 **ChangeCipherSpec** 消息并直接进行到 **Finished** 消息。
4. 连接重建过程完成后，客户端和服务器可以开始交换加密的应用数据。

注：如果未找到匹配的会话 ID，服务器将生成新会话 ID，而客户端与服务器将执行整个握手协议 [1]：RFC 2246：TLS 协议 1.0 版。

图 4. 恢复 SSL 会话的握手协议

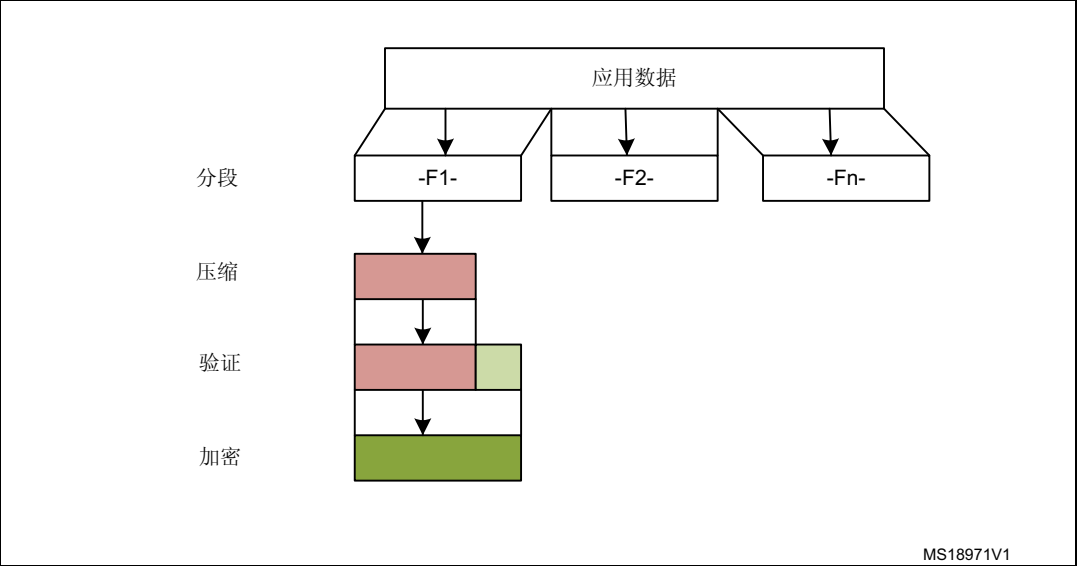


3.3.2 SSL 记录协议

记录协议提取要传输的消息、将数据分段为可处理的块、压缩数据（可选）、应用 MAC、加密和传输处理后的结果。

接收到的数据经过解密、验证、解压和重组后，再传送到更高层级的客户端。

图 5. SSL 记录协议



MS18971V1

3.3.3 SSL 报警协议

SSL 报警协议提示 SSL 会话中存在的问题，这些问题涉及简单的警告（未知证书、废除的证书、过期证书）直至导致 SSL 连结立即终止的致命错误消息。

3.3.4 密码规格变更协议

SSL 密码规格变更协议由一条消息组成，通过该消息指示 SSL 握手协议结束。

注：有关 SSL 协议的详细信息，请参见 [1]：RFC 2246：TLS 协议 1.0 版。

4 PolarSSL 库

PolarSSL 是一个以 C 语言编写的轻量级开源密码库与 SSL/TLS 库。该库中含有实现 SSL/TLS 服务器或客户端应用所需的所有函数。它还包含一系列散列函数和加密算法。

特性：

- SSL 3.0 和 TLS 1.0 客户端/服务器支持
- X.509 数字证书
- 对称加密算法：AES、三重 DES、DES、ARC4 和 Camellia 等
- 散列函数：MD2、MD4、MD5、SHA-1、SHA-256、SHA-384、SHA-512
- 消息验证码：HMAC MD2、HMAC MD4、HMAC MD5、HMAC SHA-1
- 软件随机数发生器：HAVEGE
- 公钥密码技术：RSA 与 Diffie-Hellman (DHM) 密钥交换

PolarSSL 库的源代码可从此链接下载：http://polarssl.org/download_overview

4.1 许可证

用户使用 PolarSSL 需符合双许可模型的要求。即 PolarSSL 用户须取得开源 GPL 第二版许可证以及闭源项目的商用许可证。

有关许可的详细信息，请参见 PolarSSL 许可网页 <http://polarssl.org/licensing>。

5 STM32F417xx 硬件密码技术

如第 4 节所述，PolarSSL 库含有一系列对称加密算法（AES 128/192/256、三重 DES）、散列函数（MD5、SHA-1）和一个软件随机数发生器 (HAVEGE)。要实现 SSL/TLS 应用，需要使用所有这些函数和算法。

为使 CPU 从处理加密/解密、散列和 RNG（随机数发生器）的任务中解脱出来，所有这些函数和算法可通过 STM32F417xx 内置的硬件加速 AES 128/192/256、三重 DES、MD5、SHA-1 和模拟 RNG 功能来实现。

5.1 加密处理器

借助加密处理器，可使用三重 DES 或 AES 算法对数据进行加密或解密。加密处理器完全兼容下列标准：

- 联邦信息处理标准出版物“FIPS PUB 46-3，1999 年 10 月 25 日”规定的加密标准 (DES) 和三重 DES (TDES)。它遵循美国国家标准协会 (ANSI) X9.52 标准。
- 联邦信息处理标准出版物“FIPS PUB 197，2001 年 11 月 26 日”规定的高级加密标准 (AES)

CRYP 处理器可用于电子密码本 (ECB) 模式、密码块链接 (CBC) 模式或计数器 (CTR) 模式（仅在 AES 中存在）下的加密与解密操作。

5.2 随机数发生器

RNG 处理器是一个以连续模拟噪声为基础的随机数发生器，在主机读数时提供一个 32 位的随机数。

5.3 散列处理器

散列处理器完全兼容安全散列算法 (SHA-1)、MD5（消息摘要算法 5）散列算法以及适合多种应用程序的 HMAC（散列消息验证码）算法。对于长达 ($2^{64} - 1$) 的消息，散列处理器将计算消息摘要（SHA-1 算法为 160 位，MD5 算法为 128 位），而 HMAC 算法则通过散列函数提供验证消息的方法。HMAC 算法涉及两次调用 SHA-1 或 MD5 散列函数。

注：有关详细信息，请参见 [2]：RM0090：STM32F405xx、STM32F407xx、STM32F415xx 和 STM32F417xx 微控制器系列参考手册的 CRYP、HASH 和 RNG 小节。

6 演示包说明

演示包含有基于 PolarSSL 库及 LwIP 堆栈运行的两个演示程序：

- **SSL 客户端演示程序：**此演示程序证明 STM32F417xx 器件具有通过 TCP/IP 上的 SSL 连接与服务器交换消息的能力。借助此演示程序，可以通过 SSL 协议将 STM3241G-EVAL 板连接到安全 web 服务器。
- **SSL 服务器演示程序：**SSL 服务器通过 HTTP 与 SSL 协议的组合来实现服务器的加密和安全识别。借助此演示程序，可以使用 SSL 协议从网络浏览器连接到 STM3241G-EVAL 板。

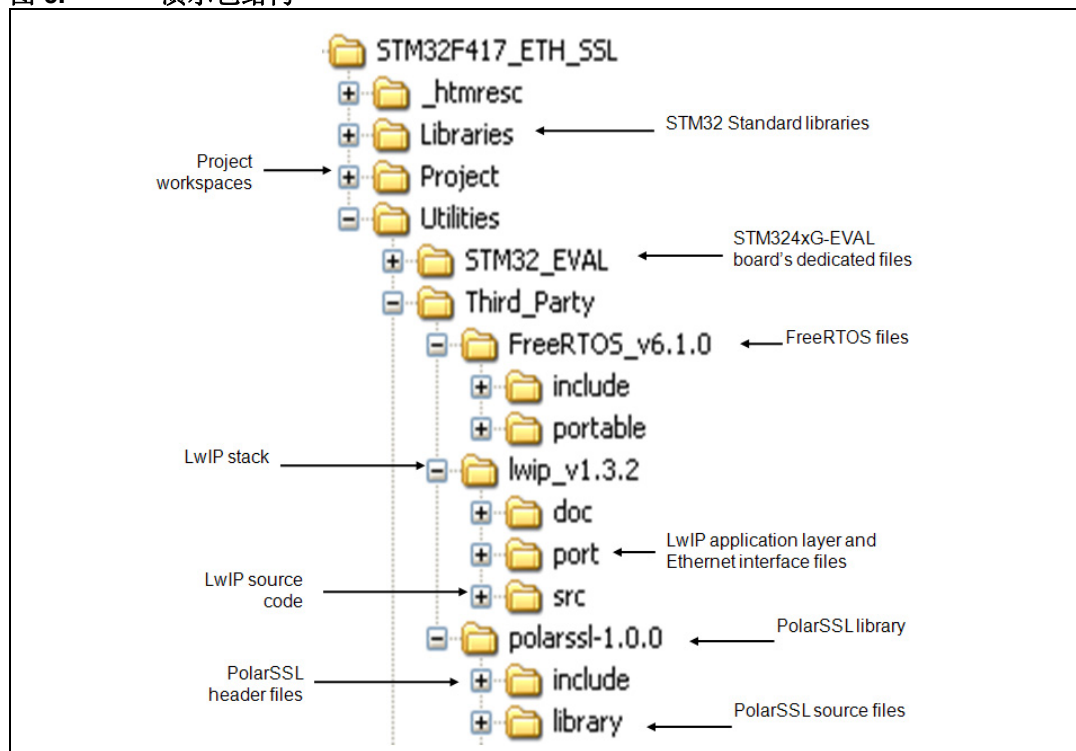
6.1 演示包目录和固件组成

6.1.1 演示包目录

演示包含有下面列出的五个主要文件夹：

- **Libraries：**此文件夹包含用于组成 STM32F4xx 标准外设库核心的所有子目录和文件。
- **Project：**此文件夹包含演示头文件和源文件。
- **FreeRTOS：**此文件夹包含实时内核源码和与调度程序源码相关的所有目录。
 - **include：**包含调度程序头文件。
 - **portable：**包含 STM32 特定代码、编译器的调度程序端口层和内存管理文件。
- **LwIP：**此文件夹包含 TCP/IP 堆栈的头文件和源文件。
- **PolarSSL：**此文件夹包含 PolarSSL-1.0.0 库的头文件和源文件。

图 6. 演示包结构



6.1.2 固件组成

这两个演示程序均基于以下三个软件模块：**LwIP-1.3.2**（免费 TCP/IP 协议栈）、**PolarSSL-1.0.0**（免费 SSL/TLS 库）和 **FreeRTOS-6.1.0**（免费实时内核）。下面对这些模块加以说明。

LwIP 协议栈

LwIP 是瑞典计算机科学研究所 (SICS) 的 Adam Dunkels 开发的免费 TCP/IP 协议栈，需要有 BSD 许可证才能使用。它的源代码可从此链接下载：
<http://download.savannah.gnu.org/releases/lwip/>

LwIP TCP/IP 协议栈支持以下协议：IPv4、IPv6、UDP、TCP、ICMP、IGMP、SNMP、ARP 和 PPP。它不包括应用层的协议，如 HTTP 或 HTTPS。

LwIP 提供三类 API（应用程序编程接口）：

- **原始 API**：适合在无操作系统时使用的原生 API。协议栈的核心部分使用此 API 进行各种协议间的交互。
- **Netconn API**：比原始 API 抽象程度更高的顺序 API。要使用 Netconn API，需要有操作系统支持。所有数据包的处理（输入和输出）都在专用线程（TCP/IP 线程）内完成。应用程序线程使用消息和信号量与此核心线程进行通信。
- **套接字 API**：基于 Berkeley 套接字接口（BSD 套接字）。要使用套接字 API，需要有操作系统支持。

PolarSSL 库

PolarSSL 是一种免费库，用于基于 SSL/TLS 协议实现安全应用。

此库的官方版未给任何微控制器提供任何端口：用户需解决。PolarSSL 库随附 `net.c` 文件，用作 PolarSSL 库与 LwIP 协议栈的接口。要使用 PolarSSL 库，需修改此文件以支持指定的协议栈。

`net.c` 含有确保 PolarSSL 库与 LwIP 协议栈间能够传输帧的函数。其主要函数有：

- `net_recv`，准备好从协议栈读取数据包时应调用此函数。
- `net_send`，准备好向协议栈发送数据包时应调用此函数。

PolarSSL 的连接层使用套接字 API 与 TCP/IP 协议栈进行通信。

注：用于演示程序的 API 是套接字 API，PolarSSL 与 LwIP 协议栈共享此套接字 API。

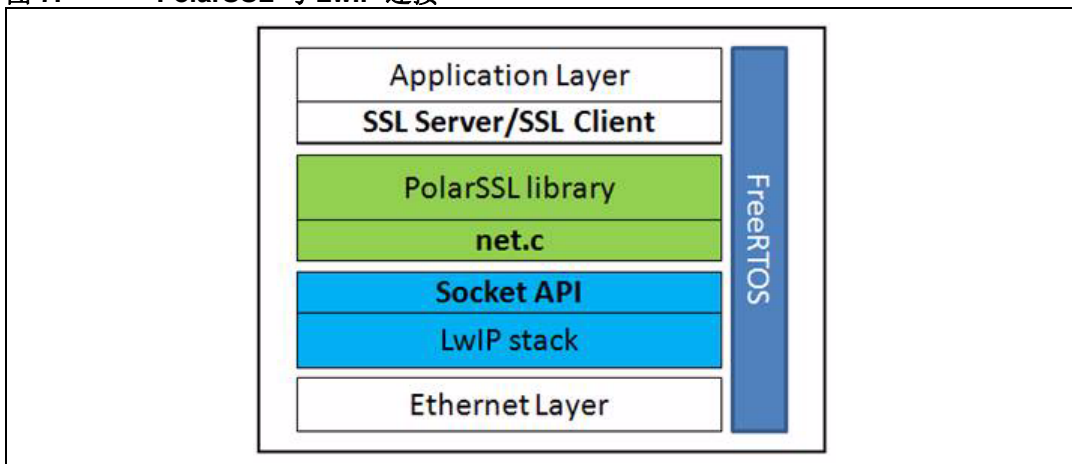
FreeRTOS

FreeRTOS 是用于嵌入式器件的迷你开源实时内核。它的源代码可从此链接下载：
<http://www.freertos.org/>

特性：

- **调度程序操作：**
 - 抢占式：总是运行优先级最高的可用任务。
 - 合作式：只有发生任务阻塞或明确调用 `YIELD()`（用于强制执行上下文切换的宏）任务时，执行上下文切换。
- **进程间通信**：通过消息队列和二进制信号量实现任务间通信。
- **必要时可更改可用的任务优先级数目。**

图 7. PolarSSL 与 LwIP 连接



6.2 演示设置

6.2.1 PHY 接口配置

演示固件用于连接支持 MII 和 RMII 这两种模式的 PHY 接口。要选择想使用的 PHY 接口模式，请转至 main.h 文件并选择以下两个定义之一：

```
#define MII_MODE
#define RMII_MODE
```

如果 main.h 文件中同时定义了 MII_MODE 和 PHY_CLOCK_MCO，则对于 MII 模式，PHY 时钟可来自外部晶振或者由 STM32 通过 MCO 引脚提供。

注： 对于 RMII 模式，必须通过在位于 CN3 下方的 U3 封装中焊接一个 50 MHz 的振荡器（参见 SM7745HEV-50.0M 或相应手册）并断开 JP5 上的跳线来提供 50 MHz 的时钟。这个振荡器并未随电路板一同提供。有关详细信息，请参见 STM3240G-EVAL 评估板用户手册 UM1461。

6.2.2 MAC 地址和 IP 地址设置

默认 MAC 地址固定为 00:00:00:00:00:01。要更改此地址，可以修改 main.h 文件中定义的六字节形式。

在 SSL 客户端演示中，IP 地址设置为静态地址，main.h 文件中定义的默认 IP 地址是 192.168.0.8。

在 SSL 服务器演示中，IP 地址既可设置为静态地址 192.168.0.8，也可设置为由 DHCP 服务器分配的动态地址。

在 main.h 文件中选择 IP 地址的配置模式：

- 取消注释 #define USE_DHCP 可通过 DHCP 配置 IP 地址
- 注释 #define USE_DHCP 可使用静态地址 (192.168.0.8)

注： 如果选择通过 DHCP 配置 IP 地址，但应用程序没有找到它之前在网络中所连接的 DHCP 服务器，则 IP 地址将自动设置为静态地址 (192.168.0.8)。

6.2.3 STM3241G-EVAL 设置

完成设置 PHY 接口模式、MAC 地址和 IP 地址后，需要按下表所示配置 STM3241G-EVAL 评估板。

表 2. STM3241G-EVAL 跳线配置

跳线	MII 模式配置	RMII 模式配置
JP5	1-2: 通过外部晶振提供 25 MHz 时钟 2-3: 通过 PA8 的 MCO 提供 25 MHz 时钟	不适用
JP6	2-3: 使能 MII 接口模式	1-2: 使能 RMII 接口模式
JP8	断开: 选择 MII 接口模式	接通: 选择 RMII 接口模式
JP22	1-2: 使能 RS232	

6.3 如何使用演示程序

6.3.1 SSL 客户端演示程序

此演示程序使用 STM3241G-EVAL 板作为客户端，连接到提供 SSL 握手协议的安全服务器。

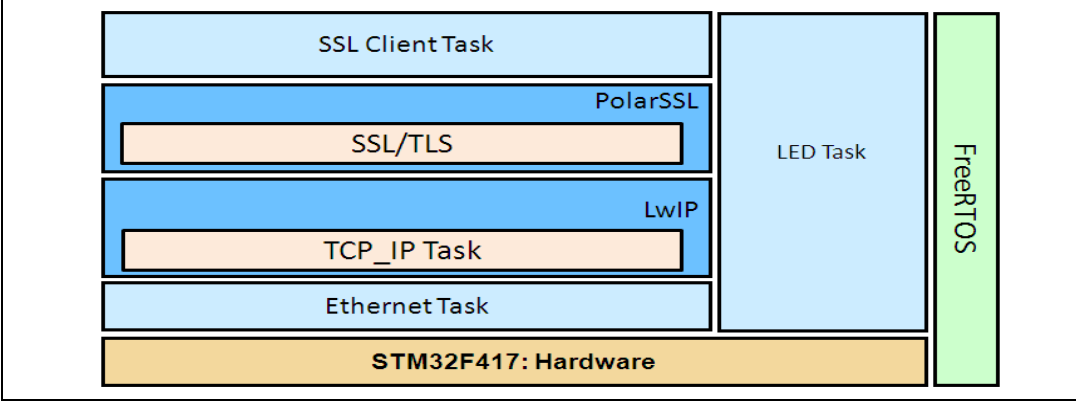
演示程序架构

如 [图 8](#) 所示，SSL 客户端演示程序包括四项任务：

- LED 任务：**LED4 每 200 ms 闪烁一次。
- 以太网任务：**更低层经过设置以检测中断请求帧的接收情况。从而以太网控制器接收到有效帧时便产生中断。在此中断的处理函数中，需创建一个二进制信号量来唤醒以太网任务。此任务将输入帧传送到 TCP/IP 协议栈。
- TCP/IP 任务：**所有数据包的处理（输入和输出）均在此线程内完成。应用程序线程使用消息框和信号量与此线程进行通信。
- SSL 客户端任务：**该任务处理 SSL 握手协议。它连接到 SSL 服务器并执行以下操作：
 - 初始化 SSL 结构（SSL 上下文、SSL 会话、SSL RNG）。
 - 连接到 SSL 服务器。
 - 建立 SSL 会话。
 - 处理 SSL 握手协议。
 - 向服务器写入消息。
 - 读取来自服务器的消息。
 - 通过 USART 发送这些消息。
 - 关闭连接。
 - 清除全部 SSL 结构。



图 8. SSL 客户端演示程序架构



如何使用演示程序

首先，按如下说明连接 STM3241G-EVAL 板：

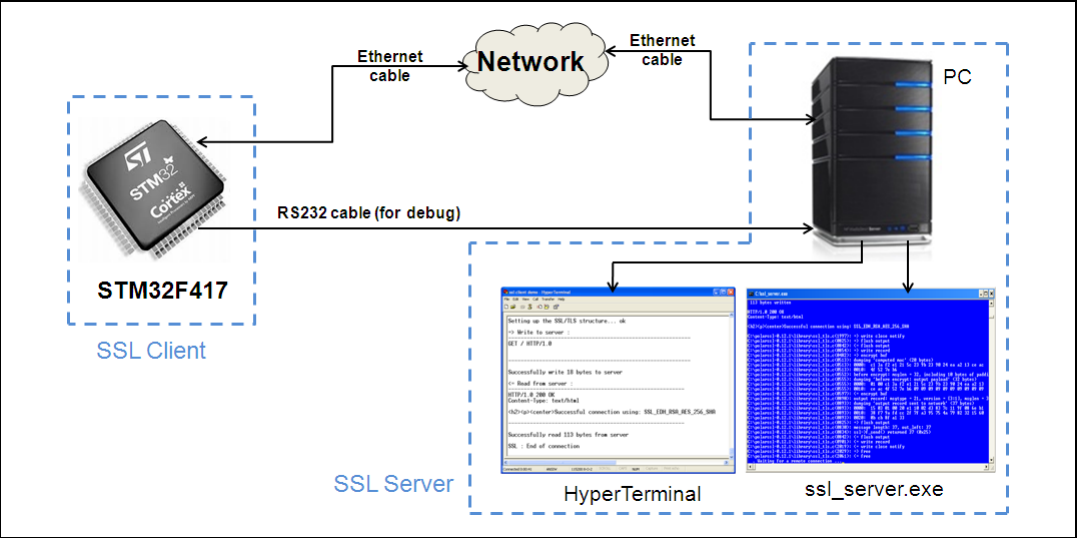
- 以太网链路：连接到远程 PC（通过交叉以太网电缆）或本地网络（通过直通以太网电缆）。
- RS232 链路（用于像超级终端那样的显示调试消息的应用程序）：在 DB9 连接器 CN16 (USART3) 与 PC 串口间，连接零调制解调器母头/母头 RS232 电缆。

要运行 SSL 客户端示例程序，请执行下列操作：

- 将 SSL 客户端代码编译并编程到 STM32F417 Flash 中。
- 在远程 PC 上运行 SSL 服务器应用程序，并运行 Utilities\PC_Software\Server 下的 ssl_server.exe。此应用程序将等待 https 端口 443 的客户端连接。
- 启动 STM3241G-EVAL 板。
- 在 SSL 服务器应用程序窗口和超级终端窗口中监视连接状态。

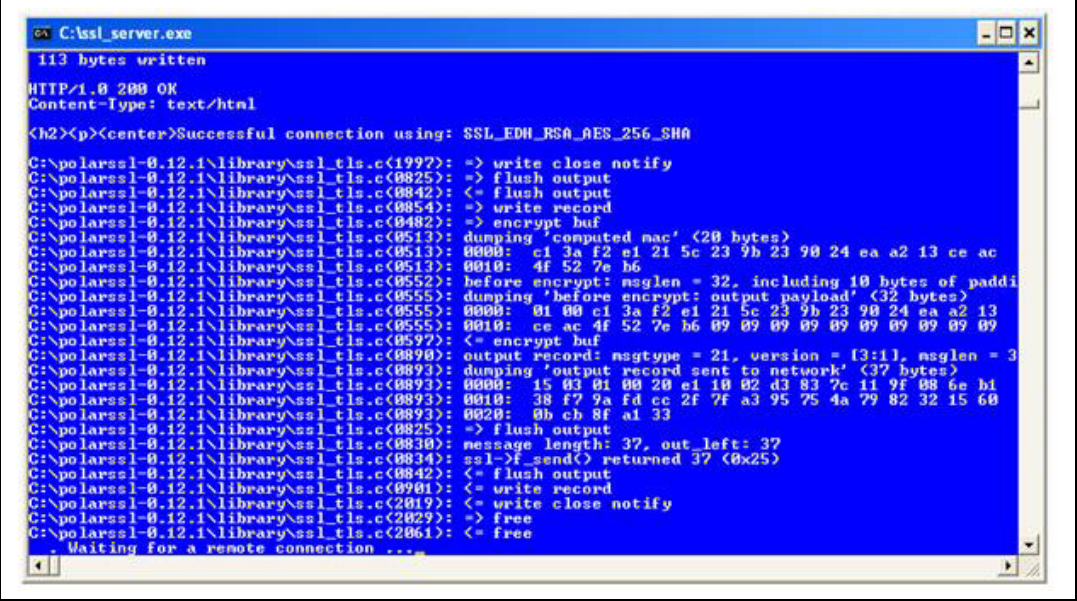
- 注：
- 1 请确保远程 PC 的 IP 地址与 ssl_client.c 文件定义的 IP 地址（`#define SSL_SERVER_NAME "192.168.0.1"`）一致。
 - 2 如果使用了防火墙，必须确保 ssl_server 应用程序能够接受连接请求。如果不能，防火墙将拒绝客户端请求。

图 9. SSL 客户端演示程序



ssl_server.exe 应用程序窗口如图 10 所示。SSL 服务器应用程序窗口显示连接请求的状态，并且显示服务器与客户端间交换的所有消息。

图 10. ssl_server 应用程序窗口



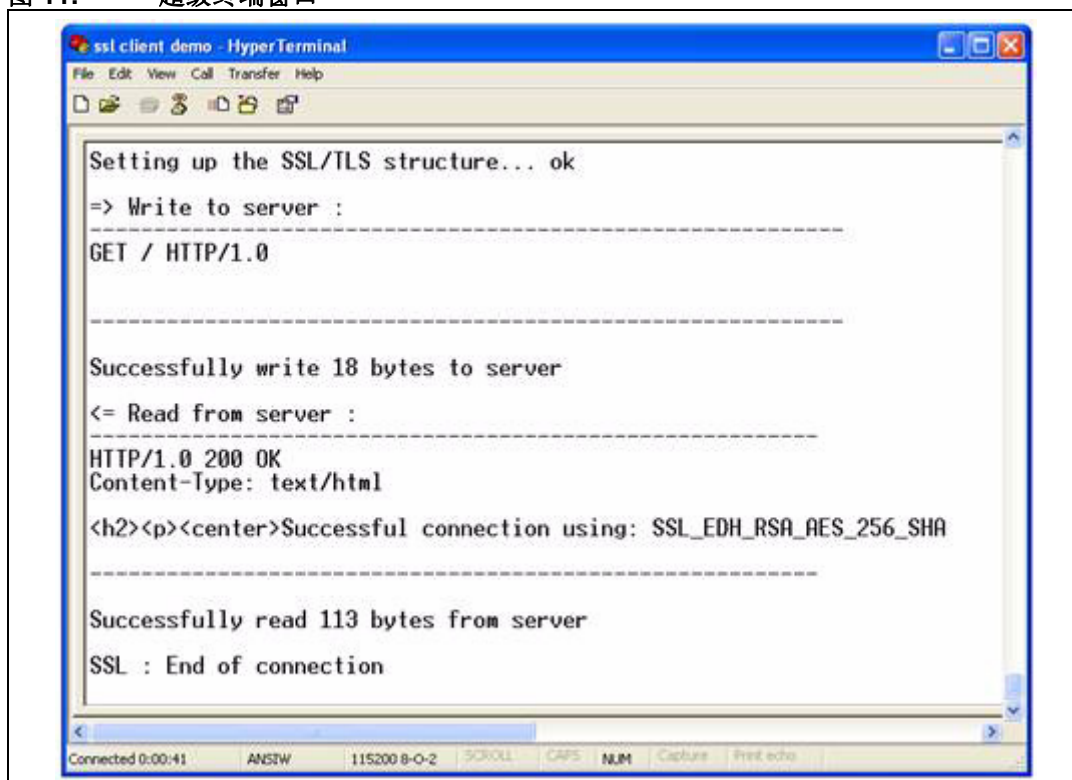
超级终端

图 11: 超级终端窗口显示了在 STM32F417xx 器件上运行的 SSL 客户端应用程序的状态（写消息与读消息）：

- SSL 结构（SSL 上下文、SSL 会话、SSL RNG）的状态
- 客户端对服务器的请求：“GET”
- 接收到的消息包含握手协议的结果：例如 “Successful connection using: SSL_EDH_RSA_AES_256_SHA”（使用 SSL_EDH_RSA_AES_256_SHA 已成功连接）。



图 11. 超级终端窗口



6.3.2 SSL 服务器演示程序

此演示程序将 STM3241G-EVAL 板设置成 SSL 服务器，等待 SSL 客户端发出连接请求。

演示程序架构

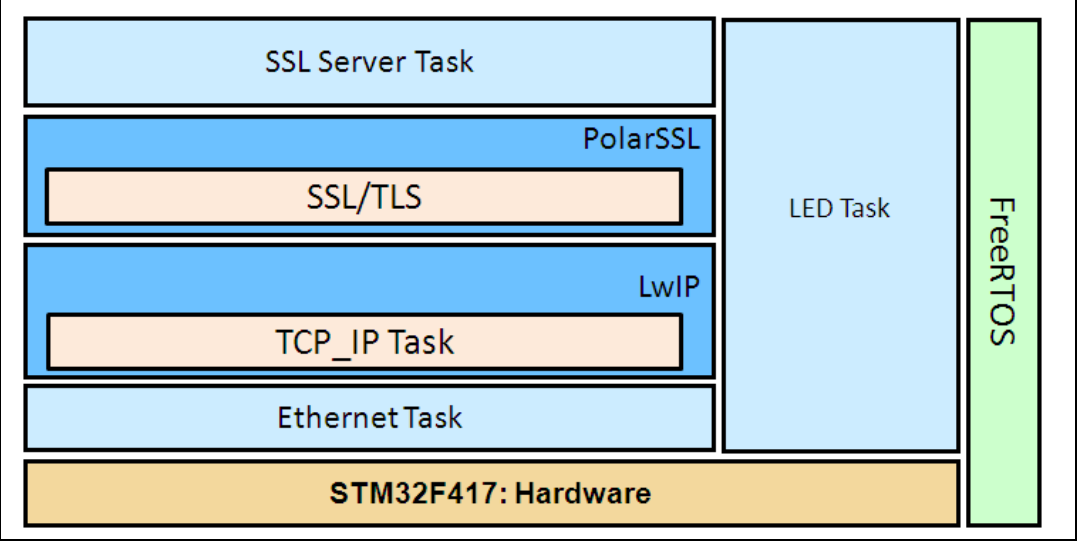
SSL 服务器演示程序包含五项任务：

LED、以太网和 TCP_IP 任务与 SSL 客户端演示程序的任务相同。

SSL 服务器任务：此任务创建 SSL 连接并等待客户端发来安全连接建立请求。建立连接后，客户端发出 Get 请求以加载 html 页面。此页面包含关于此演示中的任务运行信息。SSL 服务器任务也通过 USART 发送连接状态。

DHCP_Client 任务：此任务用于通过 DHCP 配置 IP 地址。为启用 DHCP 客户端，需取消注释 main.h 文件中的 USE_DHCP 定义。

图 12. SSL 服务器演示程序架构



如何使用演示程序

首先，按如下说明连接 STM3241G-EVAL 板：

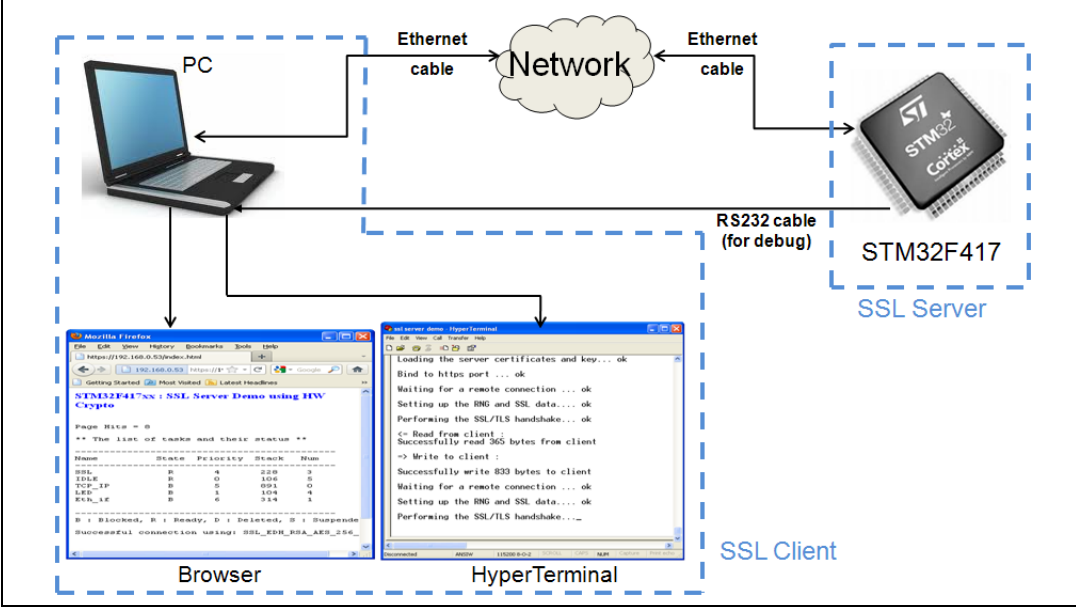
- 以太网链路：连接到远程 PC（通过交叉以太网电缆）或本地网络（通过直通以太网电缆）。
- RS232 链路（用于像超级终端那样的显示调试消息的应用程序）：在 DB9 连接器 CN16 (USART3) 与 PC 串口间，连接零调制解调器母头/母头 RS232 电缆。

要运行 SSL 服务器演示程序：

- 将 SSL 服务器代码编译并编程到 STM32F417 Flash 中。
- 启动 STM3241G-EVAL 板。
- 打开网络浏览器如 Internet Explorer 或 Firefox，然后在浏览器中输入评估板的 IP 地址，如 <https://192.168.0.8>。

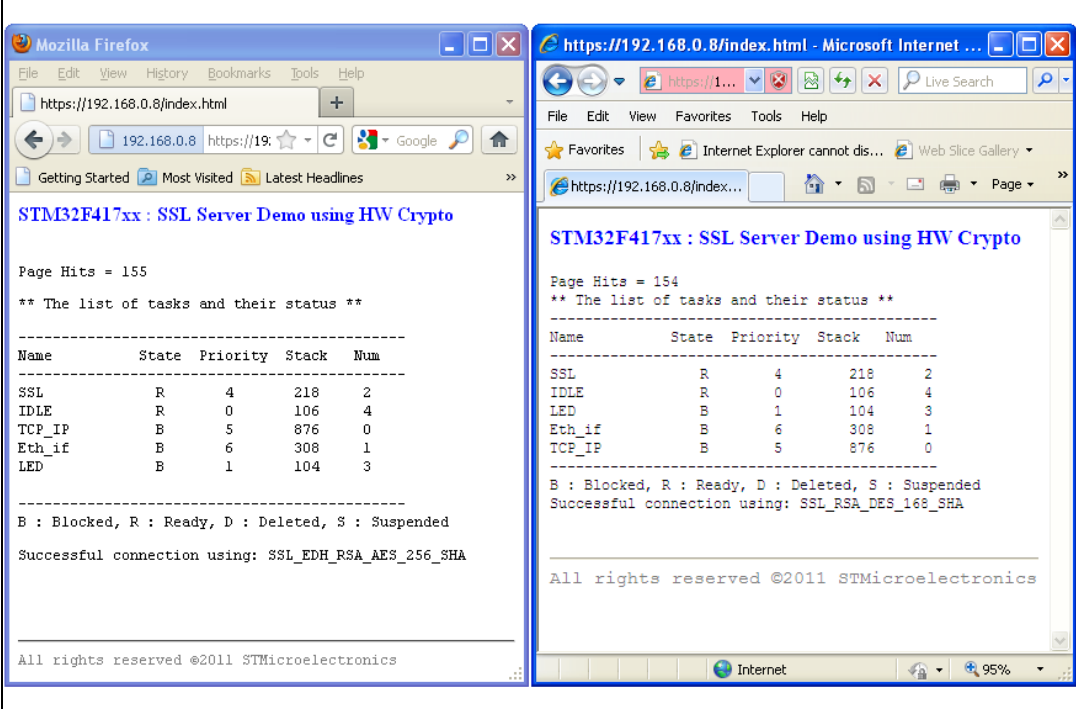
注：如果使用了防火墙，必须确保 HTTPS 端口能够接受连接请求。如果不能，防火墙将拒绝此连接。

图 13. SSL 服务器演示程序



成功连接后，出现一个页面显示运行中的任务及其状态。该页面还会显示页面命中数目和连接中使用的密码组合列表。

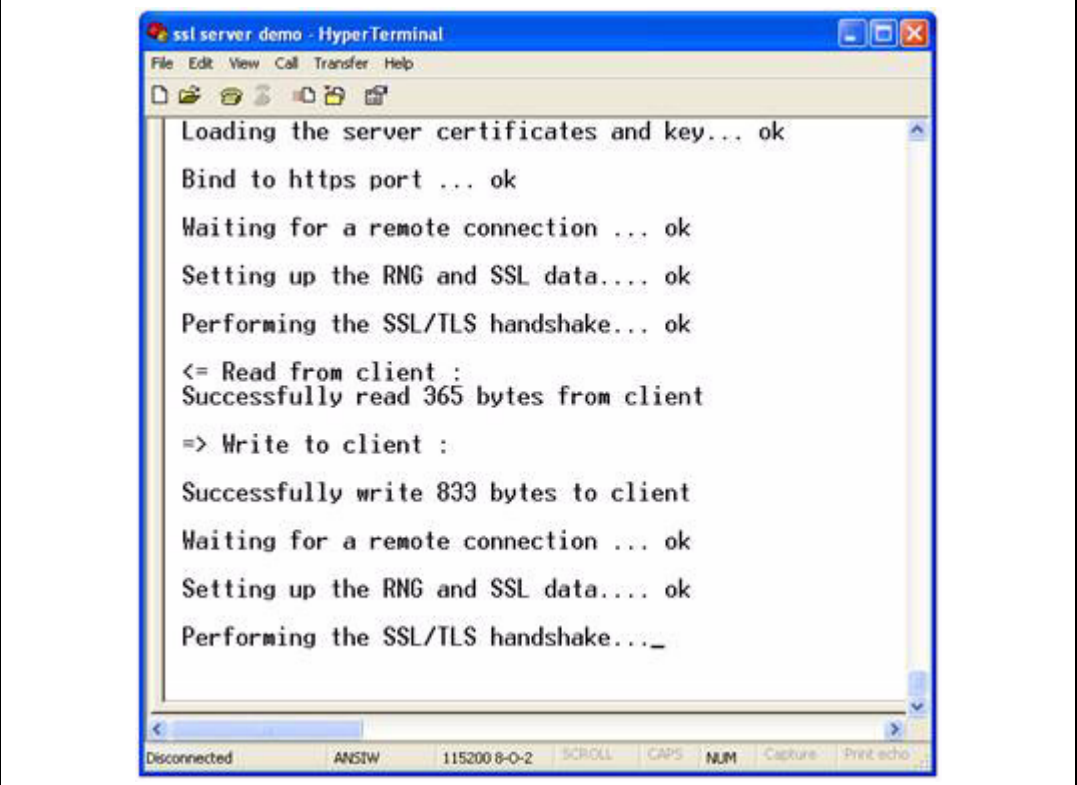
图 14. 成功连接后显示的 HTML 页面



使用超级终端窗口可以监视在 STM32F417xx 器件上运行的 SSL 服务器应用程序的连接状态。此窗口（图 15）显示：

- 连接、SSL 结构和握手协议的状态，
- 客户端发出的请求消息的大小，
- 服务器响应的消息大小（html 页面）。

图 15. 超级终端 SSL 服务器连接状态



注：首次连接到服务器时，用户将从浏览器收到一条有关所提供证书警告消息。如果浏览器无法识别证书颁发机构签发的证书，或者该证书发到了其他网址，则会出现此警告消息。这是因为 SSL 服务器应用程序使用自签名测试证书。跳过此警告仍然安全（请参见第 28 页的附录 A）。

6.4 SSL 演示程序的存储器占用情况

6.4.1 SSL 客户端演示程序

表 3: SSL 客户端演示程序的存储器占用情况列出了客户端演示程序的存储器占用信息，这是基于以下配置得到的计算结果：

- 2 个 1500 字节的缓冲区构成 lwIP 缓冲池。这些参数在 lwipopts.h 文件中通过 PBUF_POOL_SIZE 和 PBUF_POOL_BUFSIZE 定义。
- 2 KB 专用于 lwIP 堆，在 lwipopts.h 文件中通过 MEM_SIZE 定义。
- 5 个 1520 字节专用于以太网驱动程序的缓冲区，在 stm32f4x7_eth_conf.h 文件中定义。

这些值仅供演示目的。若想将当前程序包用于用户自己的应用，则应根据需要对上述参数进行调整。

表 3. SSL 客户端演示程序的存储器占用情况

模块	Flash (字节)		SRAM (字节)
	Ro 代码	Ro 数据	Rw 数据
以太网驱动程序和接口	2340	0	7816
lwIP 存储器管理和 IP 模块	18894	16	7088
PolarSSL	63004	2755	748
FreeRTOS	3050	64	13636
应用程序模块: 主程序和系统初始化	2736	0	2285
STM32F4xx 标准外设库驱动程序	2750	6	16
STM324xG-EVAL 板	1942	4576	44
其它 (栈、堆等)	21610	100	44732
总计	116326	7517	76365

注: 软件使用 IAR EWARM v6.21.3 进行编译, 并对代码长度进行了高度优化。

6.4.2 SSL 服务器演示程序

下表列出服务器演示程序的存储器占用信息, 这是基于以下配置得到的计算结果:

- 4 个 1500 字节的缓冲区构成 lwIP 缓冲池。这些参数在 lwipopts.h 文件中通过 PBUF_POOL_SIZE 和 PBUF_POOL_BUFSIZE 定义。
- 5 KB 专用于 lwIP 堆, 在 lwipopts.h 文件中通过 MEM_SIZE 定义。
- 6 个 1520 字节专用于以太网驱动程序的缓冲区, 在 stm32f4x7_eth_conf.h 文件中定义。

这些值仅供演示目的。若想将当前程序包用于用户自己的应用, 则应根据需要对上述参数进行调整。

表 4. SSL 服务器演示程序存储器占用情况

模块	Flash (字节)		SRAM (字节)
	Ro 代码	Ro 数据	Rw 数据
以太网驱动程序和接口	2340	0	9372
lwIP 存储器管理和 IP 模块	21794	10	13888
PolarSSL	65674	7820	696
FreeRTOS	3310	13	14712
应用程序模块: 主程序和系统初始化	4036	465	2953
STM32F4xx 标准外设库驱动程序	2750	1	16
STM324xG-EVAL 板	1982	4563	44
其它 (栈、堆等)	21640	56	50620
总计	123526	12928	92301

注: 软件使用 IAR EWARM v6.21.3 进行编译, 并对代码长度进行了高度优化。

7 结论

本应用笔记介绍两个实施 PolarSSL 库的 STM32F417xx 演示程序。

第一个演示程序证明 STM32F417xx 器件具有通过 SSL 连接与服务器交换消息的能力。借助该演示程序，可以将 STM3241G-EVAL 板连接到安全 web 服务器。

第二个演示程序通过 HTTP 与 SSL 协议的组合来实现服务器的加密和安全识别。借助该演示程序，可以使用 SSL 协议从网络浏览器连接到 STM3241G-EVAL 板。

8 参考资料

[1]: RFC 2246: TLS 协议 1.0 版

[2]: RM0090: STM32F405xx、STM32F407xx、STM32F415xx 和 STM32F417xx 微控制器系列参考手册

附录 A 补充信息

A.1 流程图

图 16. SSL 客户端任务流程图

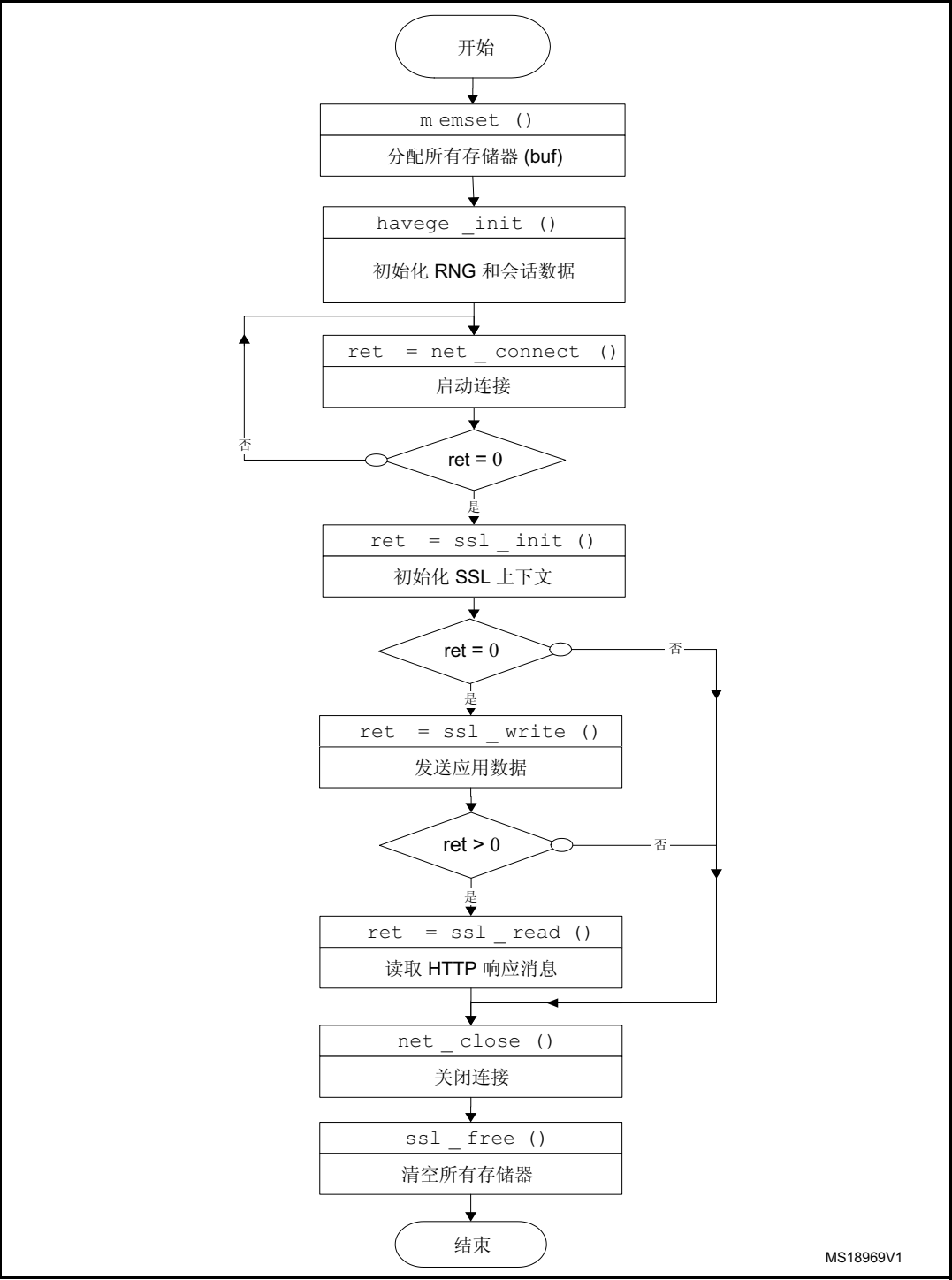
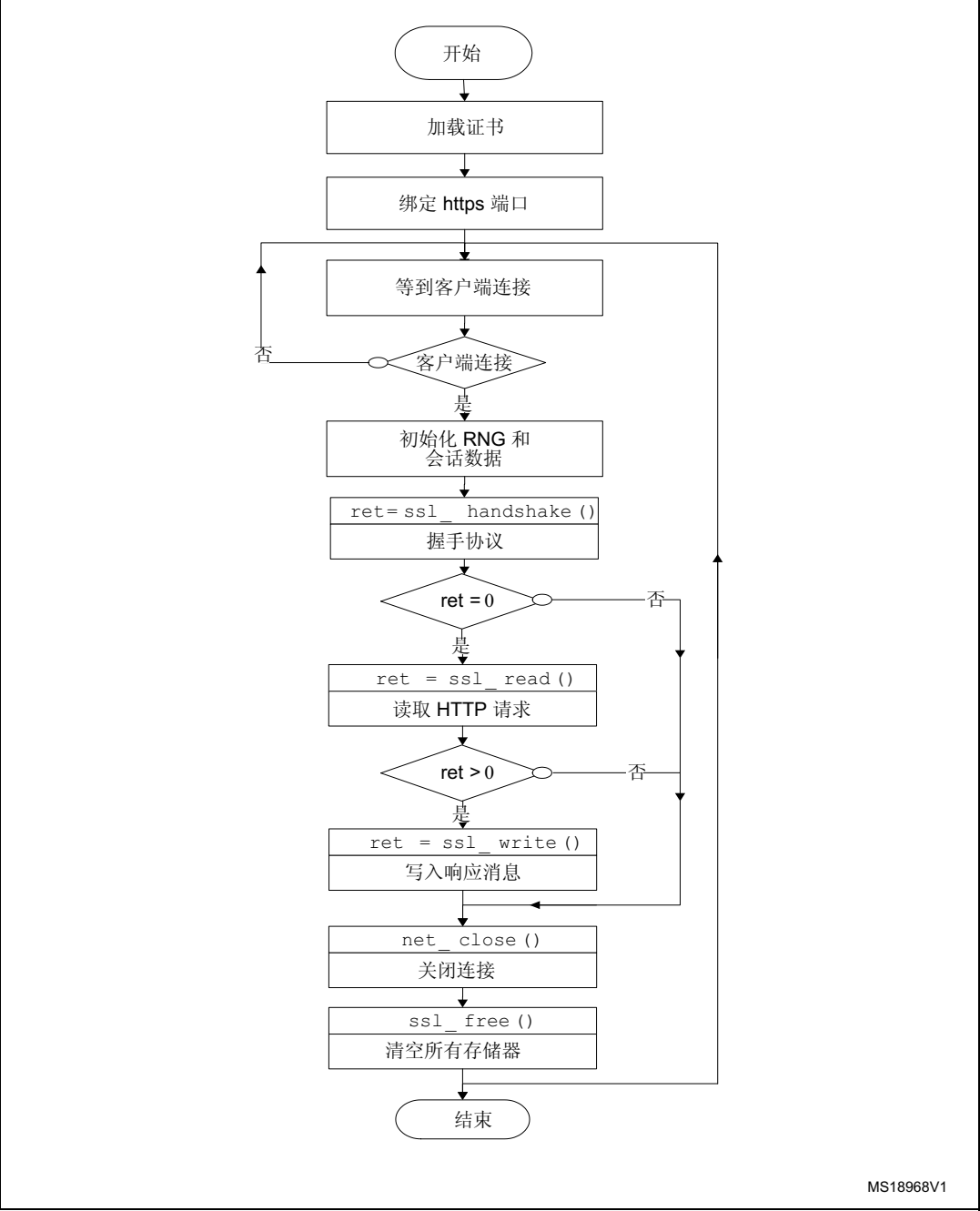


图 17. SSL 服务器任务流程图



MS18968V1

A.2 项目配置

A.2.1 LwIP 配置

表 5 列出 lwIP 软件配置。可修改 lwipopts.h 文件执行 LwIP 配置：

表 5. SSL 服务器演示程序的 lwIP 选项

选项	值	说明
MEM_SIZE	5 * 1024	堆存储器的大小
MEMP_NUM_PBUF	5	已发送但未复制的缓冲区数量
MEMP_NUM_UDP_PCB	4	同时活动的 UDP “连接” 数
MEMP_NUM_TCP_PCB	5	同时活动的 TCP 连接数
MEMP_NUM_TCP_PCB_LISTEN	5	侦听的 TCP 连接数
PBUF_POOL_SIZE	4	数据包缓冲区个数
PBUF_POOL_BUFSIZE	1500	pbuf 池中每个 pbuf 的大小
LWIP_ICMP	1	使能 ICMP 协议
LWIP_DHCP	1	使能 DHCP 协议
LWIP_UDP	1	使能 UDP 协议
LWIP_TCP	1	使能 TCP 协议
TCP_MSS	1460	TCP 最大段长度
TCP_WND	2 * TCP_MSS	TCP 窗口大小：接收缓冲空间（以字节计）
TCP_SND_BUF	2 * TCP_MSS	TCP 发送方缓冲空间

表 6. SSL 客户端演示程序的 lwIP 选项

选项	值	说明
MEM_SIZE	2 * 1024	堆存储器的大小
MEMP_NUM_PBUF	2	已发送但未复制的缓冲区数量
MEMP_NUM_UDP_PCB	2	同时活动的 UDP “连接” 数
MEMP_NUM_TCP_PCB	2	同时活动的 TCP 连接数
MEMP_NUM_TCP_PCB_LISTEN	6	侦听的 TCP 连接数
PBUF_POOL_SIZE	2	数据包缓冲区个数
PBUF_POOL_BUFSIZE	1500	pbuf 池中每个 pbuf 的大小
LWIP_ICMP	1	使能 ICMP 协议
LWIP_DHCP	1	使能 DHCP 协议
LWIP_UDP	1	使能 UDP 协议

表 6. SSL 客户端演示程序的 lwIP 选项 (续)

选项	值	说明
LWIP_TCP	1	使能 TCP 协议
TCP_MSS	1460	TCP 最大段长度
TCP_WND	2 * TCP_MSS	TCP 窗口大小: 接收缓冲空间 (以字节计)
TCP_SND_BUF	2 * TCP_MSS	TCP 发送方缓冲空间

A.2.2 PolarSSL 配置

PolarSSL 配置可通过修改 config.h 文件来完成; 可通过注释或取消注释具体的行来使能 / 禁止软件组成元素。

为了减小存储器大小, 应注释未使用的模块。

表 7. PolarSSL 选项: SSL 服务器演示程序的 config.h 文件

选项	说明
POLARSSL_DEBUG_MSG	启用所有 SSL/TLS 调试消息。
POLARSSL_AES_C	启用下列密码组合: SSL_RSA_AES_128_SHA SSL_RSA_AES_256_SHA SSL_EDH_RSA_AES_256_SHA
POLARSSL_ARC4_C	启用下列密码组合: SSL_RSA_RC4_128_MD5 SSL_RSA_RC4_128_SHA
POLARSSL_CAMELLIA_C	启用下列密码组合: SSL_RSA_CAMELLIA_128_SHA SSL_RSA_CAMELLIA_256_SHA SSL_EDH_RSA_CAMELLIA_256_SHA
POLARSSL_DES_C	启用下列密码组合: SSL_RSA_DES_168_SHA SSL_EDH_RSA_DES_168_SHA
POLARSSL_DHM_C	启用下列密码组合: SSL_EDH_RSA_DES_168_SHA SSL_EDH_RSA_AES_256_SHA SSL_EDH_RSA_CAMELLIA_256_SHA
POLARSSL_SSL_SRV_C	启用 SSL/TLS 服务器模式

表 8. PolarSSL 选项: SSL 客户端演示程序的 config.h 文件

选项	说明
POLARSSL_DEBUG_MSG	启用所有 SSL/TLS 调试消息。
POLARSSL_AES_C	启用下列密码组合: SSL_RSA_AES_128_SHA SSL_RSA_AES_256_SHA SSL_EDH_RSA_AES_256_SHA
POLARSSL_ARC4_C	启用下列密码组合: SSL_RSA_RC4_128_MD5 SSL_RSA_RC4_128_SHA
POLARSSL_CAMELLIA_C	启用下列密码组合: SSL_RSA_CAMELLIA_128_SHA SSL_RSA_CAMELLIA_256_SHA SSL_EDH_RSA_CAMELLIA_256_SHA
POLARSSL_DES_C	启用下列密码组合: SSL_RSA_DES_168_SHA SSL_EDH_RSA_DES_168_SHA
POLARSSL_DHM_C	启用下列密码组合: SSL_EDH_RSA_DES_168_SHA SSL_EDH_RSA_AES_256_SHA SSL_EDH_RSA_CAMELLIA_256_SHA
POLARSSL_SSL_CLI_C	启用 SSL/TLS 客户端模式

A.2.3 FreeRTOS 配置

可修改 FreeRTOSconfig.h 文件完成 FreeRTOS 配置。

表 9. SSL 客户端演示程序的 FreeRTOS 配置

参数	值	说明
configMAX_PRIORITIES	7	优先级最大值
configMAX_TASK_NAME_LEN	16	任务名称的最大长度
configMINIMAL_STACK_SIZE	128	分配给 Idle 任务的栈大小
configTOTAL_HEAP_SIZE	13 * 1024	总 FreeRTOS 堆大小

表 10. SSL 服务器演示程序的 FreeRTOS 配置

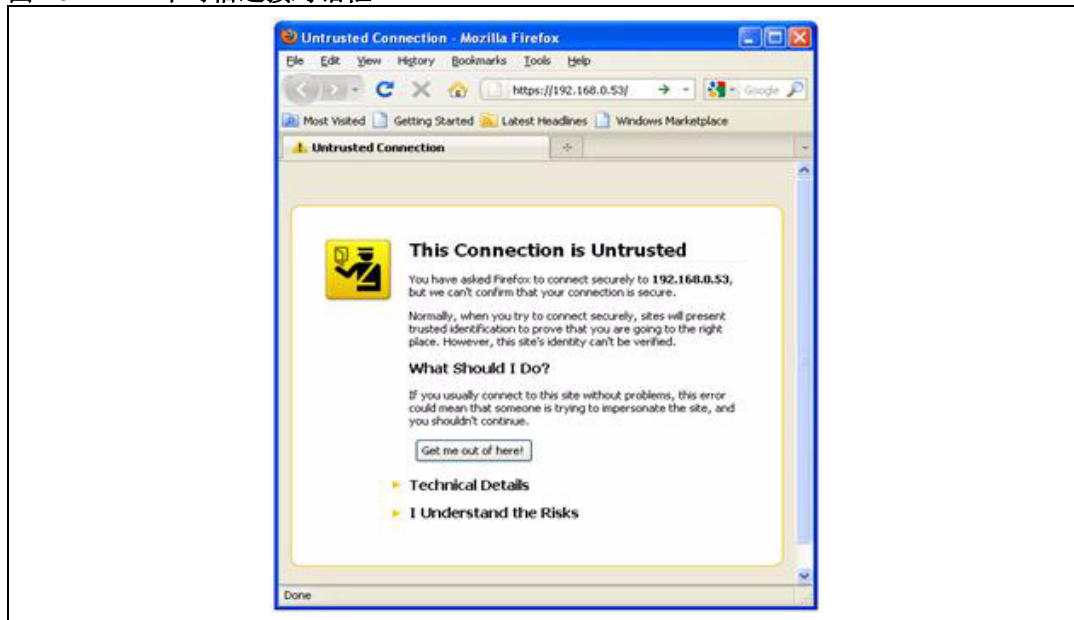
参数	值	说明
configMAX_PRIORITIES	7	优先级最大值
configMAX_TASK_NAME_LEN	16	任务名称的最大长度
configMINIMAL_STACK_SIZE	128	分配给 Idle 任务的栈大小
configTOTAL_HEAP_SIZE	14 * 1024	总 FreeRTOS 堆大小

A.3 使用 Mozilla 7.0.1 运行 SSL 服务器演示程序

下面是使用 Mozilla 7.0.1 运行 SSL 服务器演示程序的步骤。

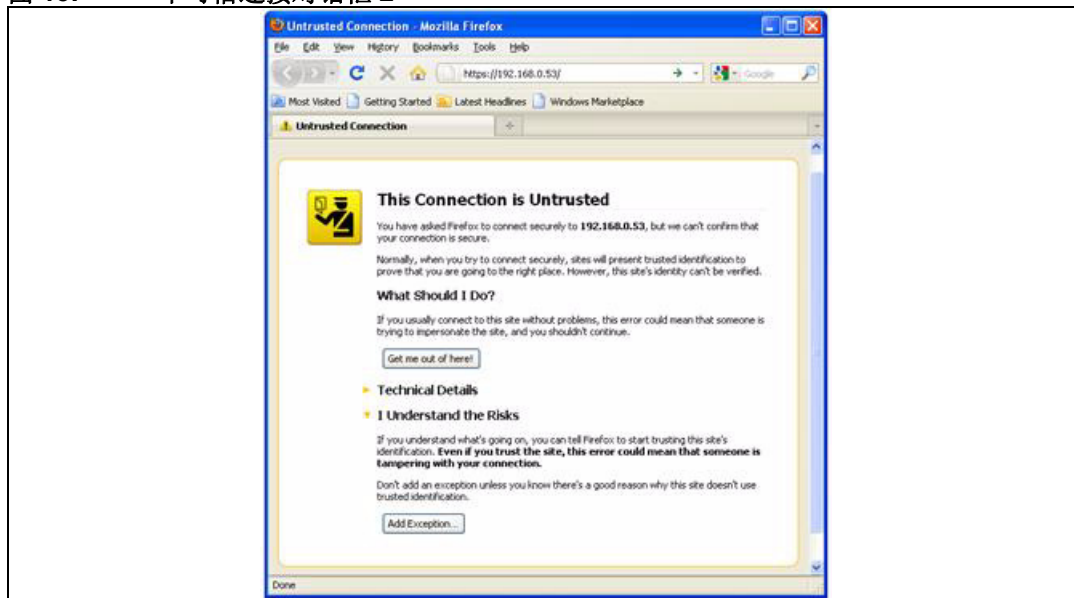
1. 打开浏览器并输入 URL，如 <https://192.168.0.53/>。浏览器显示一条如 [图 18](#) 所示的警告消息。

图 18. 不可信连接对话框 1



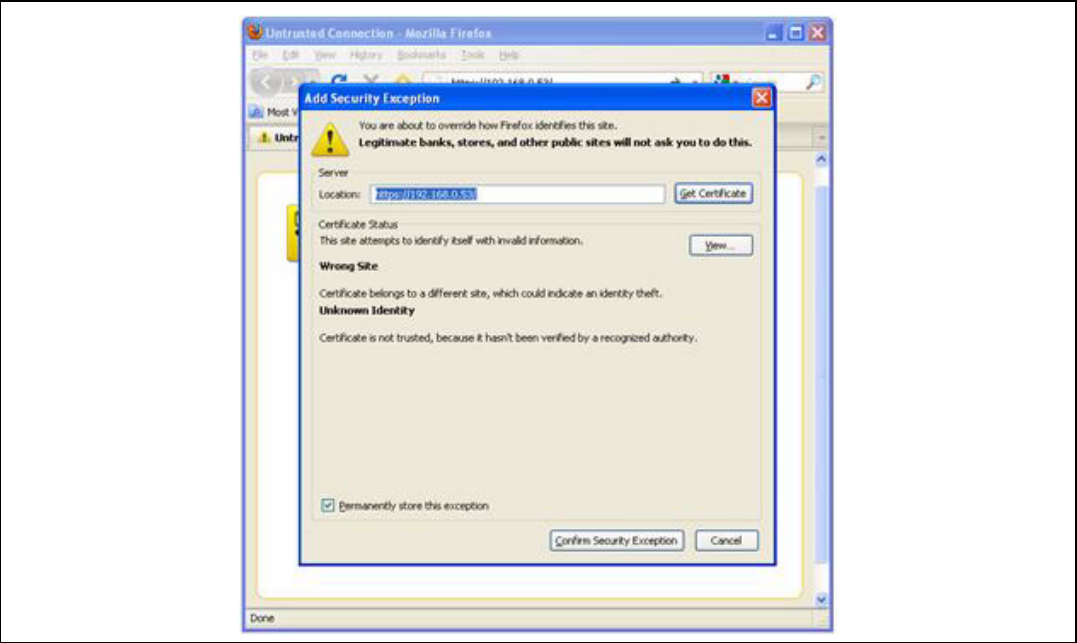
2. 选择 **I Understand the Risks**（我了解此风险），将显示如 [图 19](#) 所示的另一条消息。

图 19. 不可信连接对话框 2



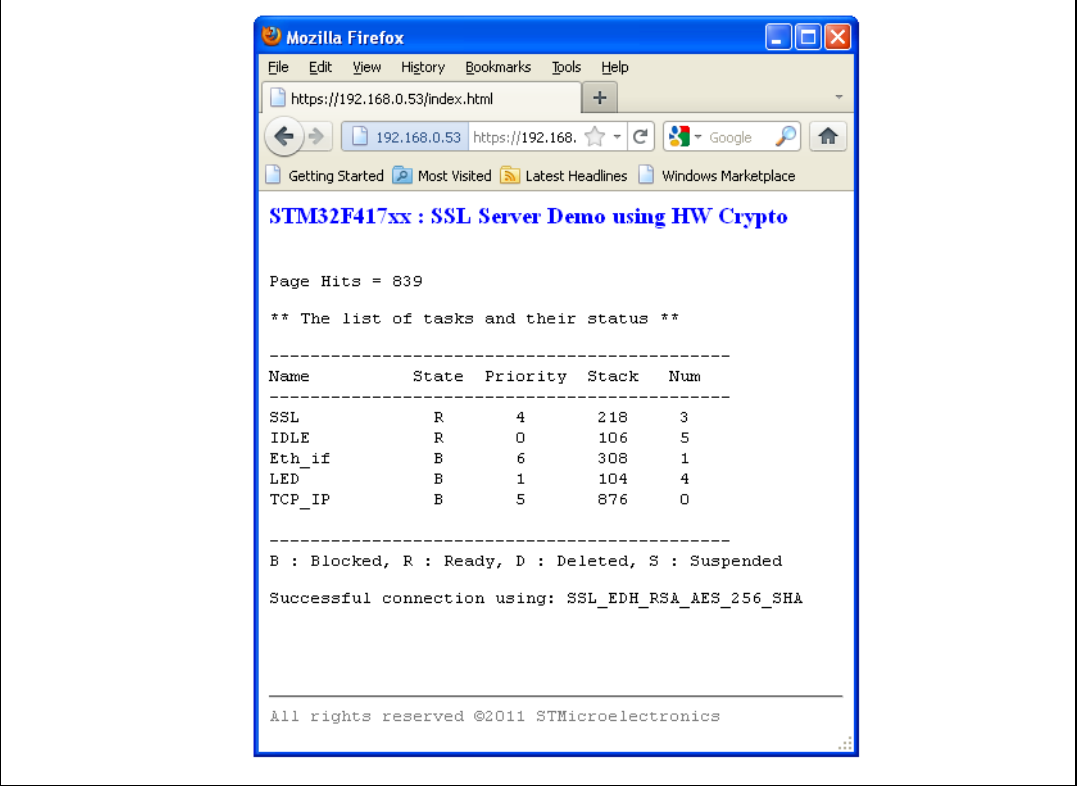
3. 单击 **Add Exception**（添加例外）。浏览器下载 (CA) 证书颁发机构颁发的证书。

图 20. Add Security Exception（添加安全例外）对话框



4. 单击 **Confirm Security Exception**（确认安全例外）启动安全连接，如 图 20 所示。成功连接后，出现 图 21：任务状态页面，显示运行中的任务及其状态。该窗口还显示页面命中数目。

图 21. 任务状态页面

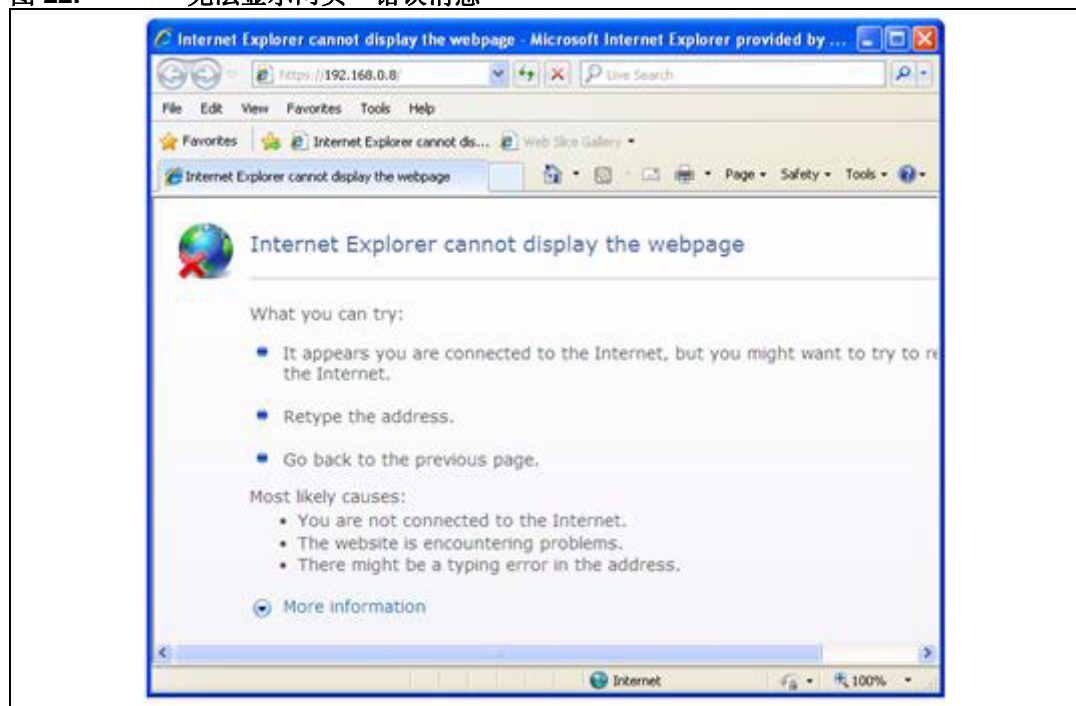


A.4 使用 IE8 运行 SSL 服务器演示程序

下面是使用 IE8 (Windows Internet Explorer 8) 运行 SSL 服务器演示程序的步骤。

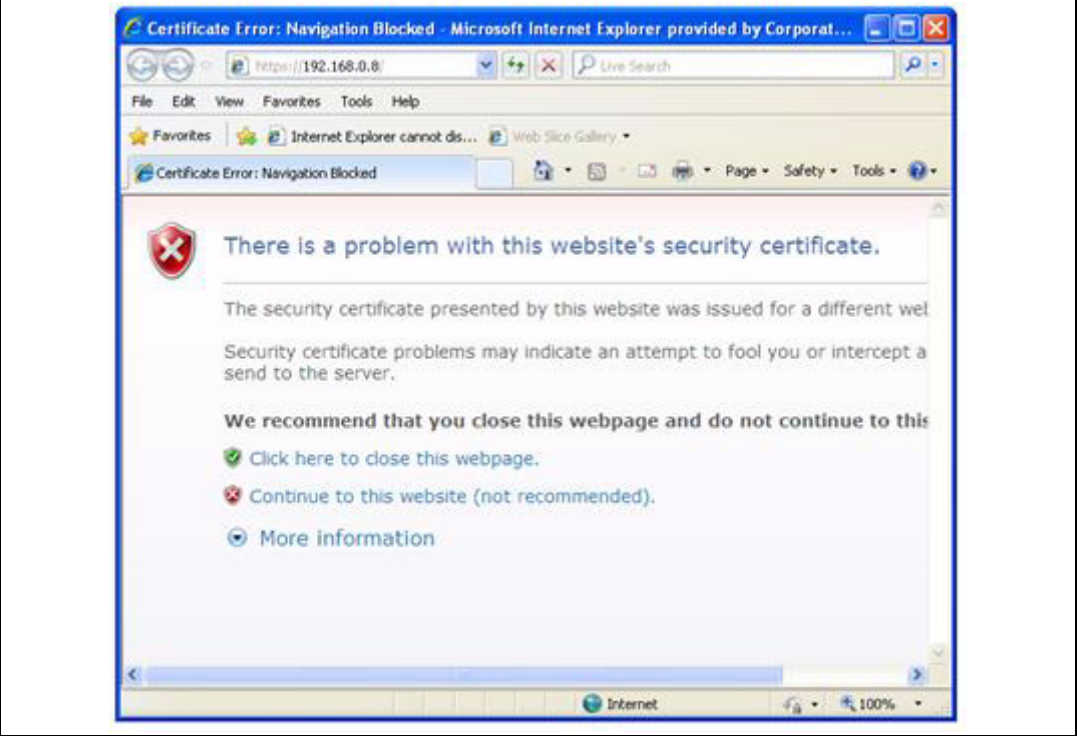
1. 打开浏览器并输入 URL，如 `https://192.168.0.8`。浏览器显示一条警告消息。

图 22. “无法显示网页” 错误消息



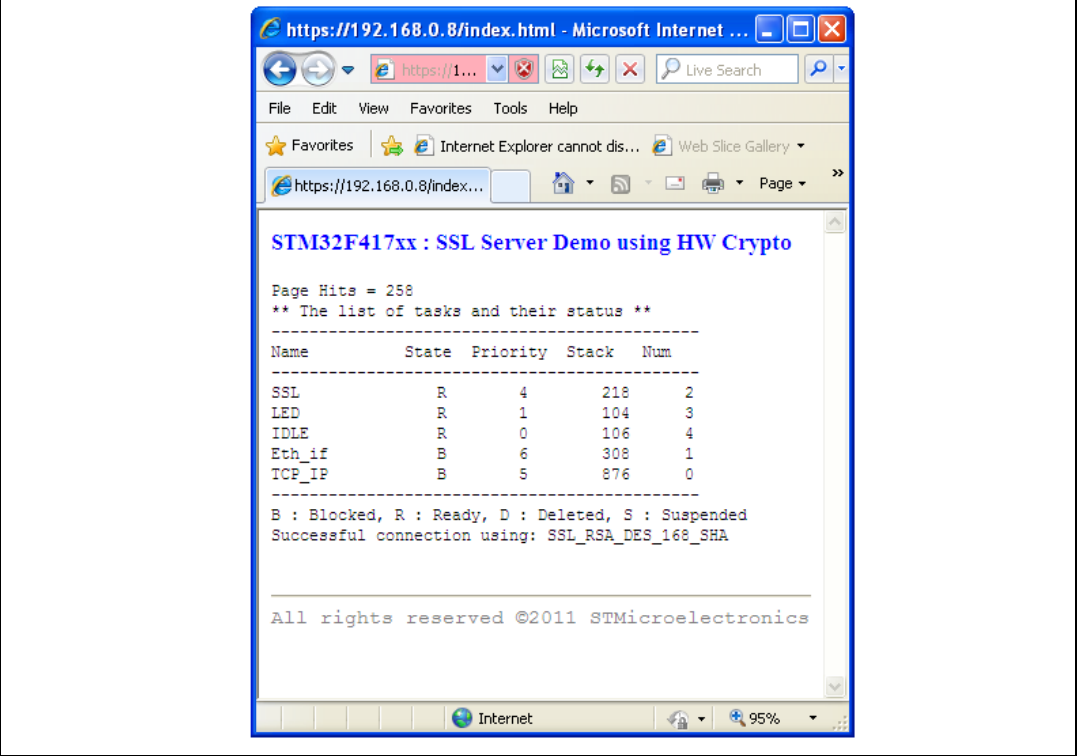
2. 刷新当前页面。

图 23. 证书错误消息



3. 选择 **Continue to this website (not recommended)** [继续访问该网站（不推荐）]: 如果连接成功, 应可看到下列网页, 如图 24: 任务状态页面所示。

图 24. 任务状态页面



9 版本历史

表 11. 文档版本历史

日期	版本	变更
2011 年 10 月 31 日	1	初始版本

请仔细阅读：

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文档中信息的提供仅与ST产品有关。意法半导体公司及其子公司（“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有ST产品均根据ST的销售条款出售。

买方自行负责对本文所述ST产品和服务的选择和使用，ST概不承担与选择或使用本文所述ST产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为ST授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在ST的销售条款中另有说明，否则，ST对ST产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且/或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此而导致的任何风险，意法半导体的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML或JAN正式认证产品适用于航天应用。

经销的ST产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致ST针对本文所述ST产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大ST的任何责任。

ST和ST徽标是ST在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2013 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

www.st.com