|  |
| --- |
| OpenCPU |
| SDK SSLconfig API说明 |
| 版本1.1日期2025-02-25 |

版权声明

版权所有 © 深圳市有方科技股份有限公司2025。深圳市有方科技股份有限公司保留所有权利。

未经深圳市有方科技股份有限公司书面同意，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

是深圳市有方科技股份有限公司所有商标。

本文档中出现的其他商标，由商标所有者所有。

说明

本文档的使用对象为系统工程师，开发工程师及测试工程师。

本设计指南为客户产品设计提供支持，客户须按照本文中的规范和参数进行产品设计和调试。如因客户操作不当造成的人身伤害和财产损失，有方概不承担责任。

由于产品版本升级或其它原因，本文档内容会在不预先通知的情况下进行必要的更新。

除非另有约定，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市有方科技股份有限公司为客户提供全方位的技术支持，任何垂询请直接联系您的客户经理或发送邮件至以下邮箱：

Sales@neoway.com

Support@neoway.com

公司网址：http://www.neoway.com

目录

[关于本文档 iv](#_Toc191395950)

[范围 iv](#_Toc191395951)

[读者对象 iv](#_Toc191395952)

[修订记录 iv](#_Toc191395953)

[符号约定 iv](#_Toc191395954)

[1 范围 5](#_Toc191395955)

[2 数据结构 6](#_Toc191395956)

[2.1 nwy\_ssl\_version\_e 6](#_Toc191395957)

[2.2 nwy\_ssl\_auth\_mode\_e 6](#_Toc191395958)

[2.3 nwy\_ssl\_cert\_t 6](#_Toc191395959)

[2.4 nwy\_ssl\_conf\_t 6](#_Toc191395960)

[2.5 nwy\_ssl\_version\_info\_t 7](#_Toc191395961)

[3 函数 9](#_Toc191395962)

关于本文档

范围

本文档对应产品为N706B系列（RTOS系统）。

读者对象

本文档的使用对象为系统工程师，开发工程师及测试工程师。

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 变更 | 作者 |
| 1.0 | 2024-12 | 初始版本 | Gao Zheng |
| 1.1 | 2025-02 | 更新目录，新增版本查询功能 | Gao Zheng |

符号约定

|  |  |
| --- | --- |
| 符号 | 含义 |
|  | 危险或警告，用户必须遵从的规则，否则会造成模块或客户设备不可逆的故障损坏，甚至可能造成人员身体伤害。 |
|  | 注意，警示用户使用模块时应该特别注意的地方，如不遵从，模块或客户设备可能出现故障。 |
|  | 说明或提示，提供模块使用的意见或建议。 |

# 范围

本文档介绍了SSL配置文件，nwy\_ssl\_config.h。

# 数据结构

## nwy\_ssl\_version\_e

typedef enum{

NWY\_VERSION\_SSL\_V3\_E, //SSL3.0

NWY\_VERSION\_TLS\_V1\_0\_E, //SSL1.0

NWY\_VERSION\_TLS\_V1\_1\_E, //SSL1.1

NWY\_VERSION\_TLS\_V1\_2\_E, //SSL1.2

NWY\_VERSION\_MAX\_E

}nwy\_ssl\_version\_e;

## nwy\_ssl\_auth\_mode\_e

typedef enum{

NWY\_SSL\_AUTH\_NONE\_E = 0, //No authentication

NWY\_SSL\_AUTH\_ONE\_WAY\_E, /\*Manage server authentication

NWY\_SSL\_AUTH\_MUTUAL\_E, //Manage server and client authentication if requested by the remote server\*/

NWY\_SSL\_AUTH\_MAX\_E

}nwy\_ssl\_auth\_mode\_e;

## nwy\_ssl\_cert\_t

typedef struct

{

char \*cert\_data;

int cert\_len;

}nwy\_ssl\_cert\_t;

## nwy\_ssl\_conf\_t

typedef struct

{

nwy\_ssl\_version\_type\_e ssl\_version;

nwy\_ssl\_auth\_mode\_e authmode;

nwy\_ssl\_cert\_t cacert;

nwy\_ssl\_cert\_t clientcert;

nwy\_ssl\_cert\_t clientkey;

int sni\_name\_size; /\*\*< Length of the SNI server name. \*/

char \*sni\_name; /\*\*< Server name for SNI. \*/

}nwy\_ssl\_conf\_t;

Server Name Indication（SNI）：

SNI (Server Name Indication)是用来改善服务器与客户端 SSL (Secure Socket Layer)和 TLS (Transport Layer Security) 的一个扩展。主要解决一台服务器只能使用一个证书(一个域名)的缺点，随着服务器对虚拟主机的支持，一个服务器上可以为多个域名提供服务，因此SNI必须得到支持才能满足需求。

SSL存在验证证书的时候，有这么一个判断：比较“浏览器输入的地址”和“获取的证书的名称”，如果一样，那么接着验证，如果不一样，那么认为证书是不可信的。

假设有公司的域名存在2个：[www.123.com](http://www.123.com/)和[www.567.com](http://www.567.com/)，它们对应的ip都是222.12.34.56，假设服务证书的名字是 ”[www.123.com](http://www.123.com/)” ，那么从 ”[www.567.com](http://www.567.com/)” 访问过来的用户无法信任服务器证书，总之，总有一个域名访问会导致客户端无法信任服务器。为了解决这个问题，客户端在client hello中带上server name 拓展（如果使用ip地址进行访问，那么就不会有server name拓展），它会捎带上域名地址，服务器解析到server name后，就会根据server name中的域名，选择合适的证书。

## nwy\_ssl\_version\_info\_t

typedef struct {

char version\_string[64];// 库版本信息

int tls\_version;// 支持的 TLS 版本信息

int dtls\_version; //支持 dtls 协议版本信息

} nwy\_ssl\_version\_info\_t;

**参数nwy\_ssl\_version\_info\_t字段说明**

（1）version\_string[64]

存储 MbedTLS 的版本字符串，例如 "MbedTLS 3.3.0"。

（2）tls\_version

存储支持的 TLS 协议版本信息。

版本信息以位掩码的形式存储。每个位代表一个特定的 TLS 协议版本。

例如：

第 0 位（1 << 0）表示是否支持 SSL 3.0。

第 1 位（1 << 1）表示是否支持 TLS 1.0。

第 2 位（1 << 2）表示是否支持 TLS 1.1。

第 3 位（1 << 3）表示是否支持 TLS 1.2。

通过位运算（如 & 和 <<）可以检查是否支持某个特定版本。

（3）dtls\_version

存储支持的 DTLS 协议版本信息。

同样以位掩码的形式存储。每个位代表一个特定的 DTLS 协议版本。

例如：

第 0 位（1 << 0）表示是否支持 DTLS 1.0。

第 1 位（1 << 1）表示是否支持 DTLS 1.2。

通过位运算可以检查是否支持某个特定版本。

# 3 函数

* + - 1. nwy\_ssl\_version\_info\_get

|  |  |
| --- | --- |
| 格式 | int nwy\_ssl\_version\_info\_get(nwy\_ssl\_version\_info\_t \*info); |
| 描述 | 用于获取 Mbed TLS 的版本信息，包括版本字符串和支持的 TLS/DTLS 版本 |
| 参数说明 | **输入参数**：  nwy\_ssl\_version\_info\_t \*info：  指向 nwy\_ssl\_version\_info\_t 结构体的指针，用于存储版本信息。  **输出参数：**  nwy\_ssl\_version\_info\_t \*info：  填充后的结构体，包含 Mbed TLS 的版本信息。 |
| 返回值 | 成功：NWY\_SUCCESS  失败：检查nwy\_error\_e |
| 例子 | 查询当前支持的ssl 版本信息：  ret =nwy\_ssl\_version\_info\_get(&info);  if (ret == NWY\_SUCCESS) {  NWY\_SDK\_LOG\_INFO("Mbed TLS Version: %s", info.version\_string);  NWY\_SDK\_LOG\_INFO("Supported TLS Versions: 0x%02X", info.tls\_version);  NWY\_SDK\_LOG\_INFO("Supported DTLS Versions: 0x%02X", info.dtls\_version);  if (info.tls\_version & (1 << 0)) NWY\_SDK\_LOG\_INFO(" - SSL 3.0");  if (info.tls\_version & (1 << 1)) NWY\_SDK\_LOG\_INFO(" - TLS 1.0");  if (info.tls\_version & (1 << 2)) NWY\_SDK\_LOG\_INFO(" - TLS 1.1");  if (info.tls\_version & (1 << 3)) NWY\_SDK\_LOG\_INFO(" - TLS 1.2");  if (info.dtls\_version & (1 << 0)) NWY\_SDK\_LOG\_INFO(" - DTLS 1.0");  if (info.dtls\_version & (1 << 1)) NWY\_SDK\_LOG\_INFO(" - DTLS 1.2");  } |