|  |
| --- |
| OpenCPU |
| SDK HTTP API说明 |
| 版本1.1日期2025-02-26 |

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](#_Toc191470581)

[范围 iv](#_Toc191470582)

[读者对象 iv](#_Toc191470583)

[修订记录 iv](#_Toc191470584)

[符号约定 iv](#_Toc191470585)

[1 概述 5](#_Toc191470586)

[2 数据结构 6](#_Toc191470587)

[2.1 nwy\_http\_handle\_t 6](#_Toc191470588)

[2.2 nwy\_http\_event\_e 6](#_Toc191470589)

[2.3 nwy\_http\_content\_type\_e 6](#_Toc191470590)

[2.4 nwy\_http\_param\_t 6](#_Toc191470591)

[2.5 http\_request\_param\_t 7](#_Toc191470592)

[2.6 http\_get\_param\_t 7](#_Toc191470593)

[2.7 http\_head\_param\_t 7](#_Toc191470594)

[2.8 http\_put\_param\_t 7](#_Toc191470595)

[2.9 http\_post\_param\_t 7](#_Toc191470596)

[2.10 回调函数定义 7](#_Toc191470597)

[3 函数 8](#_Toc191470598)

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 | 2025-01 | 初始版本 | Li Chuyu |
| 1.1 | 2025-02 | 更新部分API描述 | 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。 |

# 概述

本文档介绍了OpenCSDK中http操作相关的API, 主要包括了以下几个方面.

* param信息

主要用于设置http服务器、端口、通知回调函数等参数。

* HTTP命令

主要用于建立连接、发送请求、关闭连接。

# 数据结构

## nwy\_http\_handle\_t

typedef void \*nwy\_http\_handle\_t;

## nwy\_http\_event\_e

typedef enum

{

NWY\_HTTP\_DNS\_ERR, //Dns error

NWY\_HTTP\_OPEN\_FAIL, //open fail

NWY\_HTTP\_OPENED, //setup success

NWY\_HTTPS\_SSL\_CONNECTED, //https setup success

NWY\_HTTP\_CLOSED\_PASV, //server closed

NWY\_HTTP\_CLOSED, //client closed

NWY\_HTTP\_DATA\_RECVED, //recv data

NWY\_HTTP\_DATA\_SEND\_ERR, //data send error

NWY\_HTTP\_EVENT\_SSL\_CONNECT\_FAIL,

NWY\_HTTP\_DATA\_SEND\_FINISHED, //data send finished

NWY\_HTTP\_EVENT\_UNKOWN, //unkown error

}nwy\_http\_event\_e;

## nwy\_http\_content\_type\_e

typedef enum

{

NWY\_HTTP\_CONTENT\_TYPE\_URLENCODE,//"Content-Type:application/x-www-form-urlencoded",

NWY\_HTTP\_CONTENT\_TYPE\_TEXT,//"Content-Type: application/text",

NWY\_HTTP\_CONTENT\_TYPE\_JSON,//"Content-Type: application/json",

NWY\_HTTP\_CONTENT\_TYPE\_XML,//"Content-Type: application/xml",

NWY\_HTTP\_CONTENT\_TYPE\_HTML,//"Content-Type: application/html",

}nwy\_http\_content\_type\_e;

## nwy\_http\_param\_t

typedef struct{

unsigned short cid;

int timeout\_s;

char \*host;

int port;

nwy\_http\_resultcb cb;//callback function

}nwy\_http\_param\_t;

## http\_request\_param\_t

typedef struct

{

unsigned int keepalive;

nwy\_http\_content\_type\_e type;

const char\* data;

int len;

char \* uri;// 资源标识符，资源路径以反斜杠“/”开头，例如：“/boot/test.txt”

}http\_request\_param\_t;

## http\_get\_param\_t

typedef struct

{

unsigned char keepalive;

int offset;

int len;

char \* uri; 资源标识符，资源路径以反斜杠“/”开头，例如：“/boot/test.txt”

}http\_get\_param\_t;

## http\_head\_param\_t

typedef struct

{

unsigned char keepalive;

char \* uri; 资源标识符，资源路径以反斜杠“/”开头，例如：“/boot/test.txt”

}http\_head\_param\_t;

## http\_put\_param\_t

typedef http\_request\_param\_t http\_put\_param\_t;

## http\_post\_param\_t

typedef http\_request\_param\_t http\_post\_param\_t;

## 回调函数定义

typedef struct{

unsigned short cid;//数据通道

nwy\_http\_handle\_t http\_handle; //http客户端句柄

unsigned int event; /\*事件id \*/

unsigned int data\_len; /\*获取接收的数据长度 \*/

char\* data; /\* 获取的数据内容\*/

}nwy\_http\_result\_t;

typedef void (\*nwy\_http\_resultcb)(nwy\_http\_result\_t \*event);

# 函数

本章节提供HTTP相关的操作接口。

此接口函数定义位于nwy\_http\_api.h，用于进行设备相关操作。

## nwy\_http\_setup

|  |  |
| --- | --- |
| 格式 | nwy\_http\_handle\_t nwy\_http\_setup(nwy\_http\_param\_t http\_param, nwy\_ssl\_conf\_t \*ssl\_cfg) |
| 描述 | 建立http/ssl http链路 |
| 参数说明 | nwy\_http\_handle\_t handle：http/ssl http客户端句柄  http\_param：  cid通道范围1-7，一个通道可支持3个http 客户端  char \*url：资源所在路径  int port, 目标端口号  nwy\_http\_resultcb cb 事件通知回调函数  int timeout设置http连接超时时间，单位秒  nwy\_app\_ssl\_conf\_t \*ssl\_cfg SSL配置信息具体定义见nwy\_ssl\_config.h,NULL 表示http,非NULL 表示使用ssl http加密 |
| 返回值 | 成功：返回http/ssl http 客户端句柄  失败：NULL |

## nwy\_http\_get

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_http\_get(nwy\_http\_handle\_t handle, http\_get\_param\_t \*param); |
| 描述 | 发起一个GET请求 |
| 参数说明 | nwy\_http\_handle\_t handle：http/ssl http客户端句柄  http\_get\_param\_t \*param:http get 参数，如下说明  unsigned char keepalive（0，1）：  0：短链接  1：长连接  int offset：偏移量指定下载起始位置.  int len：长度指定下载长度.  char\*uri：资源标识符，资源路径以反斜杠“/”开头，例如：“/boot/test.txt”. |
| 返回值 | 成功：NWY\_SUCCESS  失败：检查nwy\_error\_e |
| 注意事项 | offset 为0 和 len 为0 表示全包下载  总包大小可以通过接口nwy\_http\_get\_head()获取。  len 不为0表示断点请求下载 |

## nwy\_http\_head

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_http\_head(nwy\_http\_handle\_t handle, http\_head\_param\_t \*param) |
| 描述 | 发起一个HEAD请求,获取资源头部信息 |
| 参数说明 | nwy\_http\_handle\_t handle：http/ssl http客户端句柄  http\_head\_param\_t \*param:http head 参数，如下说明  unsigned char keepalive（0，1）：  0：短链接  1：长连接  char\*uri：资源标识符，资源路径以反斜杠“/”开头，例如：“/boot/test.txt” |
| 返回值 | 成功：NWY\_SUCCESS  失败：检查nwy\_error\_e |

## nwy\_http\_post

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_http\_post(nwy\_http\_handle\_t handle, http\_post\_param\_t \*param) |
| 描述 | 发起一个POST请求，修改资源 |
| 参数说明 | nwy\_http\_handle\_t handle：http/ssl http客户端句柄  http\_post\_param\_t \*param:http post 参数，如下说明  unsigned char keepalive（0，1）：  0：短链接  1：长连接  nwy\_http\_content\_type\_e,报文类型.  const char\* data, post数据，长度小于等于3072.  int len, 数据长度.  char\*uri：资源标识符，资源路径以反斜杠“/”开头，例如：“/boot/test.txt”. |
| 返回值 | 成功：NWY\_SUCCESS  失败：检查nwy\_error\_e |
| 注意事项 | 最大上传数据大小3072  如果要上传大数据，请使用open方式上传，调用接口nwy\_http\_open\_mode() |

## nwy\_http\_put

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_http\_put(nwy\_http\_handle\_t handle, http\_put\_param\_t \*param) |
| 描述 | 发起一个PUT请求，创建一个资源 |
| 参数说明 | nwy\_http\_handle\_t handle：http/ssl http客户端句柄  http\_put\_param\_t \*param:http put 参数，如下说明  unsigned char keepalive（0，1）：  0：短链接  1：长连接  const char\* data, 资源数据，长度小于等于3072  int len, 数据长度  nwy\_http\_content\_type\_e type:报文类型  char \*uri：资源标识符，资源路径以反斜杠“/”开头，例如：“/boot/test.txt” |
| 返回值 | 成功：NWY\_SUCCESS  失败：检查nwy\_error\_e |
| 注意事项 | 最大上传数据大小3072  如果要上传大数据，请使用open方式上传，调用接口nwy\_http\_open\_mode() |

## nwy\_http\_open\_mode

|  |  |
| --- | --- |
| 格式 | nwy\_error\_e nwy\_http\_open\_mode(nwy\_http\_handle\_t handle, unsigned char keepalive ,const char\* data, int len) |
| 描述 | 发起一个自定义报文请求 |
| 参数说明 | nwy\_http\_handle\_t handle：http/ssl http客户端句柄  unsigned char keepalive（0，1）：  0：短链接  1：长连接  const char\* data, 自定义报文数据  unsigned char len, 数据长度 |
| 返回值 | 成功：NWY\_SUCCESS  失败：检查nwy\_error\_e |
| 注意事项 | 自定义模式接口需要添加http 头和body  可用于操作客户自定义http头部信息  可用于处理大数据 |

## nwy\_http\_close

|  |  |
| --- | --- |
| 格式 | nwy\_error\_enwy\_http\_close(nwy\_http\_handle\_t handle) |
| 描述 | http/https链路主动关闭 |
| 参数说明 | nwy\_http\_handle\_t handle：http/ssl http客户端句柄 |
| 返回值 | 成功：NWY\_SUCCESS  失败：检查nwy\_error\_e |