



XR806 SHTTDP 网络模块 开发指南

版本号：1.0

发布时间：2020-11-9

版本历史

版本	日期	责任人	版本描述
1.0	2020-11-9	AWA 1096	创建文档。

目录

版本历史.....	i
目录.....	ii
表格目录.....	iv
1 前言.....	1
1.1 文档简介.....	1
1.2 目标读者.....	1
1.3 适用范围.....	1
1.4 文档约定.....	1
1.4.1 标志说明.....	1
2 概述.....	2
2.1 背景说明.....	2
2.2 规格特性.....	2
2.3 文件位置.....	2
3 应用说明.....	4
3.1 应用简述.....	4
3.2 配置说明.....	4
3.3 接口说明.....	5
3.3.1 初始化接口.....	5
3.3.1.1 shttpd_init_local_file.....	5
3.3.1.2 shttpd_init.....	6
3.3.2 释放接口.....	6
3.3.2.1 shttpd_fini.....	6
3.3.3 执行接口.....	6
3.3.3.1 shttpd_poll.....	6
3.3.4 设置接口.....	7
3.3.4.1 shttpd_set_ssl_cert.....	7
3.3.4.2 shttpd_set_option.....	7
3.3.4.3 shttpd_register_ssi_func.....	7
3.3.4.4 shttpd_register_uri.....	8
4 示例说明.....	9
4.1 示例简介.....	9
4.1.1 获取方式.....	9
4.1.2 准备工作.....	9

4.1.3 操作步骤.....	9
4.2 效果展示.....	10
4.3 实现流程.....	10



表格目录

表 2-1	SHTTPD 模块的规格特性.....	2
表 2-2	SHTTPD 网络协议的文件位置.....	2
表 2-3	SHTTPD 网络协议的文件说明.....	3
表 3-1	XR806 SHTTPD 模块配置列表.....	4
表 3-2	SHTTPD 模块接口简介.....	5
表 3-3	_shttpd_init_local_file 接口函数说明.....	5
表 3-4	shttpd_init 接口函数说明.....	6
表 3-5	shttpd_fini 接口函数说明.....	6
表 3-6	shttpd_poll 接口函数说明.....	6
表 3-7	shttpd_set_ssl_cert 接口函数说明.....	7
表 3-8	shttpd_set_option 接口函数说明.....	7
表 3-9	shttpd_register_ssl_func 接口函数说明.....	7
表 3-10	shttpd_register_uri 接口函数说明.....	8
表 4-1	SHTTPD 示例工程的命令列表.....	9

1 前言

1.1 文档简介

本文档介绍了 XR806 平台上 SHTTPD 模块的使用方法。

1.2 目标读者

XR806 开发及维护人员

1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

2 概述

2.1 背景说明

SHTTPD 提供了实现轻量级 web server 的应用接口，内部功能、特性丰富，支持 CGI, SSL, cookie, MD5 认证等。XR806 SDK 中移植版本为 1.42，裁剪并保留了部分功能，例如单链接、HTTP 1.0/ 1.1、GET/ POST、SSL、SSI，网页支持 html/shtml 等，通过上述的所支持的功能，并利用其接口，可以快速的在 XR806 上开发出 web 服务器。基于 web 服务器，用户可以实现不同的功能，比如设备控制、信息设置等。

2.2 规格特性

SHTTPD 模块提供了以下的功能特性

表 2-1 SHTTPD 模块的规格特性

规格类型	支持规格	规格描述	备注
协议标准	HTTP 1.0	HTTP 1.0 协议标准	
	HTTP 1.1	HTTP 1.1 协议标准	
功能	GET	HTTP GET 功能，请求指定的页面信息，并返回实体主题	
	POST	HTTP POST 功能，向指定资源提交数据进行处理请求	
	SSL	加密功能	
	SSI	服务器端内嵌功能	
网页类型	html	静态 html 网页	
	shtml	静态 html 网页，可以使用 SSI	

2.3 文件位置

以 SDK 包为根目录，本网络协议涉及到的主要文件位置如下。

表 2-2 SHTTPD 网络协议的文件位置

组件名	文件分类	文件位置
SHTTPD	源码文件	sdk/src/net/shttpd-1.42/
	头文件	sdk/include/net/shttpd/
	示例工程	sdk/src/net/shttpd-1.42/examples/web_server_demo.c

关键文件说明如下。

表 2-3 SHTTPD 网络协议的文件说明

文件名	文件说明
shttpd.c	SHTTPD 模块的主要 api 实现文件
shttpd.d	SHTTPD 模块的主要 api 头文件



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

3 应用说明

3.1 应用简述

SHTTPD 网络协议已经内嵌到 XR806 SDK 中，通过配置以及函数接口调用即可使用。

3.2 配置说明

表 3-1 XR806 SHTTPD 模块配置列表

配置项	配置说明
SSL 功能使能	设置说明： 此项配置 SSL 功能使能与否 设置位置： /include/net/shttpd/compat_rtos.h 设置方式： 定义宏 SHTTPD_SSL，如使能 SSL： #define SHTTPD_SSL
最大 BUFFER 配置	设置说明： 此项配置 uri 的最大 buffer 长度 设置位置： /include/net/shttpd/config.h 设置方式： 配置宏 URI_MAX 的值，如： #define URI_MAX 4096
索引文件格式	设置说明： 此项配置索引文件的格式 设置位置： /include/net/shttpd/config.h 设置方式： 配置宏 INDEX_FILES 的值，如： #define INDEX_FILE "index.html,index.htm,index.php,index.cgi"
SSL 格式类型配置	设置说明： 此项配置 SSL 模式下支持的文件格式 设置位置： /include/net/shttpd/config.h

配置项	配置说明
	设置方式： 配置宏 SSI_EXT 的值，如： <pre>#define SSI_EXT "shtml,shtm"</pre>
超时时间	设置说明： 此项配置超时时间 设置位置： /include/net/shttpd/config.h 设置方式： 配置宏 EXPIRE_TIME 的值，单位为秒，如： <pre>#define EXPIRE_TIME 10</pre>

3.3 接口说明

表 3-2 SHTTPD 模块接口简介

接口名	简要介绍
初始化接口	本组接口用于初始化 SHTTPD，属于外部接口。
释放接口	本组接口用于释放 SHTTPD 资源，属于外部接口
执行接口	本组接口用于处理连接请求，属于外部接口
设置接口	本组接口用于设置 SHTTPD，属于外部接口

3.3.1 初始化接口

3.3.1.1 _shttpd_init_local_file

表 3-3 _shttpd_init_local_file 接口函数说明

信息项	说明
原型	void _shttpd_init_local_file(struct usr_file *list, int count);
功能	初始化本地 server 资源，用户通过该函数向系统注册文件。
参数	struct usr_file *list 含义解释：本地文件链表。 使用说明：数组链表，里面存放 SHTTPD 资源 int count 含义解释：SHTTPD 资源的数目 使用说明：略。
返回值	无

3.3.1.2 shhttpd_init

表 3-4 shhttpd_init 接口函数说明

信息项	说明
原型	struct shhttpd_ctx *shhttpd_init(int argc, char *argv[]);
功能	申请并初始化 SHHTTPD 上下文。
参数	int argc 含义解释：参数的个数。 使用说明：该功能并没有使能，应定义为 0 char *argv[] 含义解释：配置的参数 使用说明：该功能并没有使能，应定义为 NULL。
返回值	SHHTTPD 句柄

3.3.2 释放接口

3.3.2.1 shhttpd_fini

表 3-5 shhttpd_fini 接口函数说明

信息项	说明
原型	void shhttpd_fini(struct shhttpd_ctx *ctx);
功能	释放资源。
参数	struct shhttpd_ctx *ctx 含义解释：SHHTTPD 句柄。 使用说明：shhttpd_init 返回的句柄
返回值	无

3.3.3 执行接口

3.3.3.1 shhttpd_poll

表 3-6 shhttpd_poll 接口函数说明

信息项	说明
原型	void shhttpd_poll(struct shhttpd_ctx *ctx, int milliseconds);
功能	处理连接请求。
参数	struct shhttpd_ctx *ctx 含义解释：SHHTTPD 句柄。 使用说明：shhttpd_init 返回的句柄 int milliseconds 含义解释：超时时间，单位 ms。 使用说明：赋值即可

信息项	说明
返回值	无

3.3.4 设置接口

3.3.4.1 shhttpd_set_ssl_cert

表 3-7 shhttpd_set_ssl_cert 接口函数说明

信息项	说明
原型	int shhttpd_set_ssl_cert(void *cert);
功能	向系统注册证书。
参数	void *cert 含义解释：证书信息。 使用说明：略
返回值	0：成功 -1：失败

3.3.4.2 shhttpd_set_option

表 3-8 shhttpd_set_option 接口函数说明

信息项	说明
原型	int shhttpd_set_option(struct shhttpd_ctx *ctx, const char *opt, const char *val);
功能	设置服务器选项新的参数（如端口）。
参数	struct shhttpd_ctx *ctx 含义解释：SHHTTPD 句柄。 使用说明：shhttpd_init 返回的句柄 const char *opt 含义解释：参数名称指针。 使用说明：如“ssl_cert”，“ports” const char *val 含义解释：参数值指针。 使用说明：略
返回值	0：成功 -1：失败

3.3.4.3 shhttpd_register_ssi_func

表 3-9 shhttpd_register_ssi_func 接口函数说明

信息项	说明
原型	void shhttpd_register_ssi_func(struct shhttpd_ctx *ctx, const char *name, shhttpd_callback_t func, void *user_data);

信息项	说明
功能	设置相应 name 字段的回调函数。
参数	struct shhttpd_ctx *ctx 含义解释：SHHTTPD 句柄。 使用说明：shhttpd_init 返回的句柄 const char *name 含义解释：指定 name 字符串指针。 使用说明：略 shhttpd_callback_t func 含义解释：用户回调函数。 使用说明：略 void *user_data 含义解释：回调函数的参数。 使用说明：略
返回值	无

3.3.4.4 shhttpd_register_uri

表 3-10 shhttpd_register_uri 接口函数说明

信息项	说明
原型	void shhttpd_register_uri(struct shhttpd_ctx *ctx, const char *uri, shhttpd_callback_t callback, void *data);
功能	设置相应 name 字段的回调函数。
参数	struct shhttpd_ctx *ctx 含义解释：SHHTTPD 句柄。 使用说明：shhttpd_init 返回的句柄 const char *uri 含义解释：指定的资源标志符。 使用说明：略 shhttpd_callback_t callback 含义解释：用户回调函数。 使用说明：略 void *data 含义解释：回调函数的参数。 使用说明：略
返回值	无

4 示例说明

SHTTPD 接口用于创建轻量级 web server，下面以创建 web server 为例，简要说明 SHTTPD 接口的使用。

4.1 示例简介

SHTTPD 示例的演示的目的是简要介绍 SHTTPD 接口的基本使用方法，演示的功能为创建一个 web server。

4.1.1 获取方式

SHTTPD 源码中有示例工程代码，位于 XR806 SDK 的/src/net/shttpd-1.42/examples/web_server_demo.c，可通过串口命令使用该示例工程。串口命令实现的源码位置位于/project/common/cmd/cmd_httpd.c



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

4.1.2 准备工作

SHTTPD 示例工程的硬件准备如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 Uart0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

SHTTPD 的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》

4.1.3 操作步骤

首先添加 SHTTPD 的串口命令到使用的工程，完成编译烧写，配置网络后，启动 SHTTPD。

表 4-1 SHTTPD 示例工程的命令列表

命令	说明
快速配置网络	命令格式： net sta config <ssid> [psk] 命令解释： 快速配置被连接 AP 的 SSID 和密码 参数说明： ssid：被连接 AP 的 ssid psk：连接 AP 的密码，如果为开放网络，则不需要此项。
启用网络	命令格式： net sta enable 命令解释：

命令	说明
	开始连接 AP，需要先完成网络配置
启动 SHTTPD	命令格式： net httpd 1 命令解释： 启动 SHTTPD

4.2 效果展示

在评估板中启动 SHTTPD 后，在控制台中会打印输出启动成功与否的信息，如下所示。

```
<ACK> 200 OK
```

此时可以在其它设备上，使用浏览器访问创建的 web server。

4.3 实现流程

第一步：注册服务器文件资源

```
_shttpd_init_local_file(file_list,ARRAY_SIZE(file_list));

//file_list 为本地文件资源，通过该函数将本地文件资源注册到服务器中。

struct usr_file file_list[] = {
    {"/", ""},
    {"/index.html", index_html},
    {"/config.shtml", config_shtml},
    #if !defined(NO_CHECKSUM)
    {"/checksum1.txt", checksum1_txt},
    {"/checksum2.txt", checksum2_txt},
    #endif
    {NULL},
};
```

第二步：服务器初始化

```
if ((ctx = shttpd_init(argc, argv)) == NULL) {
    _shttpd_elog(E_FATAL, NULL, "Cannot initialize SHTTPD context.");
    return -1;
}
```

第三步：设置服务器选项及参数

```
#if defined(SHTTPD_SSL)
```

```
ca_param.nCa = mbedtls_test_cas_pem_len;
ca_param.pCa = (char *)mbedtls_test_cas_pem;
ca_param.nCert = mbedtls_test_srv_cert_len;
ca_param.pCert = (char *)mbedtls_test_srv_cert;
ca_param.nKey = mbedtls_test_srv_key_len;
ca_param.pKey = (char *)mbedtls_test_srv_key;
```

```
shttpd_set_ssl_cert((void*) &ca_param);
shttpd_set_option(ctx, "ssl_cert", NULL);
shttpd_set_option(ctx, "ports", "443s");
#else
shttpd_set_option(ctx, "ports", "80");
#endif
```

如果使用 ssi（提供 CALL，用于解析 shtml 文件）机制，需要向系统注册对应关键字的回调函数

```
shttpd_register_ssi_func(ctx, "DeviceName", ssi_get_ssid, ap_info.g_ssid);
shttpd_register_ssi_func(ctx, "DevicePasswd", ssi_get_passwd, ap_info.g_passwd);
```

如果实现 post 机制，需要向系统注册对应 uri 的回调函数

```
shttpd_register_uri(ctx, "/devicename", set_ap_info, (void *)&ap_info);

#if !defined(NO_CHECKSUM)
shttpd_register_uri(ctx, "/post/key/checksum1_txt", set_post_info, NULL);
shttpd_register_uri(ctx, "/post/key/checksum2_txt", set_post_info, NULL);
shttpd_register_uri(ctx, "/get/key/checksum1_txt", set_post_info, NULL);
#endif
```

第四步：调用服务器运行处理函数，处理相应的连接

```
_shttpd_exit_flag = 0;
while (_shttpd_exit_flag == 0)
    shttpd_poll(ctx, 1000);
```

第五步：释放资源，服务器退出

```
shttpd_fini(ctx);
```

著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本文档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明



（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。