



XRADIO SNTP Developer Guide

Revision 1.0

Oct 22, 2019

Declaration

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF XRADIO TECHNOLOGY (“XRADIO”). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF XRADIO AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE PURCHASED PRODUCTS, SERVICES AND FEATURES ARE STIPULATED BY THE CONTRACT MADE BETWEEN XRADIO AND THE CUSTOMER. PLEASE READ THE TERMS AND CONDITIONS OF THE CONTRACT AND RELEVANT INSTRUCTIONS CAREFULLY BEFORE USING, AND FOLLOW THE INSTRUCTIONS IN THIS DOCUMENTATION STRICTLY. XRADIO ASSUMES NO RESPONSIBILITY FOR THE CONSEQUENCES OF IMPROPER USE (INCLUDING BUT NOT LIMITED TO OVERVOLTAGE, OVERCLOCK, OR EXCESSIVE TEMPERATURE).

THE INFORMATION FURNISHED BY XRADIO IS PROVIDED JUST AS A REFERENCE OR TYPICAL APPLICATIONS, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT DO NOT CONSTITUTE A WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. XRADIO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE.

NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF XRADIO. THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADIO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADIO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

Revision History

Version	Date	Summary of Changes
1.0	2019-10-22	Initial Version

Contents

Declaration.....	2
Revision History.....	3
Contents.....	4
1 模块概要.....	5
1.1 功能介绍.....	5
1.2 代码位置.....	5
2 模块结构体.....	6
3 模块接口.....	7
4 模块接口例程.....	8
5 其他配置.....	9
5.1 配置选项.....	9
5.2 关于 ntp 服务器地址的说明:	9

1 模块概要

1.1 功能介绍

SNTP 是简单网络时间协议（Simple Network Time protocol）的简称，它是目前 Internet 网上实现时间同步的一种重要工程化方法。

1.2 代码位置

模块	文件类型	代码位置
SNTP	source	sdk/src/net/sntp/sntp.c
	header	sdk/include/net/sntp/sntp.h
	demo	sdk/project/common/cmd/cmd_sntp.c
		sdk/project/example/sntp/main.c

2 模块结构体

这是 sntp 参数结构体:

```
typedef struct {
    char *server_name; //远程 ntp 服务器地址，如果该参数不为空，则会优先使用该地址，如果为空，则使用默认的 ntp 地址
    int recv_timeout; //接收超时时间
    uint8_t retry_times; //超时时循环发送请求包的次数
} sntp_arg;
```

该结构体是旧的服务器返回时间结果结构体，已经弃用。

```
typedef struct {
    uint8_t sec; //秒 [0,59]
    uint8_t min; //分 [0,59]
    uint8_t hour; //时[0,23]
    uint8_t day; //日[1,31]
    uint8_t mon; //月[1,12]
    uint8_t week; //星期[0,6]
    uint8_t year; //年[0,127]，该参数对年进行了%100，故 2019 会表示为 19
} sntp_time;
```

3 模块接口

int sntp_request(void *arg)

功能	请求远程 NTP/SNTP 服务器时间。
参数	arg: 该参数无效, 可直接设置为 NULL
返回值	0: 成功, -1: 失败
备注	该接口是旧的接口, 已弃用

void* sntp_obtain_time()

功能	获取时间, 在调用 sntp_request 之后使用。
参数	void
返回值	返回 sntp_time 类型指针
备注	该接口是旧的接口, 已弃用

int sntp_get_time(sntp_arg *arg, struct timeval *ntp_time)

功能	从远程服务器获取时间。
参数	ntp_time: timeval 类型结构体, 用于存储服务器返回的时间 arg: sntp 参数结构体指针
返回值	0: 成功, -1: 失败

int sntp_set_server(uint8_t idx, char *server_name);

功能	设置 ntp 服务器地址
参数	idx: ntp 地址索引 server_name: 服务器地址
返回值	0: 成功, -1: 失败
备注	

4 模块接口例程

本节提供上节介绍接口的示例，描述接口的使用方法及流程，文中代码皆为参考代码，不能直接运行，运行代码可直接参考 sdk 示例工程 `sdk/project/example/sntp/main.c` 或文件 `cmd_sntp.c`。

第一步：获取时间

```
ret = sntp_get_time(NULL, &ntp_time);
```

第二步：转换时间

```
if (ret == 0) {
    time = localtime(&ntp_time.tv_sec);
    printf("<sntp> %u-%02u-%02u %02u:%02u:%02u\n",
        time->tm_year + 1900, time->tm_mon + 1, time->tm_mday,
        time->tm_hour, time->tm_min, time->tm_sec, time->tm_wday);
}
```


5 其他配置

5.1 配置选项

1. 默认远端 NTP 服务器地址

```
文件: sdk/src/net/sntp/sntp.h
#define SNTP_SERVER_ADDRESS      "pool.ntp.org"
```

2. sntp 接收超时时间

```
文件: sdk/src/net/sntp/sntp.h
#define SNTP_RECV_TIMEOUT      3000 /* ms */
```

3. 获取超时时重传的次数

```
文件: sdk/src/net/sntp/sntp.h
#define SNTP_RETRY_TIMES      3
```

4. 配置是否支持多 ntp 服务器文件: sdk/src/net/sntp/sntp.h

```
文件: sdk/src/net/sntp/sntp.h
#define SNTP_SUPPORT_MULTIPLE_SERVERS  1
```

5.2 关于 ntp 服务器地址的说明:

sntp 模块共有三个 ntp 服务器地址来源:

1. sntp_get_time 接口传入的 sntp_arg.server_name
2. sntp_set_server 接口传入的 server_name
3. 默认的 ntp 服务器

sntp 选择服务器地址的优先级:

优先选择 sntp_get_time 接口传入的 sntp_arg.server_name, 如果 sntp_get_time 传入的参数为 NULL, 则按照顺序选择 sntp_set_server 接口设置的 server_name, 如果前两项都没有设置, 则使用默认的 ntp 服务器。