SCM1612
Wi-Fi 6 and BLE 5 Low-Power SoC

SNTP Development Guide

Revision 0.1
Date 2024-3-13

Contact Information
Senscomm Semiconductor (www.senscomm.com)
Room 303, International Building, West 2 Suzhou Avenue,
SIP, Suzhou, China
For sales or technical support, please send email to
info@senscomm.com

Disclaimer and Notice

This document is provided on an “as-is” basis only. Senscomm reserves the right to make corrections, improvements and other changes to it or any specification contained herein without further notice.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

All third party’s information in this document is provided as is with NO warranties to its authenticity and accuracy.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners and are hereby acknowledged.

Version History

Version	Date	Description
0.1	2024-3-13	Initial draft

Table of Contents

1 Introduction

This document serves as a guide that helps implementing applications that requires running an SNTP.

1.1 Overview

The SCM1612 SDK uses the lwIP's sntp port:

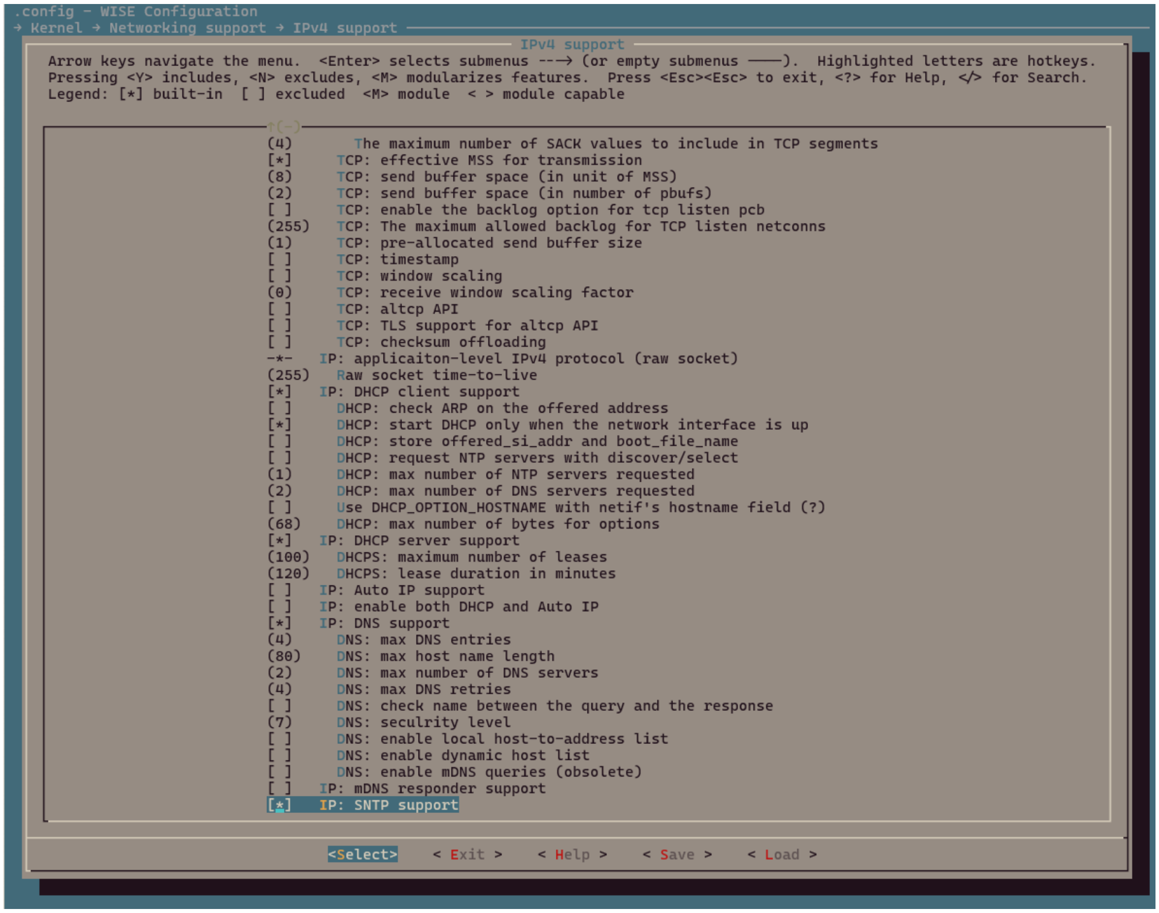
- API and CLI
  - Located in: lib/net/sntp

1.2 Build

To use SNTP APIs and CLI, user should enable corresponding feature in build configuration.

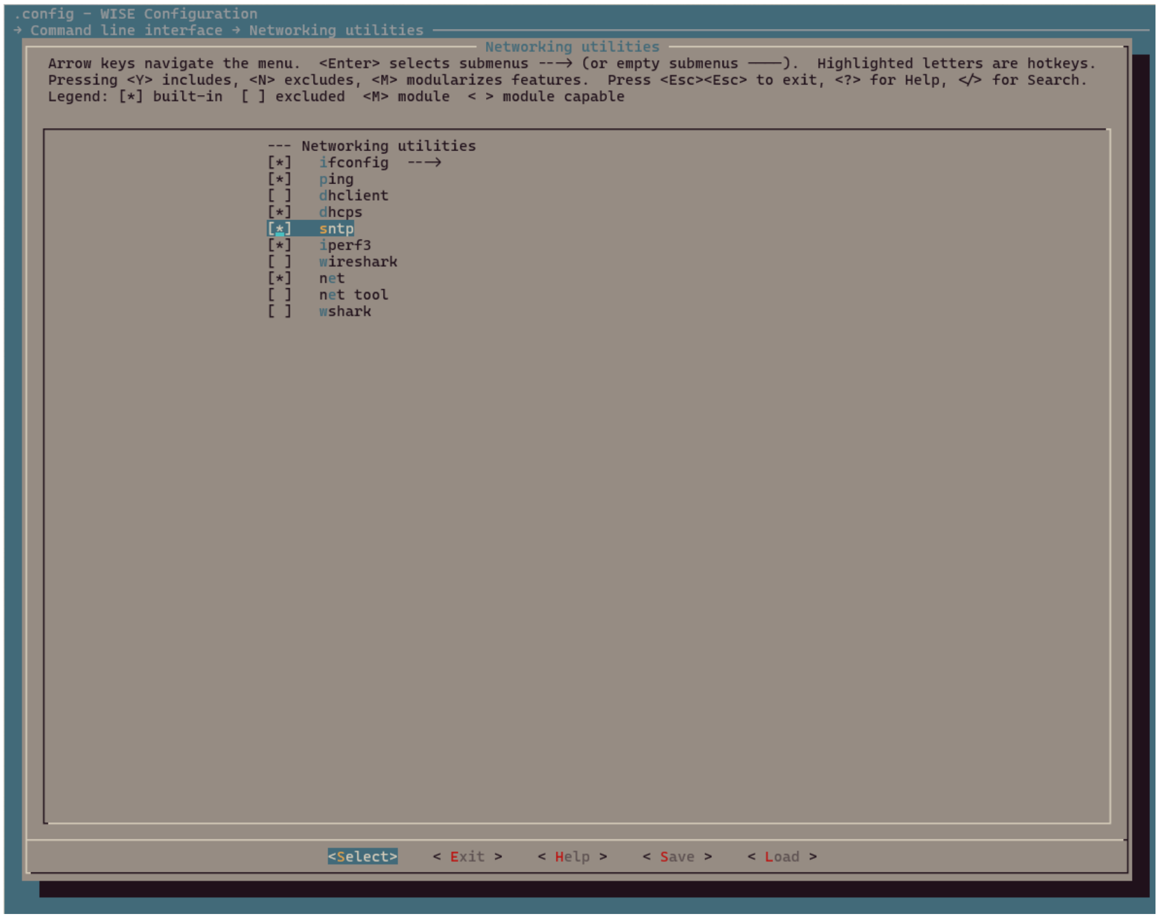
```
$ make scm1612s_defconfig
$ make menuconfig
```

Navigate to `Kernel -> Networking support -> IPv4 support -> IP: SNTP support` and enable it.



```
.config - WISE Configuration
→ Kernel → Networking support → IPv4 support ─────────────────────────────
                                   IPv4 support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.
  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                          ↑(-)
                          (4)         The maximum number of SACK values to include in TCP segments
                          [*]       TCP: effective MSS for transmission
                          (8)       TCP: send buffer space (in unit of MSS)
                          (2)       TCP: send buffer space (in number of pbufs)
                          [ ]       TCP: enable the backlog option for tcp listen pcb
                          (255)     TCP: The maximum allowed backlog for TCP listen netconns
                          (1)       TCP: pre-allocated send buffer size
                          [ ]       TCP: timestamp
                          [ ]       TCP: window scaling
                          (0)       TCP: receive window scaling factor
                          [ ]       TCP: altcp API
                          [ ]       TCP: TLS support for altcp API
                          [ ]       TCP: checksum offloading
                          -*-     IP: applicaiton-level IPv4 protocol (raw socket)
                          (255)     Raw socket time-to-live
                          [*]     IP: DHCP client support
                          [ ]       DHCP: check ARP on the offered address
                          [*]       DHCP: start DHCP only when the network interface is up
                          [ ]       DHCP: store offered_si_addr and boot_file_name
                          [ ]       DHCP: request NTP servers with discover/select
                          (1)       DHCP: max number of NTP servers requested
                          (2)       DHCP: max number of DNS servers requested
                          [ ]       Use DHCP_OPTION_HOSTNAME with netif's hostname field (?)
                          (68)      DHCP: max number of bytes for options
                          [*]     IP: DHCP server support
                          (100)     DHCPS: maximum number of leases
                          (120)     DHCPS: lease duration in minutes
                          [ ]     IP: Auto IP support
                          [ ]     IP: enable both DHCP and Auto IP
                          [*]     IP: DNS support
                          (4)       DNS: max DNS entries
                          (80)      DNS: max host name length
                          (2)       DNS: max number of DNS servers
                          (4)       DNS: max DNS retries
                          [ ]       DNS: check name between the query and the response
                          (7)       DNS: seculrity level
                          [ ]       DNS: enable local host-to-address list
                          [ ]       DNS: enable dynamic host list
                          [ ]       DNS: enable mDNS queries (obsolete)
                          [ ]     IP: mDNS responder support
                          [*]     IP: SNTP support

                <Select>    < Exit >    < Help >    < Save >    < Load >
```

Also, navigate to `Command Line Interface -> Networking utilities -> sntp` and enable it.

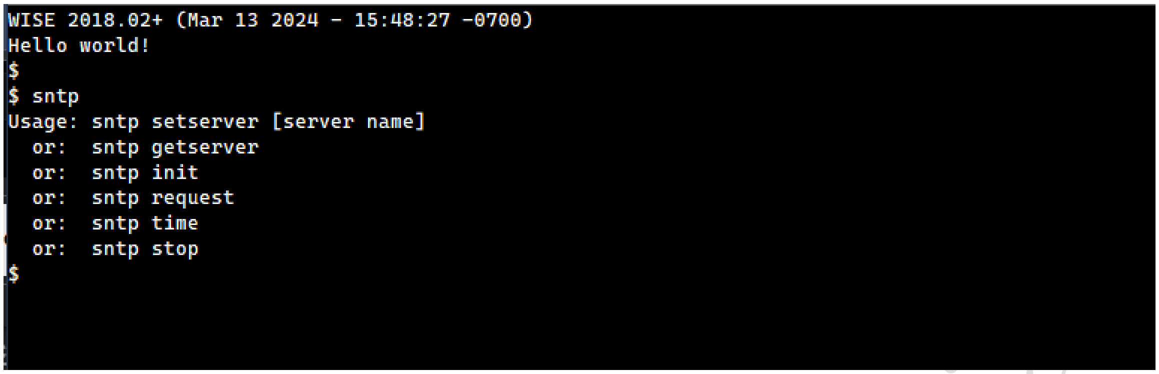


```
.config - WISE Configuration
→ Command line interface → Networking utilities ──────────────────────────────────
                                     Networking utilities
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.
  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.
  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                         --- Networking utilities
                         [*]   ifconfig  --->
                         [*]   ping
                         [ ]   dhclient
                         [*]   dhcps
                         [*]   sntp
                         [*]   iperf3
                         [ ]   wireshark
                         [*]   net
                         [ ]   net tool
                         [ ]   wshark

                  <Select>    < Exit >    < Help >    < Save >    < Load >
```

Exit & Save, then build `wise-mcuboot.bin`:

$ make

Refer to the `SDK_Getting_Started_Guide` to download the image and run it on an SCM1612 EVK.
You will be able to confirm that relevant CLI commands are available as follows.



```
WISE 2018.02+ (Mar 13 2024 - 15:48:27 -0700)
Hello world!
$
$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
```

2 API

SNTP API provides the following set of functions to set up an SNTP module and send queries to get network time.

- sntp_setservername
- sntp_init
- sntp_set_time_sync_notification_cb
- sntp_stop

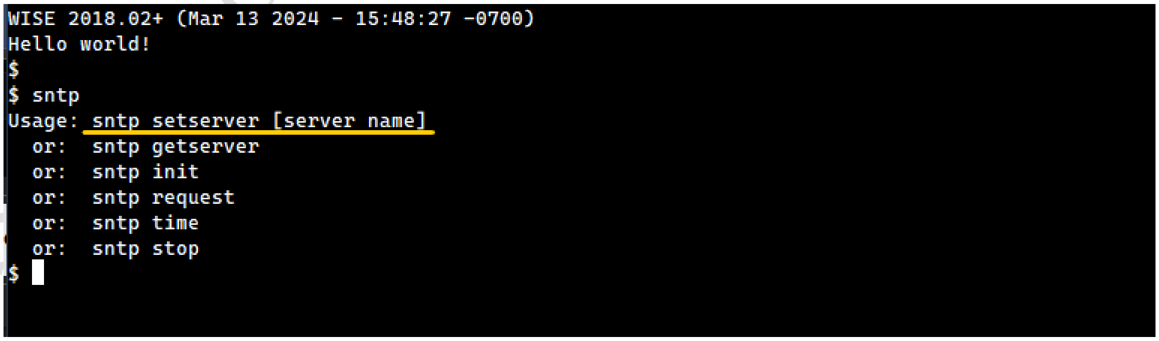
2.1 Set up and tear down

- void sntp_setservername(u8_t idx, const char *server)

  Initialize one of the NTP servers by name.

Parameter	Description
Idx	The index of the NTP server to set must be < NTP_MAX_SERVERS.
server	DNS name of the NTP server to set, to be resolved at contact time.

Corresponding CLI is:



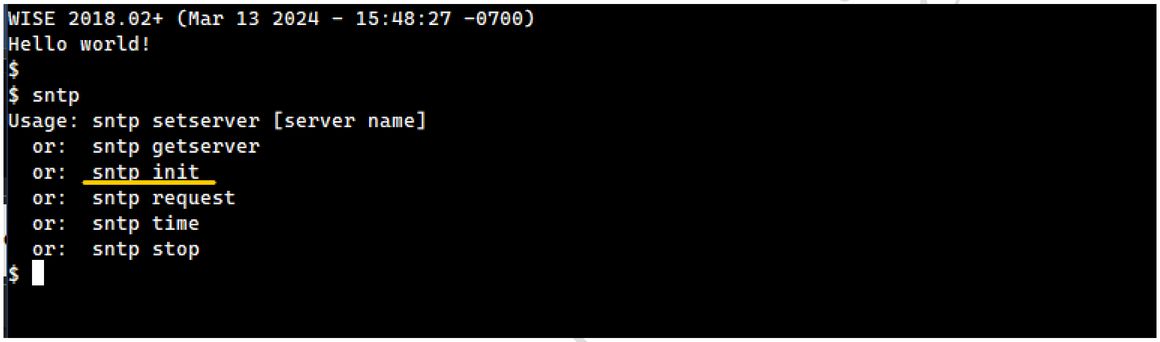
```
WISE 2018.02+ (Mar 13 2024 - 15:48:27 -0700)
Hello world!
$
$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
```

- void sntp_set_time_sync_notification_cb(void (*callback)(uint32_t sec, uint32_t us))

Install a callback to be executed when SNTP server’s update will be available.

Parameter	Description
callback	Callback to be executed when network time will be available

Corresponding CLI is:

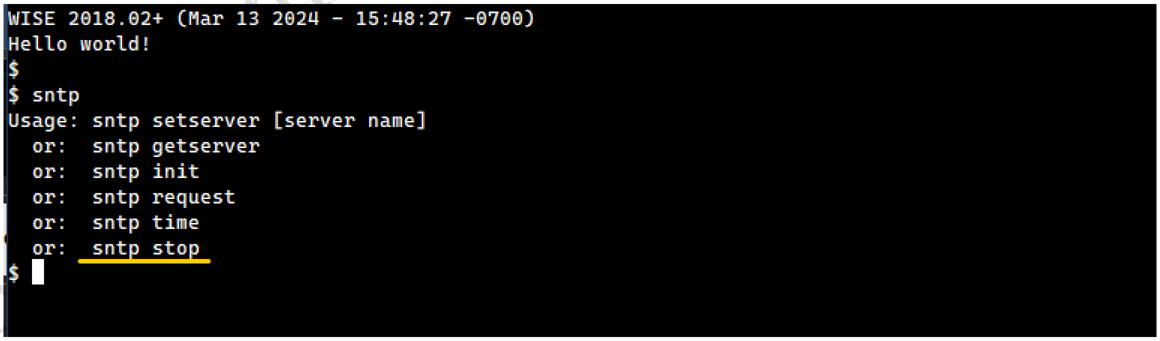


```
WISE 2018.02+ (Mar 13 2024 - 15:48:27 -0700)
Hello world!
$
$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
```

- void sntp_stop(stop)

Stops the SNTP module.

Corresponding CLI is:



```
WISE 2018.02+ (Mar 13 2024 - 15:48:27 -0700)
Hello world!
$
$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
```

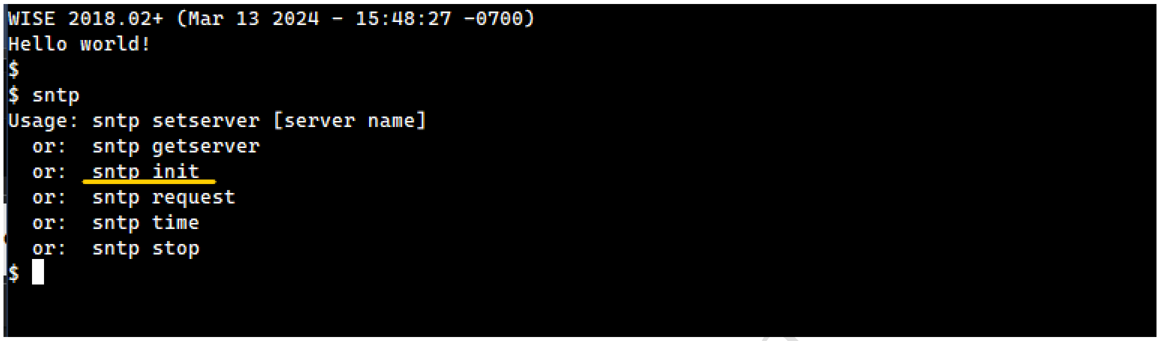
2.2 Initiate time-sync operation

- void sntp_init(void)

Initialize this module and sends out a request instantly or after

SNTP_STARTUP_DELAY(_FUNC).

Corresponding CLI is:



```
WISE 2018.02+ (Mar 13 2024 - 15:48:27 -0700)
Hello world!
$
$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
```

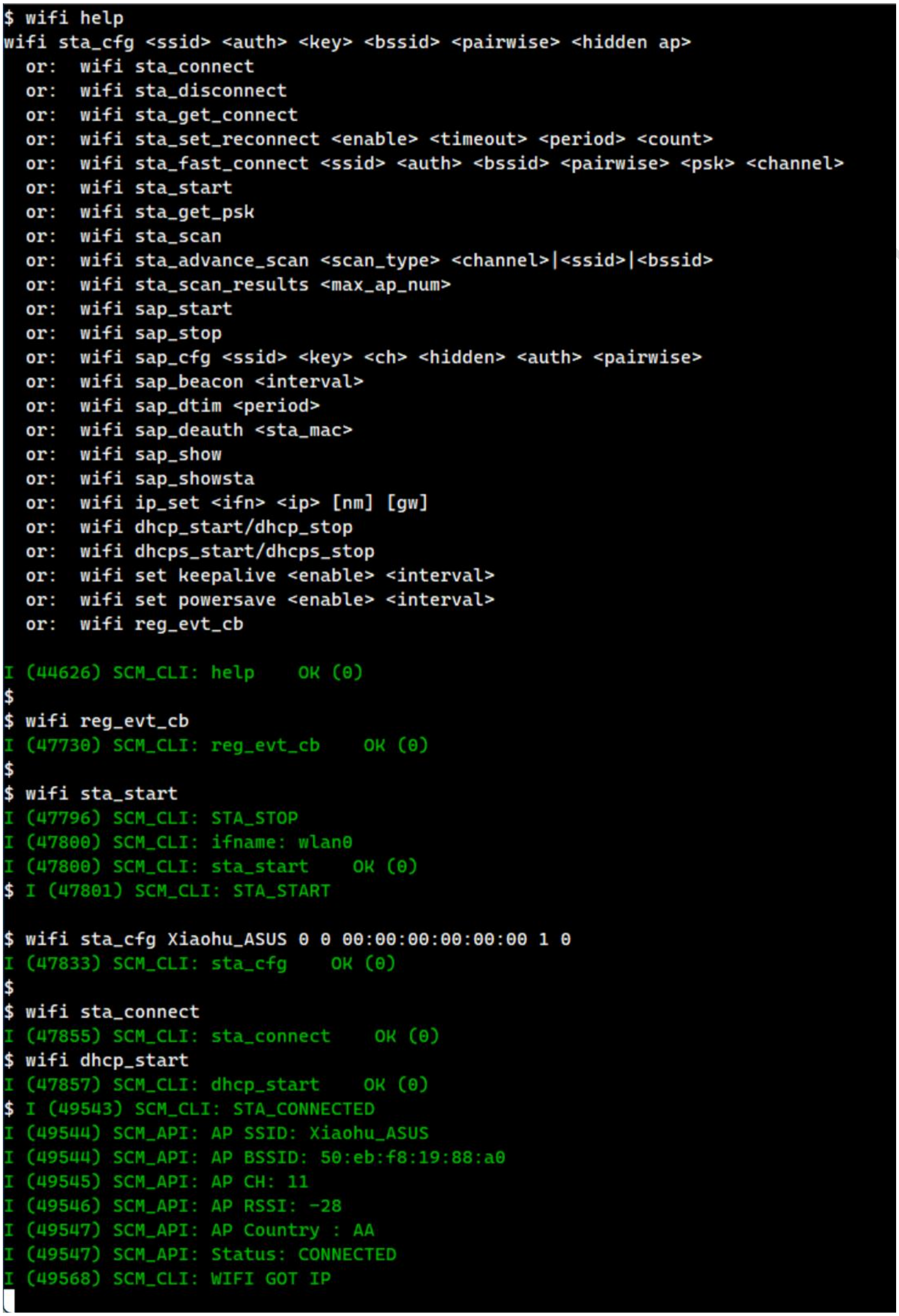
3 Demo

There is no dedicated demo application for SNTP. Instead, SNTP CLI commands introduced above can be used to test its functionality and more.

While SNTP APIs will take care of SNTP protocol itself, there should be a way to synchronize it with the system time. For this purpose, user application can use `gettimeofday` and `settimeofday` functions, and this will also be shown in the CLI demo.

3.1 Connect to an AP

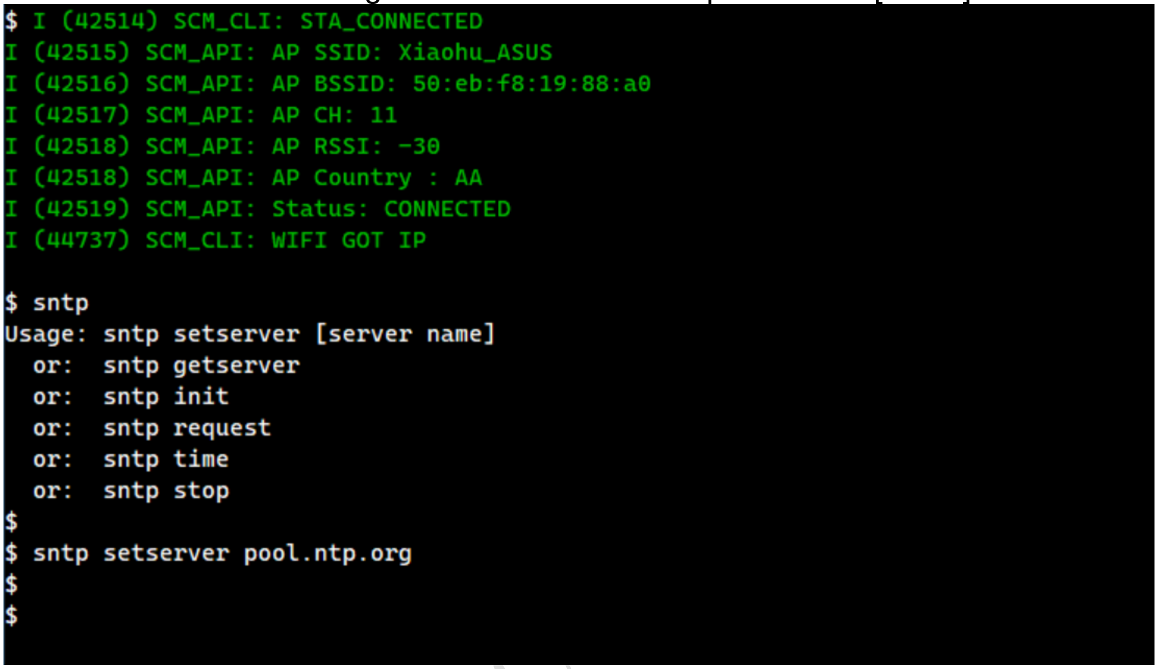
Wi-Fi STA CLI commands can be used to connect the station interface, i.e., `wlan0`, to an AP. Refer to `SCM1612_Wi-Fi_Software_Development_Guide` for use of Wi-Fi station CLI commands.



```
$ wifi help
wifi sta_cfg <ssid> <auth> <key> <bssid> <pairwise> <hidden ap>
  or:  wifi sta_connect
  or:  wifi sta_disconnect
  or:  wifi sta_get_connect
  or:  wifi sta_set_reconnect <enable> <timeout> <period> <count>
  or:  wifi sta_fast_connect <ssid> <auth> <bssid> <pairwise> <psk> <channel>
  or:  wifi sta_start
  or:  wifi sta_get_psk
  or:  wifi sta_scan
  or:  wifi sta_advance_scan <scan_type> <channel>|<ssid>|<bssid>
  or:  wifi sta_scan_results <max_ap_num>
  or:  wifi sap_start
  or:  wifi sap_stop
  or:  wifi sap_cfg <ssid> <key> <ch> <hidden> <auth> <pairwise>
  or:  wifi sap_beacon <interval>
  or:  wifi sap_dtim <period>
  or:  wifi sap_deauth <sta_mac>
  or:  wifi sap_show
  or:  wifi sap_showsta
  or:  wifi ip_set <ifn> <ip> [nm] [gw]
  or:  wifi dhcp_start/dhcp_stop
  or:  wifi dhcps_start/dhcps_stop
  or:  wifi set keepalive <enable> <interval>
  or:  wifi set powersave <enable> <interval>
  or:  wifi reg_evt_cb

I (44626) SCM_CLI: help    OK (0)
$
$ wifi reg_evt_cb
I (47730) SCM_CLI: reg_evt_cb    OK (0)
$
$ wifi sta_start
I (47796) SCM_CLI: STA_STOP
I (47800) SCM_CLI: ifname: wlan0
I (47800) SCM_CLI: sta_start    OK (0)
$ I (47801) SCM_CLI: STA_START

$ wifi sta_cfg Xiaohu_ASUS 0 0 00:00:00:00:00:00 1 0
I (47833) SCM_CLI: sta_cfg    OK (0)
$
$ wifi sta_connect
I (47855) SCM_CLI: sta_connect    OK (0)
$ wifi dhcp_start
I (47857) SCM_CLI: dhcp_start    OK (0)
$ I (49543) SCM_CLI: STA_CONNECTED
I (49544) SCM_API: AP SSID: Xiaohu_ASUS
I (49544) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (49545) SCM_API: AP CH: 11
I (49546) SCM_API: AP RSSI: -28
I (49547) SCM_API: AP Country : AA
I (49547) SCM_API: Status: CONNECTED
I (49568) SCM_CLI: WIFI GOT IP
```

3.2 Configure SNTP server by name

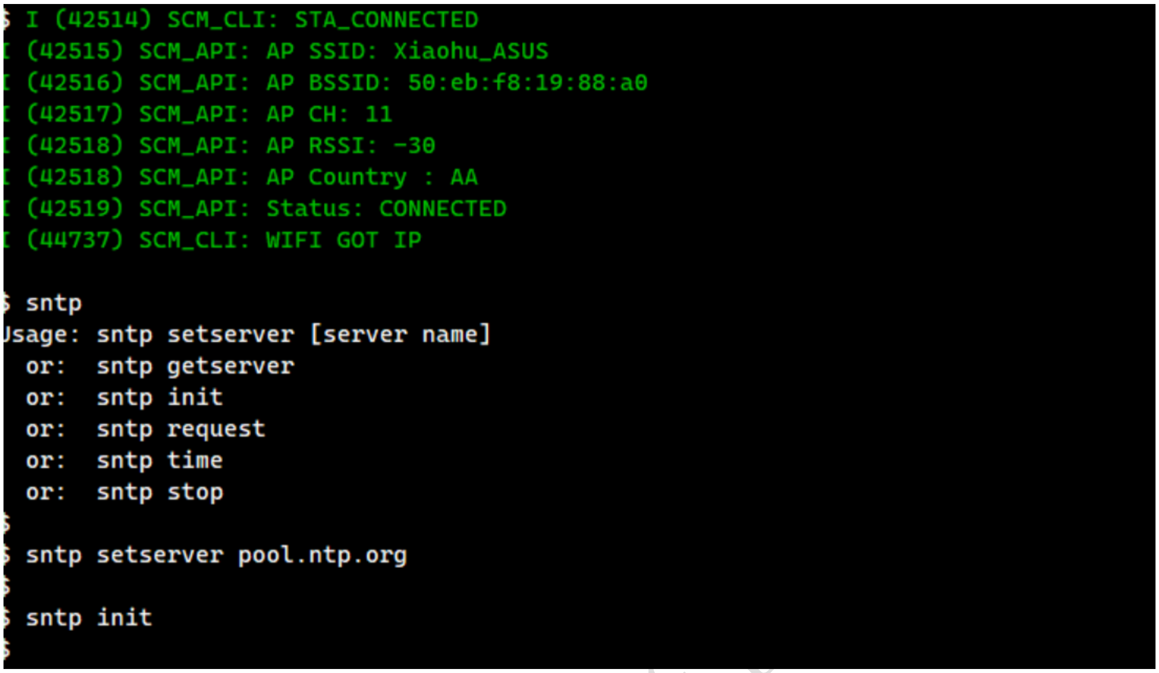
Set the SNTP server using the CLI command `sntp setserver [name]`.



```
$ I (42514) SCM_CLI: STA_CONNECTED
I (42515) SCM_API: AP SSID: Xiaohu_ASUS
I (42516) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (42517) SCM_API: AP CH: 11
I (42518) SCM_API: AP RSSI: -30
I (42518) SCM_API: AP Country : AA
I (42519) SCM_API: Status: CONNECTED
I (44737) SCM_CLI: WIFI GOT IP

$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
$ sntp setserver pool.ntp.org
$
$
```

3.3 Start SNTP service

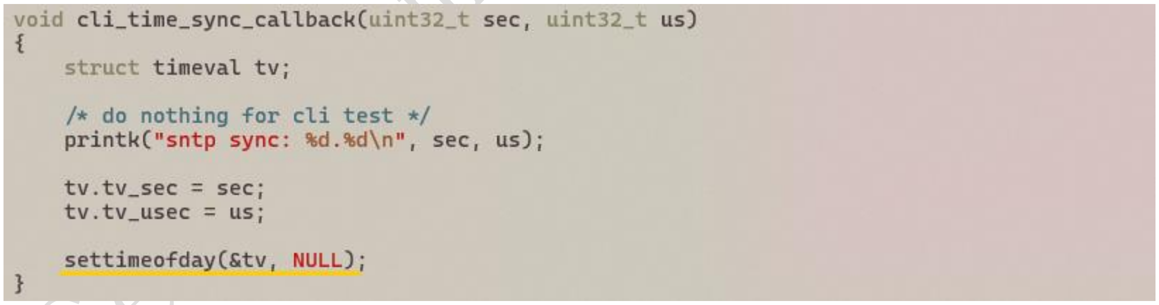
Start the SNTP service using the CLI command `sntp init`.



```
$ I (42514) SCM_CLI: STA_CONNECTED
I (42515) SCM_API: AP SSID: Xiaohu_ASUS
I (42516) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (42517) SCM_API: AP CH: 11
I (42518) SCM_API: AP RSSI: -30
I (42518) SCM_API: AP Country : AA
I (42519) SCM_API: Status: CONNECTED
I (44737) SCM_CLI: WIFI GOT IP

$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
$ sntp setserver pool.ntp.org
$
$ sntp init
$
```

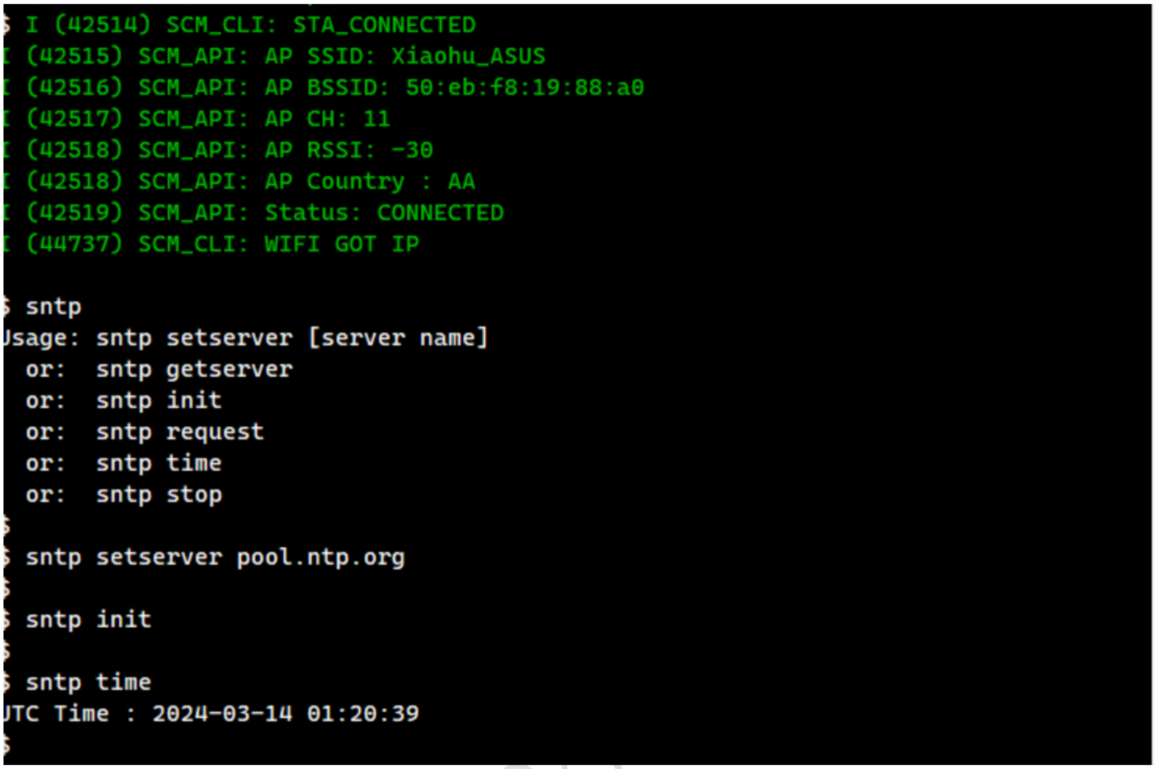
Now, the SNTP client request will be sent to the configured server, which will return its response with current network time.

The CLI command `sntp init` has also registered a callback by `sntp_set_time_sync_notification_cb`, illustrating how to synchronize system local time with the network time returned by the SNTP server.



```
void cli_time_sync_callback(uint32_t sec, uint32_t us)
{
    struct timeval tv;

    /* do nothing for cli test */
    printk("sntp sync: %d.%d\n", sec, us);

    tv.tv_sec = sec;
    tv.tv_usec = us;

    settimeofday(&tv, NULL);
}
```

After this callback have been called, the system time will be in sync with network time, which will be retrieved by calling `settimeofday`.
This is demonstrated by the CLI command `sntp time` as follows:



```
$ I (42514) SCM_CLI: STA_CONNECTED
I (42515) SCM_API: AP SSID: Xiaohu_ASUS
I (42516) SCM_API: AP BSSID: 50:eb:f8:19:88:a0
I (42517) SCM_API: AP CH: 11
I (42518) SCM_API: AP RSSI: -30
I (42518) SCM_API: AP Country : AA
I (42519) SCM_API: Status: CONNECTED
I (44737) SCM_CLI: WIFI GOT IP

$ sntp
Usage: sntp setserver [server name]
  or:  sntp getserver
  or:  sntp init
  or:  sntp request
  or:  sntp time
  or:  sntp stop
$
$ sntp setserver pool.ntp.org
$
$ sntp init
$
$ sntp time
UTC Time : 2024-03-14 01:20:39
$
```