Simple Network Time Protocol

# Overview

This application note provides details on the sntpapplication to fetch time from the NTP server by using the Simple Network Time Protocol (SNTP) component of the SDK for Talaria TWO modules.

# Simple Network Time Protocol (SNTP)

SNTP is a networking protocol for synchronizing the computer clocks to some reference over a network. SNTP is a stripped-down version of NTP that is suited to small networks and computers with limited processing power.

# Relevant APIs

## SNTP APIs

### sntp\_setoperatingmode()

Sets the operating mode.

|  |
| --- |
| sntp\_setoperatingmode(SNTP\_OPMODE\_POLL); |

SNTP has two operating modes:

1. SNTP\_OPMODE\_POLL: Poll using unicast. Requires server name to be set.
2. SNTP\_OPMODE\_LISTENONLY: This mode requires broadcast NTP on the network.

The example in this document uses SNTP\_OPMODE.

### sntp\_init()

Initializes the module.

|  |
| --- |
| sntp\_init(); |

### sntp\_stop()

Stops the module.

|  |
| --- |
| sntp\_stop(); |

### os\_systime64()

Returns the current system time in microseconds.

|  |
| --- |
| uint64\_t now = os\_systime64();  os\_printf("\r\n time:%lld \n", now); |

# Code Walkthrough

## Connecting to a Wi-Fi network

To connect to a Wi-Fi network, wcm\_create()API from the Wi-Fi Connection Manager are used.

Initially, the Wi-Fi network interface is created using wcm\_create().

|  |
| --- |
| h = wcm\_create(NULL); |

wifi\_connect\_to\_network()API, from components library, connects to the Wi-Fi network using the AP credentials provided.

|  |
| --- |
| rval = wifi\_connect\_to\_network(&h, WCM\_CONN\_WAIT\_INFINITE, &wcm\_connect\_success);  if(rval < 0) {  os\_printf("\nError: Unable to connect to network\n");  return 0;  } |

## Getting time from NTP

In the main function, application initiates the Wi-Fi connection. After the connection is successful, the ntpdate() function is called to fetch the time.

|  |
| --- |
| print\_ver("Using NTP Application", 1, 1);  os\_sem\_init(&app\_wcm\_lock, 0);  /\*wifi connection\*/  wcm\_handle = wcm\_create(NULL);  /\*Connect to WiFi N/w\*/  app\_wcm\_connect(wcm\_handle);  if(!wcm\_connected)  {  os\_printf("\n [APP]Error: Failed to connect to WiFi N/w");  return false;  }  os\_printf("Wifi connected \n");  ntpdate(); |

The SNTP module can be initiated with the operating mode of SNTP\_OPMODE\_POLL. This creates the NTP interface and fetches time. The SNTP will fetch the time data from the default server time.google.com.

The time.google.com automatically picks time servers which are geographically close to Talaria TWO.

|  |
| --- |
| int status;  int time\_now;  int times = 0;  ip\_addr\_t server\_ip;  time\_t tim;  sntp\_stop();  sntp\_setoperatingmode(SNTP\_OPMODE\_POLL);  server\_addr = "216.239.35.0";/\*time.google.org\*/  status = ipaddr\_aton(server\_addr, &server\_ip);  if(status != 1) {  return false;  }  sntp\_init();  sntp\_setserver(0, &server\_ip);  os\_msleep(2000);  uint64\_t now = os\_systime64();  os\_printf("\r\ntime:%lld \n", now);  do{  time\_now = sntp\_time();  os\_printf("\r\nwaiting for sntp, times=%d:%d\n", times++, time\_now);  if(0 != time\_now)  {  break;  }  os\_msleep(2000);  }  while(times < 16);  if(times >= 16)  return 0;  tim = time\_now;  os\_printf("\r\ndate: %s\r\n", ctime(&tim));  sntp\_stop(); |

# Running the Application

## Programming Talaria TWO Device using Download Tool

Program sntp.elf (*sdk\_x.y\examples\using\_sntp\bin*) using the Download tool:

1. Launch the Download tool provided with InnoPhase Talaria TWO SDK.
2. In the GUI window:
   1. Boot Target: Select the appropriate EVK from the drop-down
   2. ELF Input: Load the sntp.elf by clicking on Select ELF File.
   3. AP Options: Provide the SSID and Passphrase under AP Options to connect to an Access Point.
   4. Programming: Prog RAM or Prog Flash as per requirement.

For more details on using the Download tool, refer to the document: UG\_Download\_Tool.pdf (path: *sdk\_x.y\pc\_tools\Download\_Tool\doc*).

**Note**: x and y refer to the SDK release version. For example: *sdk\_2.5\doc*.

## Expected Output

On flashing the application using the Download Tool, the console output is as follows:

The application will connect to the AP specified by the SSID and passphrase. Upon successful connection, the latest time is fetched from the NTP server.

|  |
| --- |
| UART:SNWWWWWAEBuild $Id: git-ba65998b7 $  np\_conf\_path=/data/nprofile.json ssid=InnoPhase passphrase=43083191  $App:git-73e7f910  SDK Ver: sdk\_2.5  Using SNTP Application  addr e0:69:3a:00:13:90  Connecting to added network : InnoPhase  [2.401,377] CONNECT:00:5f:67:cd:c5:a6 Channel:11 rssi:-48 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [4.121,170] MYIP 192.168.0.102  [4.121,219] IPv6 [fe80::e269:3aff:fe00:1390]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : InnoPhase  Wifi connected  SNTP: using server: pool.ntp.org  time:6127449  waiting for sntp, times=0:0  waiting for sntp, times=1:0  waiting for sntp, times=2:0  waiting for sntp, times=3:0  waiting for sntp, times=4:0  waiting for sntp, times=5:0  sntp\_process: Thu Jul 7 18:04:17 2022  waiting for sntp, times=6:1657217057  date: Thu Jul 7 18:04:17 202 |