Micriμm

μC/SNTPc

Simple Network Time Protocol (Client)

User's Manual

www.Micrium.com

Table of Contents

1.00 Introduction

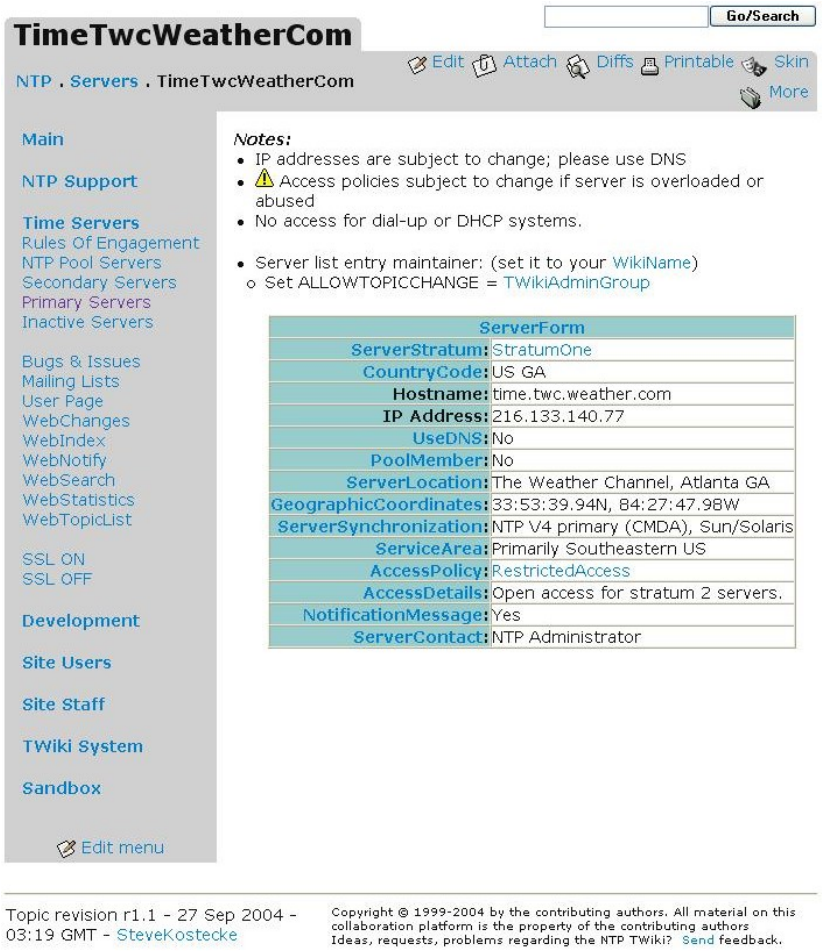
Network Time Protocol (NTP) is a protocol that is used to synchronize computer clock times in a computer network. NTP uses Universal Time Coordinated (UTC) to synchronize computer clock time to the millisecond, and sometimes to a fraction of a millisecond. µC/SNTPc implements RFC 958 (ftp://ftp.rfc-editor.org/in-notes/rfc958.txt).

1.01 Time Servers

Time servers can be found from the following web site:

http://ntp.isc.org/bin/view/Servers/StratumOneTimeServers

This web site list well over 100 time servers to chose from. The time server we used for our tests is:



1.02 Required Modules

This document describes how to configure and use µC/SNTPc. µC/SNTPc assumes the following modules:

µC/TCP-IP or any other TCP/IP stack that offers a BSD v4 socket interface.

µC/OS-II or any RTOS that allows a task to be delayed for a certain amount of time. The OS interface function(s) are isolated in its own file so you can adapt µC/SNTPc to your own RTOS if needed.

µC/SNTPc requires that you provide a function that reads the current 'local time' of your embedded target and supply it as a NTP time stamp (i.e. the number of seconds since 1900). You can purchase µC/CLK from Micriµm which provides this functionality. However, examples of this code is provided with µC/SNTPc (described later).

We used IAR's Embedded Workbench and the CSB337 board to build the examples supplied with the code, but other embedded platforms and tool chains can be used.

Please note that the NTP protocol use 32-bit time stamps which are defined as the number of seconds since 1900. These time stamps will overflow during year 2035. The NTP time stamps are different than the µC/CLK time stamps which are defined as the number of seconds since year 2000 and will overflow during year 2135.

1.03 Directories and Files

µC/SNTPc code and documentation are placed in a directory structure according to "AN-2002, µC/OS-II Directory Structure". Specifically, the files are placed in the following directories:

\Micrium\Software\uC-SNTPc\Doc

This directory contains µC/SNTPc documentation files, including this one.

\Micrium\Software\uC-SNTPc\Source

This directory contains µC/SNTPc source files (`sntp-c.c` and `sntp-c.h`).

\Micrium\Software\EvalBoards\Cogent\CSB337\IAR\uC-Apps\Ex1

This directory is the directory that contains the source code for Example #1 running on a Cogent CSB335 board with IAR tools. This directory contains:

```
app.c
app_cfg.h
clk_cfg.h
Ex1.ewp
Ex1.eww
includes.h
net_cfg.h
os_cfg.h
```

`app.c`	contains startup and example code;
`app_cfg.h`	contains the application configuration file;
`clk_cfg.h`	is the µC/CLK configuration file;
`Ex1.ewp`	IAR project file;
`Ex1.eww`	IAR workbench file;
`includes.h`	is a master include file used by the application;
`net_cfg.h`	is the µC/TCP-IP configuration file;
`os_cfg.h`	is the µC/OS-II configuration file;

2.00 Example code

As mentioned in the previous section, the test code for this board is found in the following directory and will be briefly described:

\Micrium\Software\EvalBoards\Cogent\CSB337\IAR\uC-Apps\Ex1

2.01 Example code, app.c

This file contains the application code for example #1.

app.c is written to demonstrate the capabilities of µC/SNTPc. The code doesn't really do anything useful except initializing µC/OS-II, µC/TCP-IP, µC/DHCPc, µC/CLK and µC/SNTPc, creates a few tasks and other kernel objects that will inform you about the state of the system.

The code, first, initializes and starts µC/OS-II. After, µC/TCP-IP stack is started. Then, µC/DHCPc negociation is started to get IP configuration, SNTP server IP address and SNTP Time Zone Offset from a DHCP server.

Listing 2-1, App_SNTP_Test ()

```
static  void  App_SNTP_Test (void)
{
    CLK_TS_NTP     ntp_time;
[…]
    CLK_DATE_TIME  date_time;
[…]
    NET_IP_ADDR    sntp_srvr;
    SNTP_PKT       sntp_pkt;
    CPU_INT32U     sntp_test;
    CPU_INT08U     buf[25];
    NET_ERR        err;


    Clk_Init();                                                          (1)
    Clk_DateTime_Make (&date_time, 2006, 1, 1, 0, 0, 0, AppClkUTC_Offset);
    Clk_SetDateTime(&date_time);
    Clk_DateTime_ToStr(&date_time, 1, buf);
    APP_DEBUG_TRACE("CLOCK: %s.\n", buf);

[…]

    sntp_test = SNTP_ReqRemoteTime(sntp_srvr, &sntp_pkt);                (2)
    if (sntp_test == DEF_OK) {
        ntp_time = SNTP_GetRemoteTime_s(&sntp_pkt);                      (3)
        Clk_SetTS_NTP(&ntp_time);                                        (4)
    }

    Clk_GetDateTime(&date_time);                                         (5)
    Clk_DateTime_ToStr(&date_time, 1, buf);                              (6)
    APP_DEBUG_TRACE("CLOCK: %s.\n", buf);                                (7)

    while (DEF_YES) {                                                    (8)
        OSTimeDlyHMSM(0, 0, 10, 0);
        Clk_GetDateTime(&date_time);
        Clk_DateTime_ToStr(&date_time, 1, buf);
        APP_DEBUG_TRACE("CLOCK: %s.\n", buf);
    }
```

L2-1(1) We start by initializing the µC/CLK module (assuming you have purchased it). If not, comment out every function starting with "Clk". You will have to provide a function called 'SNTP_GetLocalTime_s()' which get the "local time" and return it into a NTP time stamp. You will have to provide a way to keep the local time.

L2-2(2) Send a SNTP request to a public NTP server. See the following web site for a list of available time servers:

http://ntp.isc.org/bin/view/Servers/StratumOneTimeServers

L2-2(3) Get the local time in a NTP time stamp.

L2-2(4) Set the local time into µC/CLK.

L2-2(5) Extract the local time from µC/CLK into a CLK_DATE_TIME structure. A CLK_DATE_TIME structure is a structure containing fields like year, month, day, hour, minute, seconds.

L2-2(6) Convert the CLK_DATE_TIME structure info a printable format.

L2-2(7) Print the local time to the console.

L2-2(8) Enter into a loop that prints the local time every 10 seconds.

3.00 µC/SNTPc module

SNTP stands for Simple Network Time Protocol. The 'c' means 'client'. Other modules ending with 's', means 'server'. The files are located in:

\Micrium\Software\uC-SNTPc\Source

These files are:

```
sntp-c.h
sntp-c.c
```

3.01 µC/SNTPc module use

The µC/SNTPc example relies on the µC/TCP-IP stack and the µC/CLK module to work. You don't actually need µC/CLK if you can provide the same functionality. As you can see in the example file (see section 2.01).

µC/TCP-IP with µC/SNTPc configuration:
The TCP/IP stack must be configured before SNTPc can be used;

SNTP structure format:
The SNTP structure format is defined in `sntp-c.h`, line 45, from the RFC 958, p. 12.

3.03 µC/SNTPc module limitations

- This SNTP client implements a part of RFC 958.
- The SNTP client support fractional time division, but the µC/CLK module has a precision of one second.

References

µC/OS-II, The Real-Time Kernel, 2nd Edition
Jean J. Labrosse
CMP Books, 2002
ISBN 1-57820-103-9

Embedded Systems Building Blocks
Jean J. Labrosse
CMP Books, 2000
ISBN 0-87930-604-1

Contacts

IAR Systems
Century Plaza
1065 E. Hillsdale Blvd
Foster City, CA 94404
USA
+1 650 287 4250
+1 650 287 4253 (FAX)
e-mail: Info@IAR.com
WEB : www.IAR.com

Cogent Computer Systems, Inc.
1130 Ten Rod Road, Suite A-201
North Kingstown, RI 02852 USA
USA
+1 401 295 6505
+1 401 295 6507 (Fax)
WEB: www.CogComp.com

Micriμm
949 Crestview Circle
Weston, FL 33327
USA
+1 954 217 2036
+1 954 217 2037 (FAX)
e-mail: Jean.Labrosse@Micrium.com
WEB: www.Micrium.com

Validated Software
Lafayette Business Park
2590 Trailridge Drive East, Suite 102
Lafayette, CO 80026
USA
+1 303 531 5290
+1 720 890 4700 (FAX)
e-mail: Sales@ValidatedSoftware.com
WEB: www.ValidatedSoftware.com