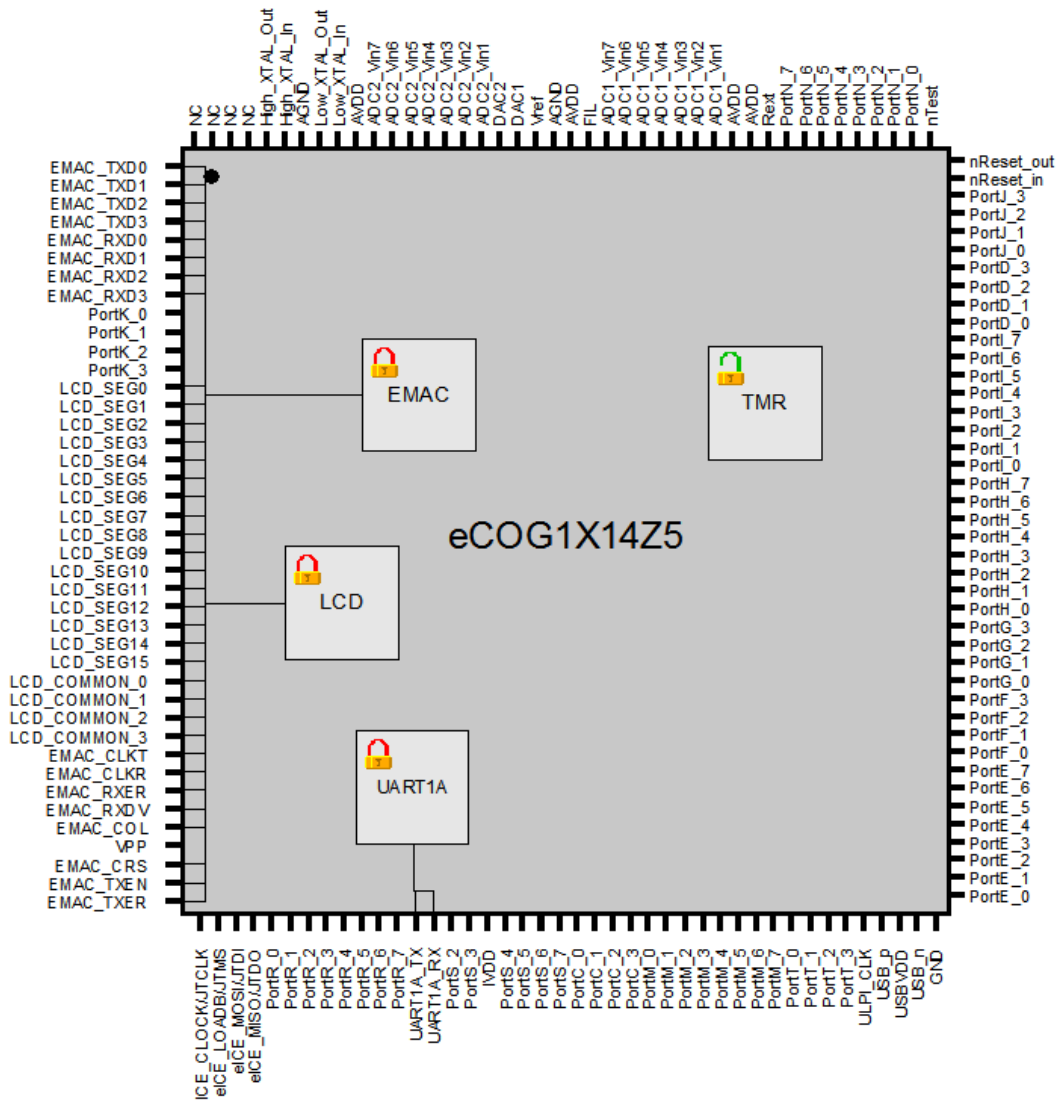




AN070 – Simple Network Time Protocol Client

Version 1.0

The Simple Network Time Protocol allows Network clients to obtain the current time from a central master clock on the network.



Confidential and Proprietary Information

©Cyan Technology Ltd, 2008

This document contains confidential and proprietary information of Cyan Technology Ltd and is protected by copyright laws. Its receipt or possession does not convey any rights to reproduce, manufacture, use or sell anything based on information contained within this document.

Cyan Technology™, the Cyan Technology logo and Max-eICE™ are trademarks of Cyan Holdings Ltd. CyanIDE® and eCOG® are registered trademarks of Cyan Holdings Ltd. Cyan Technology Ltd recognises other brand and product names as trademarks or registered trademarks of their respective holders.

Any product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by Cyan Technology Ltd in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. Cyan Technology Ltd shall not be liable for any loss or damage arising from the use of any information in this guide, any error or omission in such information, or any incorrect use of the product.

This product is not designed or intended to be used for on-line control of aircraft, aircraft navigation or communications systems or in air traffic control applications or in the design, construction, operation or maintenance of any nuclear facility, or for any medical use related to either life support equipment or any other life-critical application. Cyan Technology Ltd specifically disclaims any express or implied warranty of fitness for any or all of such uses. Ask your sales representative for details.



Revision History

Version	Date	Notes
V1.0	06/11/2007	Initial Creation

Contents

1	Introduction	5
2	Glossary	5
3	Requirements	5
4	Requirements	5
5	Using the Software	6
5.1	NTP Server Selection	6
5.2	eCOG1k Development Board	6
5.3	eCOG1X Development Board	6
5.4	Date Conversion	6
Appendix A	API Functions	7
A.1	Application Functions	7
	<i>sntp_sync</i>	7
	<i>sntp_appcall</i>	7
A.2	Callback Functions	7
	<i>sntp_synced</i>	7

1 Introduction

This application note describes the implementation of a Simple Network Time Protocol Client on the eCOG1 microcontrollers, using the uIP TCP/IP Stack Library as described in Application Note AN057.

2 Glossary

eCOG1	Cyan Technology target micro controller
SNTP	Simple Network Time Protocol as defined by RFC 1361

3 Requirements

- PC Running CyanIDE V1.4.2
- An eCOG1k development board (Issue 2) or an eCOG1X development board (Revision B).

The zip archive file named **AN070SW.zip** contains the source code for use with this application note as a pair of example projects, one for the eCOG1k and one for the eCOG1X. This zip file is available for download from the Cyan Technology website www.cyantechnology.com.

4 Requirements

Extract the example software file **AN070SW.zip** to any convenient location, such as the CyanIDE projects directory <C:\Documents and Settings\<username>\My Documents\CyanIDE Projects>. This creates a new directory AN070SW with two further subdirectories. The two directories contain the example projects for the eCOG1k development board V2.1 and for the eCOG1X development board rev B.

5 Using the Software

Two projects are provided for this application note, one for the eCOG1k Development Board (Issue 2) and one for the eCOG1X Development Board (Revision B).

After initialisation, the software uses the uIP DHCP Client to acquire an IP Address, and then it uses the DNS Resolver to acquire the IP address of a public Network Time Protocol (NTP) Server. Once the IP Address of the NTP Server has been acquired, the SNTP Client acquires the current time from that.

Once the current time has been acquired, a second timer is used to maintain the time, which is displayed on the LCD of the Development Board.

5.1 NTP Server Selection

A global pool of NTP servers exists behind the DNS entry "POOL.NTP.ORG". This DNS entry is changed regularly so that the load on the distributed pool of servers around the world is spread evenly. For greater accuracy in different locations, some Continental NTP server pools exist (for example "EUROPE.POOL.NTP.ORG"), as do some other country specific server pools (for example "UK.POOL.NTP.ORG").

5.2 eCOG1k Development Board

The <eCOG1k SNTP> example project is built for the eCOG1k on the eCOG1k Development Board. Once the time has been acquired, it is displayed in the HH:MM:SS format on the alphanumeric LCD.

If a serial terminal is connected to P1 at 9600baud, 8 data bits, no parity and 1 stop bit, then the progress of the software can be seen. When the network time is acquired, the calculated date and time are displayed.

5.3 eCOG1X Development Board

The <eCOG1X SNTP> example project is built for the eCOG1X on the eCOG1X Development Board. Once the time has been acquired, it is displayed in the HH.MM format on the multiplexed LCD.

If a serial terminal is connected to J16 at 9600baud, 8 data bits, no parity and 1 stop bit, then the progress of the software can be seen. When the network time is acquired, the calculated date and time are displayed.

5.4 Date Conversion

The software uses the Julian Date Calculation to convert the NTP time to the current date.

The formula was created by the French philosopher Josephus Justus Scaliger [1540-1609], and allows for the conversion of the Julian Day Number, as used in astronomy, into the Gregorian Calendar date.

For more information on this formula see the website:
<<http://quasar.as.utexas.edu/BillInfo/JulianDatesG.html>>

Appendix A API Functions

A.1 Application Functions

sntp_sync

```
#include "sntp.h"
void sntp_sync(uip_ipaddr_t ipaddr);
```

This function is called to initiate an SNTP Query of the NTP Server at the specified IP address.

sntp_appcall

```
#include "sntp.h"
void sntp_appcall(void);
```

This function is the main uIP application function, and must be called from the `uip_udp_appcall()` function.

A.2 Callback Functions

sntp_synced

```
#include "sntp.h"
void sntp_synced(u32_t *systeme);
```

This function must be provided by the application. It is called by the `sntp_appcall()` function when the SNTP query completes.

If the pointer is NULL, then the query has failed.

If the pointer is not null, then it points to two 32-bit values. The first contains the time in seconds since 00:00 1st January 1900, and the second contains the fractional part of the time.