```
//
//
// CPU_TimeSync for 2500P-ACP1 Version 1.2 (includes DST)
//
//
(*
This Project synchronizes time to NTP time on CTI 2500 Series and Simatic/TI505 CPUs.
It sets time over Ethernet, and requires a 2572 / CP2572 / 2572A / 2572B card.  It will also
work using the built-in Ethernet port on CTI CPUs.  An open connection on port 1505 to the
Ethernet card or CPU must be available.

IMPORTANT - THIS APPLICATION IS PROVIDED "AS-IS" WITH NO WARRANTIES,EXPRESSED OR IMPLIED.


HOW IT WORKS
------------
On the interval NTP_sync_interval, the program queries the NTP server using UDP to get the
current time, then sets the real time clock on the ACP1 from the NTP time.

On the interval CPU_sync_interval, the program sets the clock in each CPU from the list of
IP addresses, using the current clock on the ACP1.  The program will work with up to 49 IP
addresses.  The clock set command to the CPU is done using a TCP connection to port 1505.
The connection is used only long enough to set the clock, then it is closed.

The ACP1 must **NOT** be set to sync to a host CPU, as the ACP1 clock is used for ongoing sync
operations to the list of CPUs.

Because the program (optionally) uses NTP for setting the ACP1 clock, it requires internet
connectivity to communicate with the NTP server.  The default IP address setting for NTP is for
the U.S. Government "time-d.nist.gov" server.  If access to an NTP server is not possible,
then the program can still run in "non-NTP" mode where all CPUs are synchronized to the ACP1
clock.

**IMPORTANT NOTE**
------------------
Time is read from NTP servers as a 32bit number of seconds since Jan 1 1900 00:00:00.
Accordingly, NTP time overflows (and starts back at 0) on Feb 7 2036 at 06:28:15.  This
program does NOT presently handle this overflow.

**IMPORTANT NOTE**
------------------
This program synchronizes to UTC time.  UTC time does not does adjust for daylight savings
time, which is observed in different parts of the world, with different start and
end times.  The parameter UTC_offset can be adjusted to account for DST changes, but will
need to be changed manually twice per year, on the correct date.


SETUP
```

The following variables must be set:

NTP_SyncEnable = TRUE to allow periodic sync of the ACP1 clock to NTP time (default TRUE)
  = FALSE to run in "non-NTP" mode
NTP_SyncInterval = time in seconds between NTP synchronizations of the ACP1 clock
  default 21600 seconds [6hr]
  NOTE: ACP1 worst-case clock drift is 134 seconds per month
NTPServerAddr = IP address for the NTP server to be used (default 129.6.15.27)
UTC_offset = local time offset from UTC time in hours + or - (default -5 is US EST)
User_SecondsAdjustment = allows user to add or subtract seconds to NTP time (default 0)

CPU_SyncEnable = TRUE to allow periodic sync of the CPU list to the ACP1 clock (default TRUE)
CPU>SyncInterval = time in seconds between synchonizations of the entire CPU list (default = ***
  default 86400 seconds [24hr]
  NOTE: CPU worse-case clock drift is 20 seconds per month
CPU_List[2,50] = list of IP addresses and names for the CPUs (up to 50)
number_CPUs = number of CPUs in the IP address list

TROUBLESHOOTING

This program produces two variables with troubleshooting information about NTP and CPU synchonization. In the event of a problem, these are valuable for diagnosing the error.

NTP_SyncLog - this is a structure containing 18 elements with NTP diagnostics for the most recent NTP sync attempt

NTP_SyncLog.IPaddress = ip address of NTP server used
NTP_SyncLog.socket = socket used for the connection
NTP_SyncLog.connection_timeout = TRUE if there was a timeout on the OPEN attempt
NTP_SyncLog.invalid_socket = TRUE if an invalid socket was returned on the OPEN attempt
NTP_SyncLog.successful_connect = TRUE if a successful connection was made
NTP_SyncLog.AddrBufferSuccess = TRUE if the addressbuffer allocation succeeded
NTP_SyncLog.UDPSendSuccess = TRUE if the UDP command was succesfully sent to the NTP server
NTP_SyncLog.NTPQueryString = the query string sent to the NTP server
NTP_SyncLog.updCharsReceived = number of characters returned from the NTP server
NTP_SyncLog.seconds = seconds since Jan 1 1900 00:00:00 calculated from the returned NTP time
NTP_SyncLog.string_seconds = string representing number of seconds above
NTP_SyncLog.SetRTCsuccess = TRUE if the RTC on the ACP1 was successfully set
NTP_SyncLog.successful_close = TRUE if the socket was successfully closed
NTP_SyncLog.close_timeout = TRUE if there was a timeout before the socket closed
NTP_SyncLog.NTPgoodResponse = TRUE if we received a valid response from the NTP server
NTP_SyncLog.receive_timeout = TRUE if we had a timeout before received a response from NTP server
NTP_SyncLog.NTPReceiveString = the string returned from the NTP server
NTP_SyncLog.ValidNTPtime = TRUE if the NTP time returned was valid for use in this program
  (Jan 1 2017 00:00:00 throug Feb 7 2036 06:28:15)

SyncLog - this is a "structure" array where each record (49 total) contains diagnostics for the most recent CPU synchronization attempt. Each record of the array corresponds to a different CPU IPaddress from the IPaddress array.

SyncLog[n].IPaddress = IP address of the CPU where the attempt was made
SyncLog[n].SetTimeString = time set string sent to the CPU
SyncLog[n].ResponseString = response string returned from the CPU
SyncLog[n].CharactersSent = number of characters sent
SyncLog[n].OpenResult = TRUE if the TCP connection was successfully opend
SyncLog[n].CloseResult = TRUE if the TCP connection was successfully closed
SyncLog[n].SendResult = TRUE if the time set command was successfully sent
SyncLog[n].ResponseResult = TRUE if the response coming from the CPU confirmed the time set
SyncLog[n].SocketUsed = socket used for the connection
SyncLog[n].OpenTime = time it took to open the connection (msec)
SyncLog[n].CloseTime = time it took to close the connection (msec)
SyncLog[n].OpenTimeout = TRUE if there was a timeout trying to open the connection
SyncLog[n].CloseTimeout = TRUE if there was a timeout trying to close the connection
SyncLog[n].CharactersReceived = number of characters received in the reply from the CPU
SyncLog[n].InvalidSocket = TRUE if "invalid socket" is returned when opening or closing connection
Normally, this would mean the connection got dropped
SyncLog[n].ResponseTimeout = TRUE if there was a timeout while waiting for a response from the CPU

REVISION HISTORY

REV 1.0 17NOV2017 Initial Release
REV 1.1 14JUN2018 Change CPU sync logic so it syncs the first time the program is run, and then waits CPU_SyncInterval before the next sync. This was done by moving the CPU_SyncInterval check from the top of the SFC to the bottom
REV 1.2 16OCT2018 Started adding DST offset calculation