

The Basics of Setting-up Cyclic Links with CTNet using SyPT Pro

This application note was created to outline the basics of setting up Cyclic Data links to be sent over CTNet using SyptPro. It does assume some experience and knowledge of the target drive be it Unidrive Classic, Mentor II, Quantum III, Unidrive SP or DigitAx ST, their co-processor modules (UD7x, MD29AN, SM-Apps, AppsPlus) and SyPT Pro.

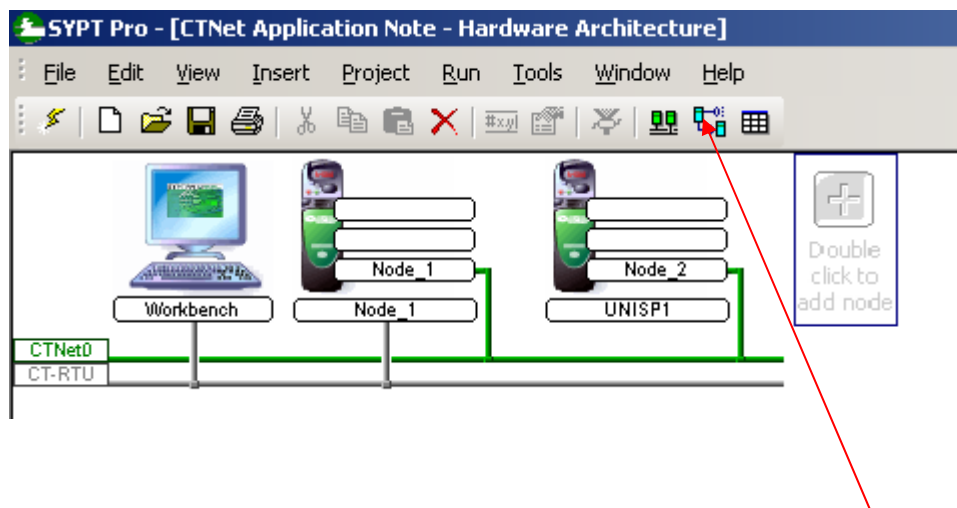
This example will show you how to set-up a CTNet Cyclic Data link between 2 drives. , We will be sending the Speed Reference from Node 1 to Node 2 and since CTNet offers peer-to-peer communications we will be demonstrating that by also sending node 2 status info from Node 2 to Node 1. This will also serve as demonstration on how to utilize data words and bit and associated bit manipulation.

This particular app note and associated program assumes that the drives are setup in either in Closed Loop Vector mode or Servo Mode.

Step 1:

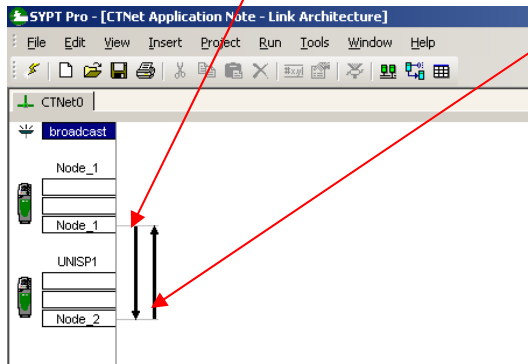
After you have created 2 nodes on the CTNet Network you should have your screen looking like this.

Note: It is not necessary to have the CT-RTU network if you are communicating to the drives via CTNet.

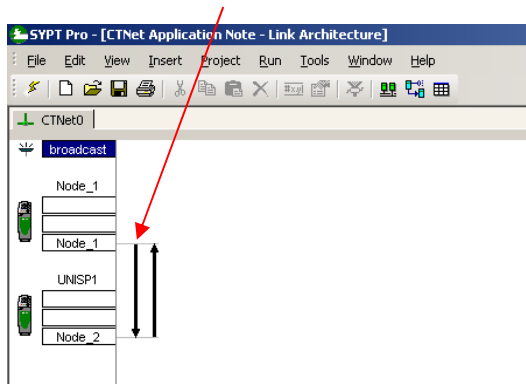


Step 2:

Next we will setup the cyclic data links. To get to this screen below, click on the Links icon. Then if you click on Node 1 you can click and drag a line to Node 2, do the same for Node 2 to Node 1.



If you double click on a particular link of interest, another screen will appear which will allow you to set up your Registers you wish to use for these cyclic transfers. Typically Registers Rxx% are used to transmit and Sxx% are receiver registers.



Step 4:

You may use _R00% - _R99% or #72.00 - #72.99 for your Sending Data, you may use _S00% - _S99% or #73.00 - #73.99 for your Receiving Data.

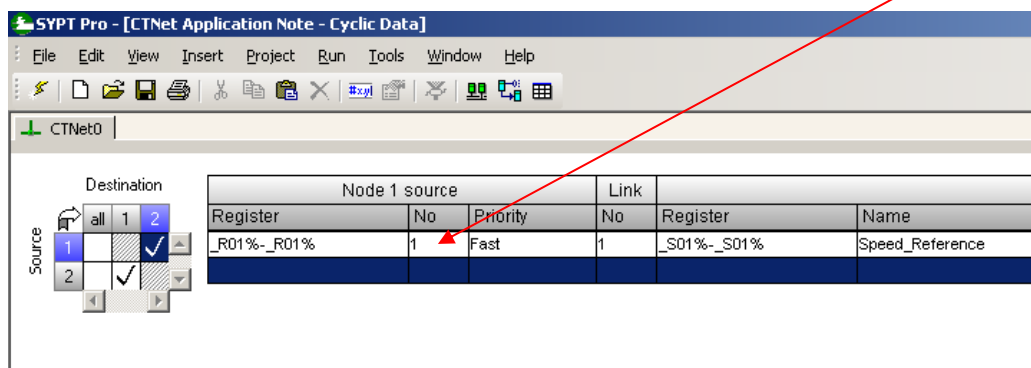
These registers can represent 32 bit signed integers or be used as packed binary data words.

It should be noted that these temporary RAM registers are volatile and are not stored on power down. Registers _P00% - _P99% or #70.00 - #70.99 and _S00% - _S99% or #71.00 - #71.99 can be saved on power down.

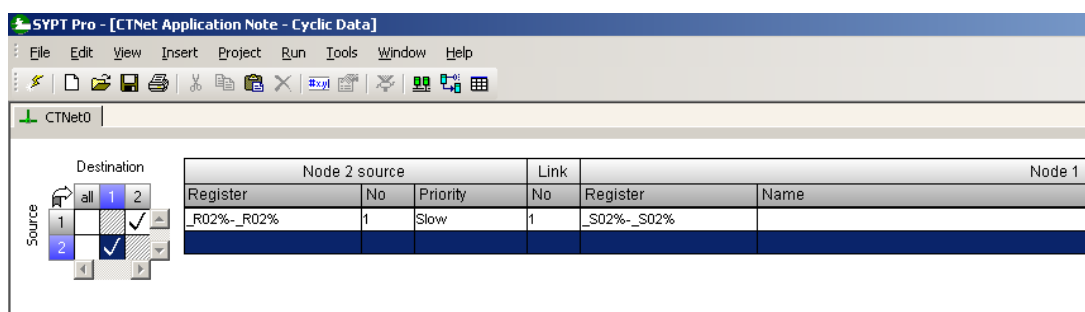
Note: It is recommended not to use _R79% and _S79% since these are reserved for AUTOSYNC.

Step 4 con't

In this example we will use _R01% to send the Speed reference from Node 1 to Node 2, Node 2 will receive this data on _S01% (although it would not have to be _S01% , it could be any of the S registers). In this example, I've elected to send this data using the Fast Cyclic rate. If you have multiple words (drive parameters) that need to go from Node 1 to Node 2 all you need to do is open up the No box (number of registers) and select the quantity you require.



Next you will need to set-up the data link contents going from Node 2 to Node 1 which will be used for the drive status which will be sent over the Slow Cyclic Link. You will need to click on the link from Node 2 to Node 1 as explained in **Step 3**.



Step 5:

Now that the CTNet Cyclic links are setup, we will need to create a program that feeds the _R registers selected with drive data we wish to transfer. For our example this will occur in Node 1 which we will consider the "Master" of the CTNet. In addition, we will need to create some code that picks up the Status Word as sent by Node 2 – that we can consider a Slave or Follower section.

It is recommended to use an Alias task to define the parameters to real words to make the program more understandable/readable. It is also recommended to use the CAPITAL case for these alias definitions as it help one readily distinguish an ALIAS from a variable. Establishing good practices is important when dealing with programming.

It should be noticed in this program that we are only using the first bit (bit 0) of the Status word for the status (_S02%.0). This will allow us to use the other 31 bits if needed in the future without needing to add to the cyclic links.

It should be pointed out here that you do not key in these "Task Templates". SyPT Pro inserts these and you either key in or cut/paste the code between the template extremes. Use the Insert pulldown functions.

"MASTER" PROGRAM NODE 1

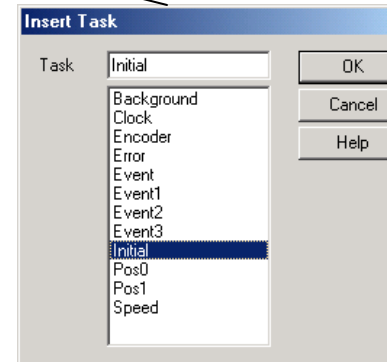
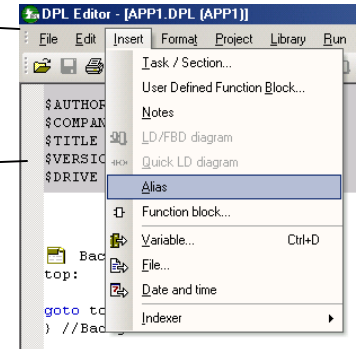
```
Alias{
// Put your list of aliases here
$DEFINE SPEED_REFERENCE%      _R01%
$DEFINE SPEED_FEEDBACK%       #3.02
$DEFINE NODE_2_DRIVE_STATUS%  _S02%.0
$DEFINE NODE_1_DRIVE_STATUS%  #10.01
} //Alias

Initial{
//Waits For absolute encoders to become primed before running
DO WHILE #3.48 = 0
    a% = 1
LOOP

;SM APPLICATION MODULE SETUP
#81.11 = 10      ;Set Clock task to 10ms
#81.12 = 3       ;Position task set to 1ms
#81.14 = 1       ;Global trip enable
#81.16 = 1       ;Update reference and feedback every POS task
#81.17 = 0       ;Disable parameter over range
#81.20 = 1       ;Save PLC registers on power down
#81.21 = 1       ;Save PLC registers on power down
#81.23 = 1       ;Sets CTNet address to 1
#81.24 = 1       ;Sets CTNet to 2.5 mbitSec
#81.25 = 510     ;Sets CTNet Master to Fast = 10ms Slow = 50ms
#81.44 = 3       ;Sets CTSync a higher priority than CTNet
REINIT
} //Initial

Background{
top:
If NODE_1_DRIVE_STATUS% = 0 or NODE_2_DRIVE_STATUS% = 0 then
    Drive_Faulted_Bit% = 1      ;Set Drive Faulted Bit
Else
    Drive_Faulted_Bit% = 0
Endif
goto top: // main background loop
} //Background

Clock{
// Put your section code here
SPEED_REFERENCE% = SPEED_FEEDBACK% * 10    ;Send speed Ref over X10 motif
} //Clock
```



“FOLLOWER” PROGRAM NODE 2

```
Alias{
// Put your list of aliases here
$DEFINE SPEED_REFERENCE%           #1.21
$DEFINE SPEED_FEEDBACK_FROM_NODE1% _S01%
$DEFINE NODE_2_DRIVE_STATUS_CTNET% _R02%.0
$DEFINE NODE_2_DRIVE_STATUS%       #10.01
} //Alias

Initial{
// Put your section code here
//Waits For absolute encoders to become primed before running
DO WHILE #3.48 = 0
    a% = 1
LOOP

;SM APPLICATION MODULE SETUP
#81.11 = 10           ;Set Clock task to 10ms
#81.12 = 3            ;Position task set to 1ms
#81.14 = 1            ;Global trip enable
#81.16 = 1            ;Update reference and feedback every POS task
#81.17 = 0            ;Disable parameter over range
#81.20 = 1            ;Save PLC registers on power down
#81.21 = 1            ;Save PLC registers on power down
#81.23 = 2            ;Sets CTNet address to 2
#81.24 = 1            ;Sets CTNet to 2.5 mbitSec
#81.25 = 0            ;Sets CTNet (This is not the master)
#81.44 = 3            ;Sets CTSync a higher priority than CTNet
REINIT
} //Initial

Background{
top:

NODE_2_DRIVE_STATUS_CTNET% = NODE_2_DRIVE_STATUS%

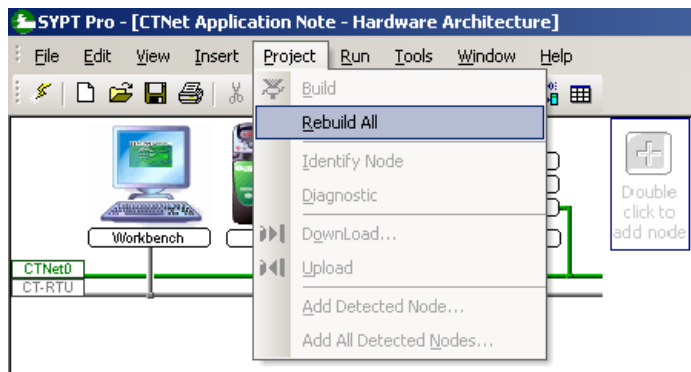
goto top: // main background loop

} //Background

Clock{
// Put your section code here
SPEED_REFERENCE% = Float(SPEED_FEEDBACK_FROM_NODE1%) /10
} //Clock
```

Step 6:

Next we will need to do a compile this program by clicking on REBUILD ALL from the main screen as shown below. This will compile your programs for each node and check it for basic syntax errors etc.



Step 7:

If no Errors are found you can then download the resultant files to the drives. However, we will need to set the basic CTNet setup few parameters in the drives.

Master SM-Apps CTNet setup

#81.23 = 1 ;Sets CTNet address to 1
#81.24 = 1 ;Sets CTNet to 2.5 mbitSec
#81.25 = 510 ;Sets CTNet Master to Fast = 10ms Slow = 50ms

Slave SM-Apps CTNet setup

#81.23 = 2 ;Sets CTNet address to 2
#81.24 = 1 ;Sets CTNet to 2.5 mbitSec
#81.25 = 0 ;Sets CTNet (This is not the master)

Note: In the above example, we are setting up the SM-Apps module CTNet parameters that exist in the co-processor in Menu 81.

- If a **single** SM-Apps is being used in the drive, you may address the CTNet setting using menu 81 implicitly regardless of which slot the SM-Apps is placed.
- If **multiple** SM-Apps are used with 1 drive, they must be accessed explicitly via the slot menu 15,16 or 17 representing slot 1,2 or 3 respectively.

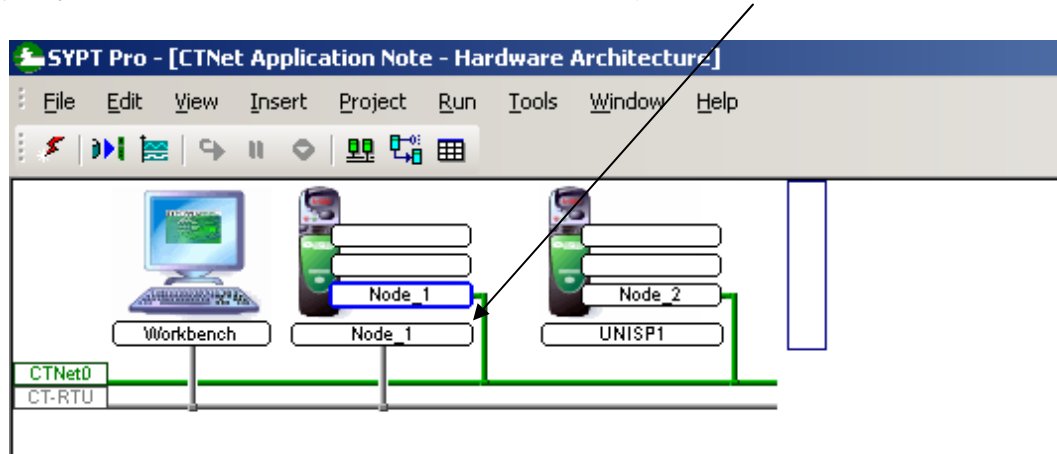
Step 8:

Next you will want to Save Parameters in the Drive **and** reset the application module.

- To Save enter in 1000 in xx.00 and press the **Red Reset** button.
- To Reset Application module enter in 1070 in xx.00 and press the **Red Reset** button.

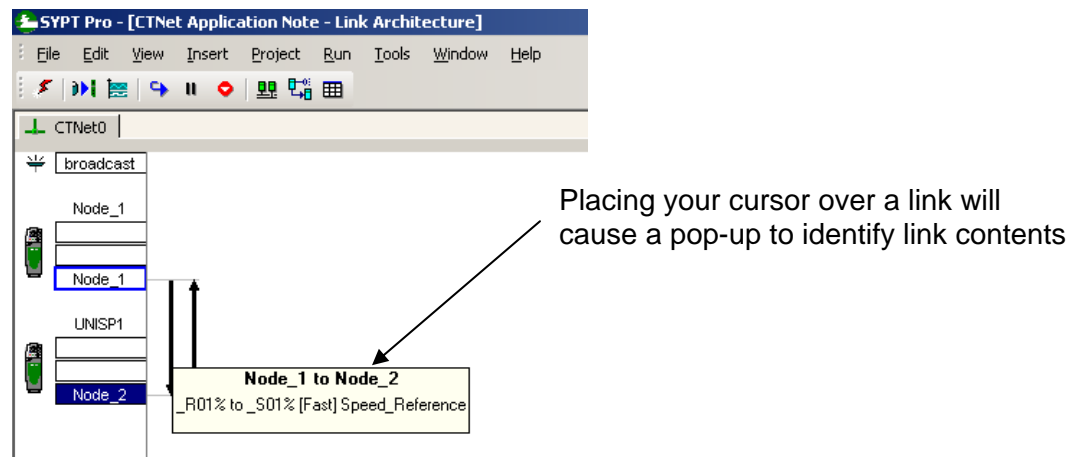
Step 9:

Now you are ready to go on-line and download your project. After you have downloaded the project you should see a blue circle around Node 1 (this means it is the CTNet Master)



Step 10:

Next go to your links and verify that there are no Red lines on your cyclic data. Red lines on Cyclic links indicates there is an error. F5 will refresh the cyclic links. One can go to Project and select Diagnostic and it will show the Diagnostics for the node as well as the CTNet Diagnostics for the Network.



Step 11:

Now you are ready to run and test your program. If you run your master drive you should see #3.02 (Speed Feedback) passed over to #1.21 (Preset #1) on your Follower Drive. Also, you can see the drive status of Node 2, #10.01 (Drive Healthy) being passed over to Node 1 on NODE_2_DRIVE_STATUS_CTNET%.

It may be helpful to set up a watch window to view all at once. If you want to see the Slave drive follow the Master reference that you see on Node 2 #1.21 you will need to set the slave appropriately to select Preset references (#1.14 = 3) or set #0.05=Pr.

Other References:

[Basic CTNet Practices and Checkout](#)

[CTNet User Guide](#)

[Quick Setup Guide w/CTNet](#)

[CTNet Troubleshooting Guide](#)

[CTNet Revision FAQ's](#)

[Option Module Trip Codes](#)



Author: Tim Micklich

e-mail : <mailto:tim.micklich@emerson.com>