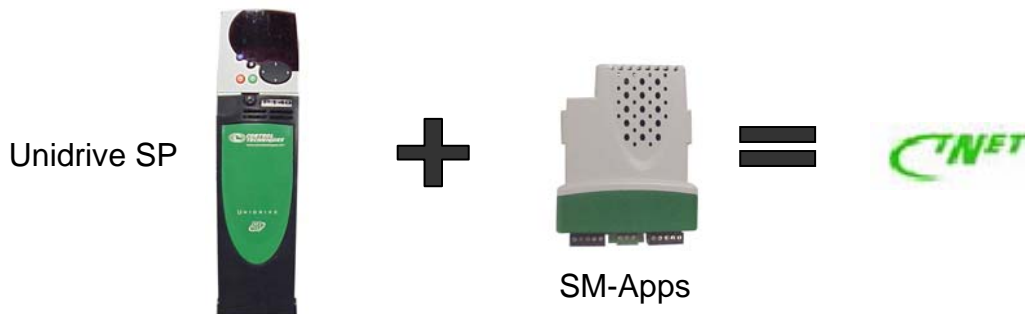


UnidriveSP CTNET Quick Setup

This document pertains to Unidrive SP's using the SM Applications with CTNET



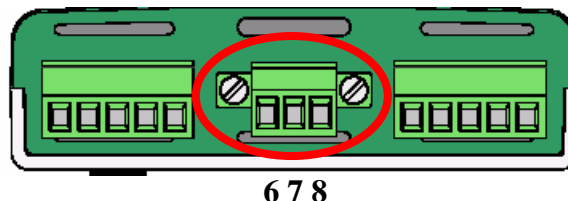
The ability to send data on a peer-to-peer network can be easily established using CTNET, which is available on the SM-Applications Module. This is highly valuable because there is no master (scanner) needed to direct communications. Cyclic and Non-Cyclic data can be transferred. DPL code will be needed to accomplish Non-Cyclic data transmissions. Cyclic data can be transferred using DPL code or simply utilizing “**Easy Mode**”.

Definitions:

Cyclic Data: A method of automatic real-time data transfer.

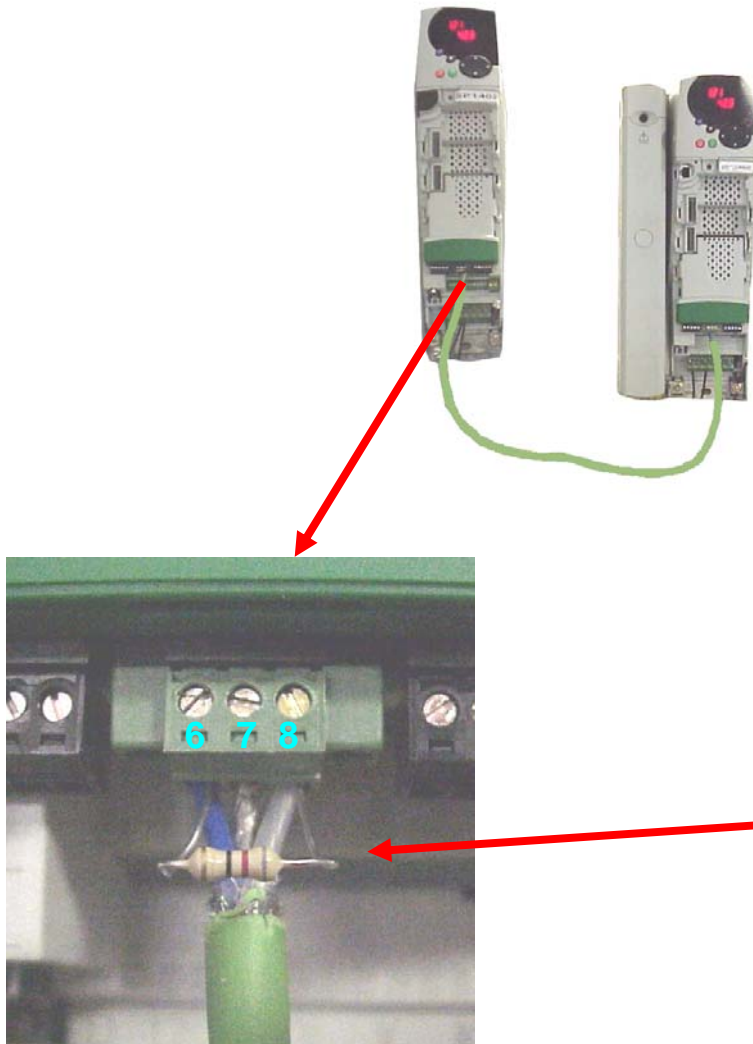
Non-Cyclic Data: A method of scheduled data transferred in the form of requests and responses.

SM-Applications Wiring Diagram for CTNET



Pin 6 = CTNet A
Pin 7 = CTNet Shield
Pin 8 = CTNet B

Standard Wiring for CTNet



Terminating Resistor fitted
at the end of each cable.
Do not fit if two CTNet wires
are connected to the drive.

82 Ohm
1/4 Watt 1%

CTNet Cable

When using Control Techniques UD75, MD29AN, SM-Applications coprocessors and associated CTNet options, it is mandatory that this cable be utilized for proper network operation.

P/N CTNet-XXX (XXX in feet)

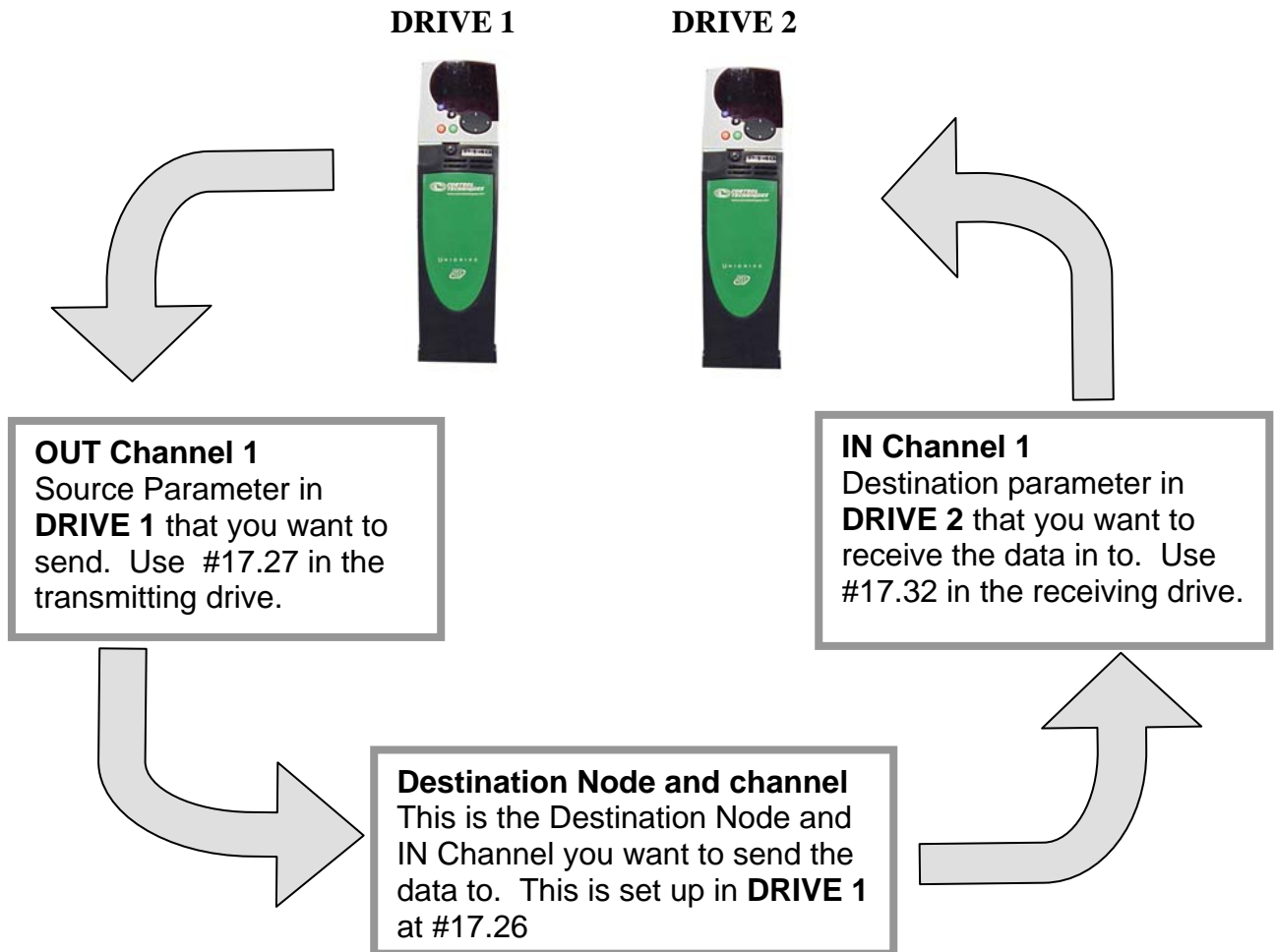


CALL 1-800-367-8067 to purchase CTNet Cable

The entire CTNet manual can be downloaded from www.emersonct.com. You can obtain information about the standard wiring practices needed to obtain reliable data transfer. CTNet cable must be used with all CTNet nodes.

Cyclic Data Transfer “Easy Mode”

It is very common for one drive to follow the speed/torque of another drive in the same application. “Easy Mode” Data transfer allows for this to be set up easily. The flow of a registers data from one drive to another is shown below.



Easy Mode Example Cyclic

To use Cyclic data transfer to send #1.21 from Drive 1 to #1.21 in Drive 2 the following parameters would need to be set. SM-Applications is installed in slot 3 in both drives. Make the changes in the order that they appear below.

#17.23 = Drive 1 = 1	CTNet node address
#17.23 = Drive 2 = 2	CTNet node address
#17.24 = Drive 1 and 2 = 1	Sets the baud rate for 2.5Mbit/s
#17.25 = Drive 1 = 1005 Drive 2 = 0	Sets fast cyclic data rate to 5ms and slow to 50ms
#17.27 = Drive 1 = 121	Source parameter in drive 1
#17.26 = Drive 1 = 201	Destination node and channel
#17.32 = Drive 2 = 121	Destination parameter in drive 2
#17.19 = Drive 1 and 2 = ON	Activates the changes then self resets
#17.00 = Drive 1 and 2 = 1000 then reset	Saves all changes.

**Check #17.36 is showing around 400 (messages/sec).
This indicates the network is active.**

When the above changes are made #1.21 in Drive 2 will follow any changes made to #1.21 in Drive 1. This set up can be repeated for a total of 3 IN Channels and 3 OUT Channels for each node on the CTNetwork.

Non-Cyclic Data Transfer

DPL programming is necessary to achieve non-cyclic data transmissions. This section will show how the RDNET and NETREPLY function blocks are used. The example will read a register from a remote node and transfer it to a local address.

Definitions:

RDNET: This function sends a request to read a single parameter from another node over CTNet.

NETREPLY: This function returns the value of a CTNet read operation (RDNET or RDNETB). It also returns the CTNet drivers to an idle state in preparation for another read or write request.

The code below can be compiled and loaded into the SM-Applications module in Drive #1. It will read #1.22 from Drive 2 and store the data in Drive 1 at #18.11. Set the parameters below and load the code to activate the non-cyclic data request.

#17.23 = Drive 1 = 1	CTNet node address
#17.23 = Drive 2 = 2	CTNet node address
#17.24 = Drive 1 and 2 = 1	Sets the baud rate for 2.5Mbit/s
#17.25 = Drive 1 = 1005	Sets fast cyclic data rate to 5ms
Drive 2 = 0	and slow to 50ms
#17.19 = Drive 1 and 2 = ON	Activates the changes then self resets
#17.00 = Drive 1 and 2 = 1000 then reset	Saves all changes.

**Check #17.36 is showing around 400 (messages/sec).
This indicates the network is active.**

```
$AUTHOR Jeff Wilcox
$COMPANY Control Techniques
$TITLE Non-Cyclic Data example
$VERSION 1.0
$DRIVE UNIDRVSP
```

```
Initial {
status% = 0
invalue% = 0
$define NET_OK 1
}
```

```
Background{
top:
```

```
Status% = RDNET( 2, 01, 22, 100 ) // Read parameter #01.22 from drive (with
//CTNet node address) 2
// 2 = destination drive ID
// 01 = menu number on destination drive
// 22 = parameter number on destination
//drive
// 100 = timeout in msec
// Did we get a response or time out?
```

```
if Status% = NET_OK then
```

```
invalue% = NETREPLY(0) // Read the incoming value
#18.11 = invalue% // Make #18.11 in Drive 1 = #1.22 in Drive 2
```

```
else
```

```
Status% = NETREPLY(0) // Timed out, force the CTNet drivers to an
//idle state
```

```
endif
```

```
goto top: // main background loop
} //Background
```



For questions call Control Techniques Technical Support at
716-774-1193 Grand Island, NY 14072