

**CTNet**

**X-Link / CTNet  
Device Driver**

**User's Guide**

The contents of this guide are believed to be correct at the time of printing. In the interests of a commitment to a policy of continuous development and improvement, the manufacturer reserves the right to change the specification of the product or its performance or the contents of the User's Guide without notice.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical including photocopying, recording or by any information storage or retrieval system without permission from the publisher in writing.

Copyright 1999 Control Techniques plc

Part No. 0410-0000

Issue 2

September 1999

# CONTENTS

- 1. Introduction
- 2. Quick Start
  - 2.1 Powering the X-Link
  - 2.2 Installing the X-Link Battery
  - 2.3 Installing the X-Link Network Cards
  - 2.4 Connecting the Network Cards into their Respective Networks
  - 2.5 Resetting the X-Link Engine
  - 2.6 Starting the Configuration Program
  - 2.7 Loading the Kernel
  - 2.8 Loading the Allen-Bradley Data Highway Plus Driver
  - 2.9 Loading the Control Techniques CTNet Driver
  - 2.10 Loading the Mappings
- 3. Mapping Fundamentals
  - 3.1 Selecting the Mapping Source
    - 3.1.1 Allen-Bradley SD-DHP PLC Write
    - 3.1.2 Allen-Bradley SD-DHP X-Link Read
    - 3.1.3 CTNet Node Write
    - 3.1.4 CTNet X-Link Read
  - 3.2 Selecting the Mapping Destination
    - 3.2.1 CTNet X-Link Write
    - 3.2.2 CTNet Node Read
    - 3.2.3 SD-DHP PLC Read
    - 3.2.4 SD-DHP X-Link Write
- 4. Mapping Strategy
  - 4.1 Unsolicited Transactions are Faster
  - 4.2 All X-Link Transactions are Queued
  - 4.3 The X-Link Box can do 91 Transactions / sec
  - 4.4 The Danger of Setting Polled Mappings to Zero Poll Rate
  - 4.5 Staggered Polling Allows Important Data to Get Through
  - 4.6 Polling can Interfere with Unsolicited Requests
  - 4.7 Mapping Strategies to Try
- 5. Advanced Concepts
  - 5.1 Multiple Destinations
  - 5.2 Floating Point Transfers
  - 5.3 Status Counters
  - 5.4 X-Link Real Time Values Display
- AP 1 Sample Application

# 1. INTRODUCTION

CTNet is a collection of hardware and software modules that permit Control Techniques customers to fashion sophisticated real-time distributed control applications. With inexpensive twisted-pair cables, 5.0 Mbit/sec transmission speed, easy interface to the popular Intellution and Wonderware SCADA systems, internal coprocessor modules, an inexpensive remote I/O module and a ready source of hubs and repeaters, the Control Techniques CTNet solution is an excellent choice for tomorrow's factory automation applications.

The need may arise for a CTNet system to interoperate with equipment on another industrial network, such as Allen-Bradley's Data Highway Plus or Modicon's Modbus Plus. In this situation, the X-Link Bridge from SST may be used to connect CTNet systems to another industrial network. A X-Link bridge is a 386-based single board computer that accepts ISA-style industry standard interface boards and uses a suite of software drivers to map data from one network into another. Figure 1 below shows a typical CTNet system using an X-link bridge to interface to a Data Highway Plus system.

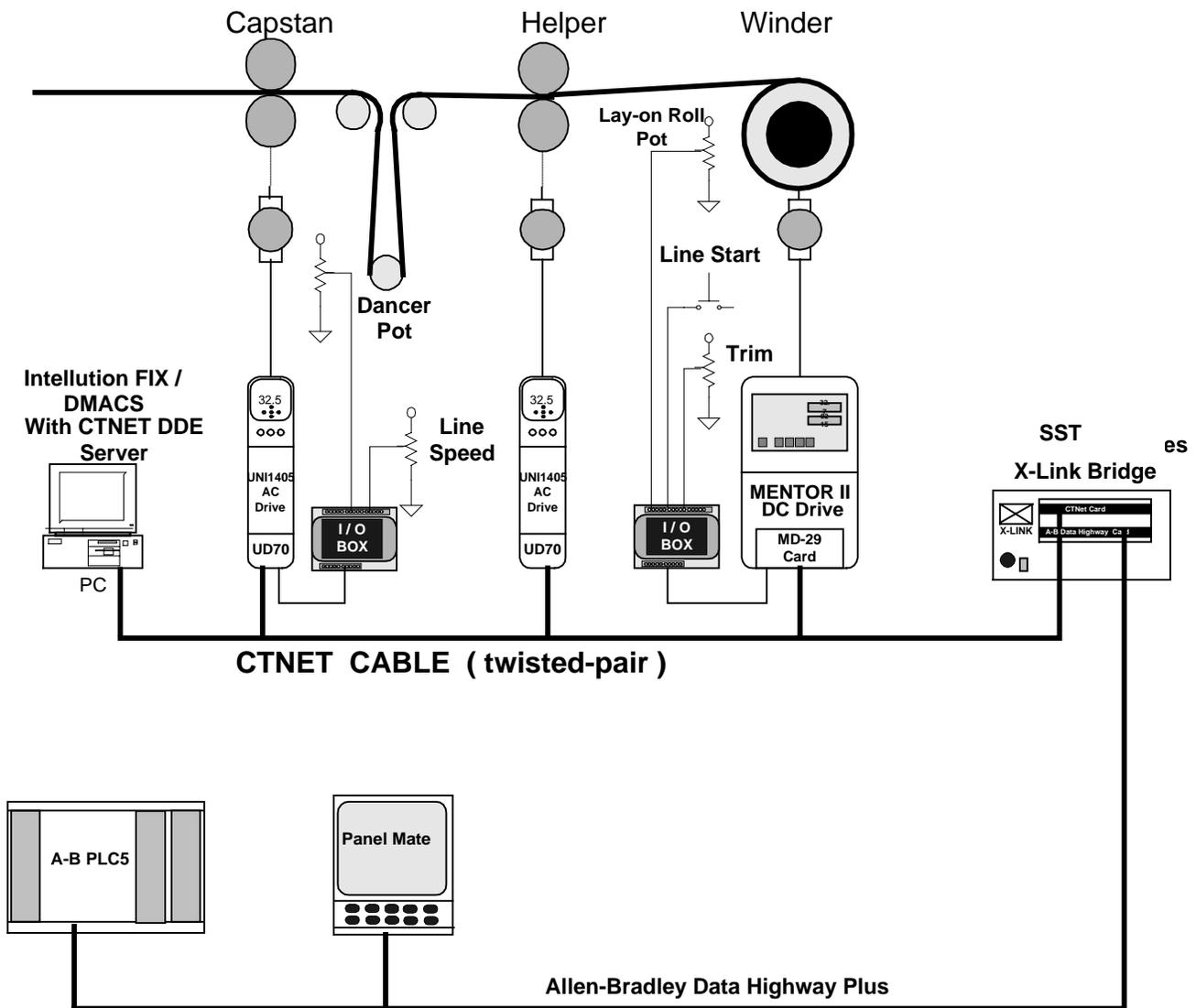


Figure 1. Typical CTNet System

Currently, the S-S Technologies X-Link supports the following PLC standards:

- Allen-Bradley Data Highway / Data Highway Plus
- Allen-Bradley DF1
- Allen-Bradley TCP/IP
- Allen-Bradley RIO
- Allen-Bradley ControlNet
- Control Techniques CTNet
- Cooper Industries Quad2000
- Echelon Lonworks
- GEC Alsthom Gemstart
- GE Fanuc Genius
- MagneTek Microtrac
- Modicon Modbus
- Modicon Modbus Plus
- Phoenix Contact Interbus-S
- Profibus-DP
- Reliance Automate R-NET
- Reliance Automax DCS
- Siemens L1
- Siemens Simatic TIWAY network
- Square D SY/LINK
- Westinghouse PC100
- others coming

Creating an X-Link bridge is straightforward. The X-Link box, called the X-Link Gateway Engine, has two slots to accept a CTNet ISA adapter card and a network ISA card for one of the above industrial networks. If the Allen-Bradley Data Highway is the choice, then either an Allen-Bradley KT card or the SST 5136-SD card could be fitted. Then the X-Link's "Config" port can be connected to a laptop or other PC for installation of the resident kernel software and network drivers. An onboard lithium battery gives the loaded kernel, drivers and configuration data four years of backup if power is turned off.

A SST configuration program is used to first load the kernel then the two drivers (CTNet driver and A-B Data Highway Plus driver). Then the user must create a "mapping" between parameters on the CTNet network and device registers in the Allen-Bradley PLC. Up to 1000 items of data can be mapped between the two networks. The configuration program allows you interactively to specify a register in the PLC and associate it with one or more destination parameters in the CTNet network and vice versa.

When the mappings have been satisfactorily entered, the configuration is saved in the X-Link's battery-backed RAM and the system executes as a "black box" from that point onward.

The configuration program, all drivers (including the CTNet driver) and documentation can be downloaded from the SST web site: **[www.sstech.on.ca](http://www.sstech.on.ca)**

## 2. QUICK START

For our customers anxious to get a rapid start, the following section gives a simple example of interfacing an Allen-Bradley PLC-5 using Data Highway Plus to a CTNet network. A device register on the A-B PLC-5 is interfaced to a parameter on the CTNet network.

### 2.1 Powering the X-Link

The X-Link runs off 24 volts AC or DC. A simple AC step-down transformer is most convenient, but a 24 VDC power supply can also be used. In locales with frequent lightening strikes, a DC power supply may provide better transient protection. The power cable included with the short and long X-Link units has a three pin connector (Molex Part Number 26-03-4030). Pin 1 is the bottom pin on the POWER connector. When powering the X-Link with DC power, connect positive to pin 1 and ground to pin 2. When powering with AC, connect AC to pins 1 and 3 and connect ground to pin 2.

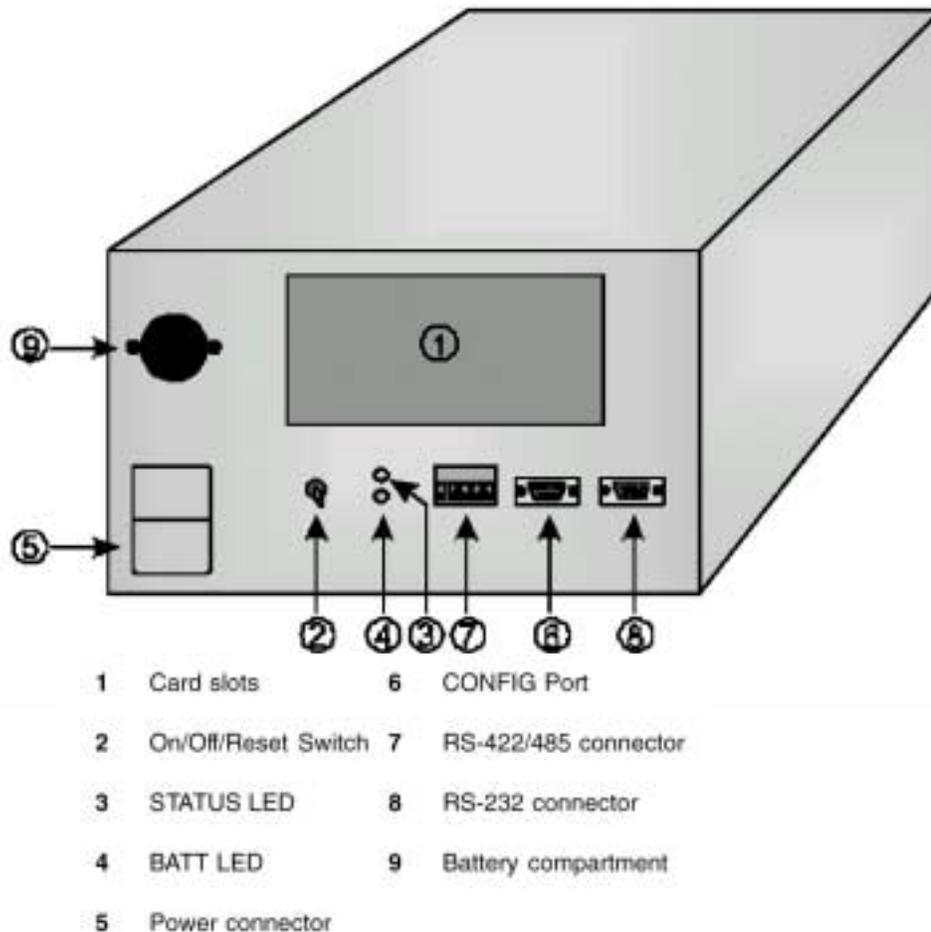


Figure 2. Powering the X-Link Box

## 2.2 Installing the X-Link Battery

The X-Link uses a single 3.6-v AA lithium battery to hold the configuration during a power failure. Open the black battery cover on the lower left and remove the old battery. A coin is quite useful in getting the battery cover on and off. The new lithium battery (TADIRAN TL-5104 or equivalent) should be installed with the positive terminal towards the outside. If the battery is installed improperly, it will not damage anything but the battery-backup function will not operate. There is a “BATT” LED on the X-Link, if it is “green” then the battery is installed properly and still charged. A “red” indicator suggests improper installation or that the battery is exhausted and needs replacing.

## 2.3 Installing the X-Link Network Cards

The X-Link box can accept two ISA industrial network cards. Remove the X-Link cover by loosening the four knurled thumbscrews on the top and side of the case. Remove the two dummy filler brackets. **These operations must be done with the power off!**

Installation of the network cards is similar to installation of any PC adapter card into a desktop computer. The card is inserted carefully into the ISA connector and a hold-down screw is installed to lock the card's mounting bracket to the case.

Each network card should be handled with care since electrostatic discharges can damage the card. Although a grounded wrist strap is best, the simple precaution of grabbing a good ground first to drain any bodily electrostatic potential will lessen the risk.

Each network card will have jumpers and DIP switches to set before installation. Typically the card will allow you to specify the card's base address, IRQ setting, network address, etc. It's wise, however, to jot down these down since many of the X-Link drivers will require you to enter these settings during the driver software configuration step.

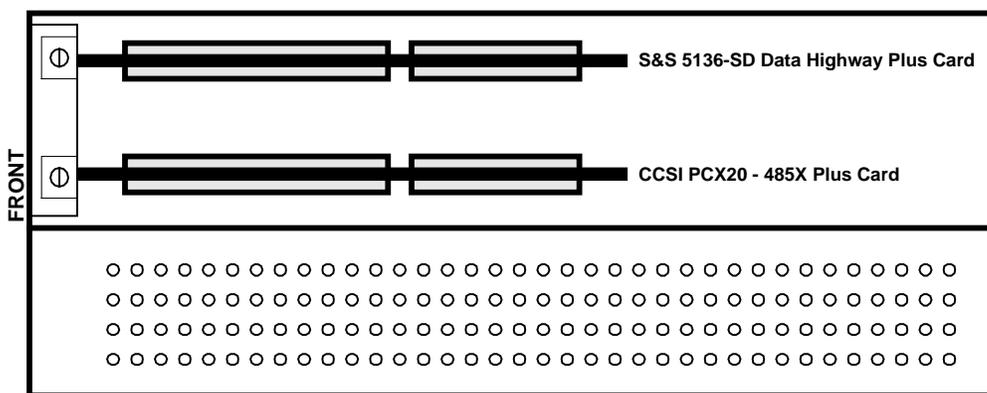


Figure 3. Installing the Network Adapter Cards in the X-Link Box

## 2.4 Connecting the Network Cards into their Respective Networks

The Allen Bradley Data Highway Plus network uses a shielded, twisted-pair cable commonly referred to as the “blue hose”. This cable is attached to pins 4 and 6 of the Data Highway connector (shield to pin 5) on the S-S Technologies 5136-SD card with a  $150\Omega$  resistor added (across pins 1 and 3) if the node is the last one in the trunk. Likewise, the “blue hose” is attached to pins 1 and 3 of the Data Highway connector on the PLC-5 (the shield goes to pin 2) with a  $150\Omega$  terminating resistor attached across pins 1 and 3 if this node is at the other end of the trunk.

The CTNet network uses a special high data rate, low loss shielded, twisted pair cable that is available from the Systems and Software Products division – SSPD. We call our cable the “green hose”. The data conductors are attached to either the left outer pair (pins 1 and 2) or the right outer pair (pins 4 and 5) of the green CTNet connector. CTNet is non-phasic and it doesn’t matter how the data conductors are oriented. The shield is always attached to the center terminal (pin 3). If the node is situated at either end of the trunk line, then a  $78\Omega$  terminating resistor must be fitted across pins 1 and 2 or pins 4 and 5.

Figure 4 below illustrates the interconnection wiring between an Allen-Bradley PLC-5 and a Control Techniques drive system using the X-Link system as a bridge.

## 2.5 Resetting the X-Link Engine

The X-LINK engine may be “hard” reset by depressing the recessed RESET button for ten to twenty seconds until the status LED has stopped flashing and assumes a solid yellow color. This completely resets the X-LINK and establishes 38400 as the baud rate. This also prevents the kernel from starting, therefore you must download a new copy of the X-Link kernel.

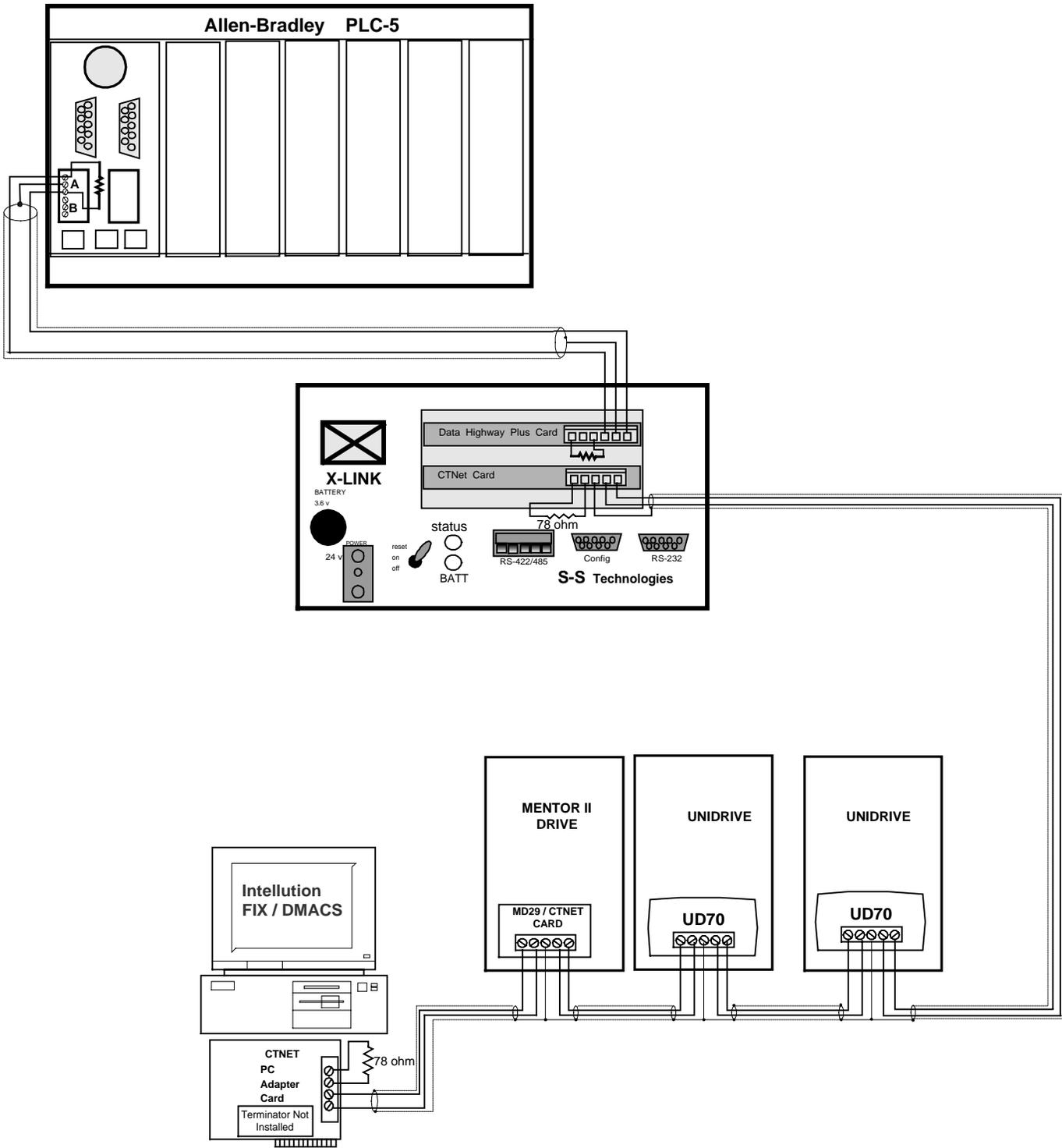
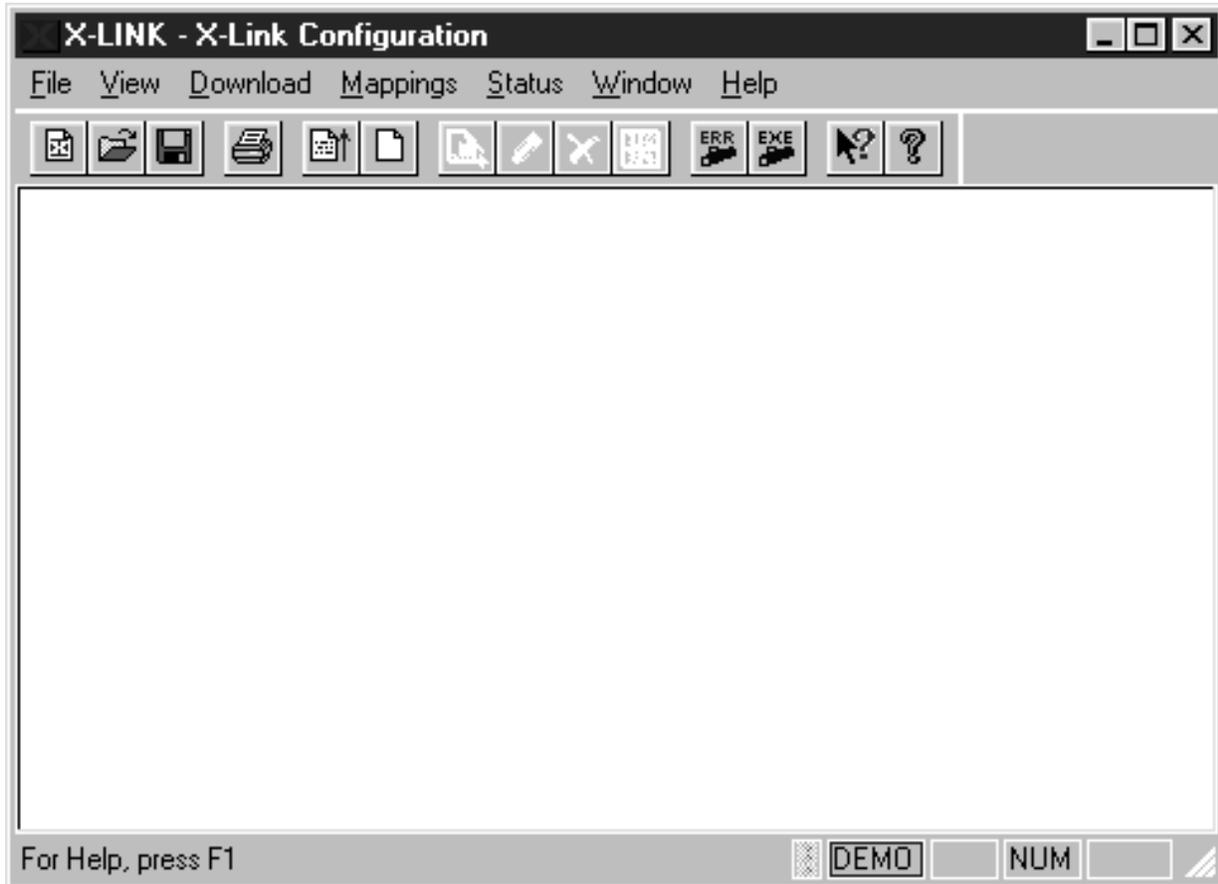


Figure 4. Typical Network Connections using the X-Link Box

## 2.6 Starting the X-LINK Configuration Program

Copy the X-LINK distribution disk to a C:\XLINK directory on your hard drive. The X-LINK engine can be configured by a Windows 95 program called CONFIG.EXE. You can execute it directly from the WIN95 RUN menu or via a shortcut.

Once started, the X-LINK configuration program will display the following screen:

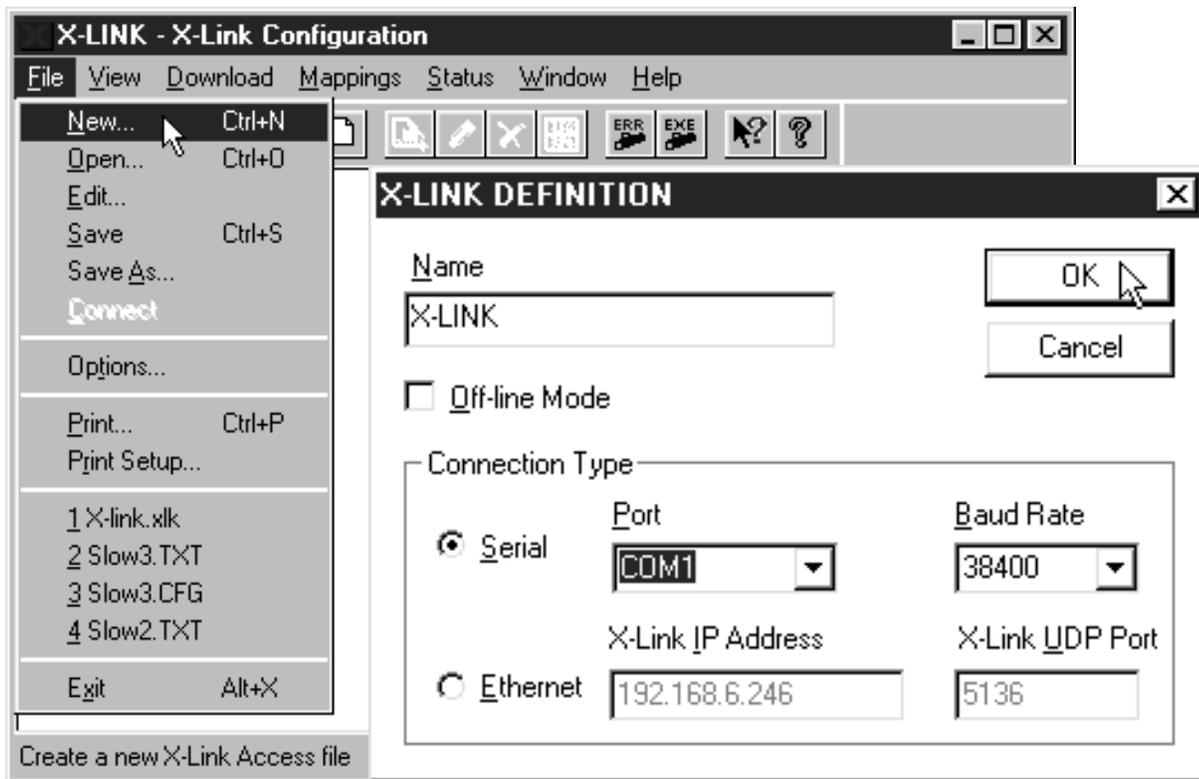


**Figure 5. Initial X-Link Configuration Screen**

Before attempting to load the kernel and the drivers, the configuration program should have its baud rate and COM port specified. These default to COM1 and 38400 baud. If this needs to be changed, then selecting the X-Link menu followed by NEW will give the following screen shown in Figure 6 below.

The screen in Figure 6 will allow setting of several baud rates and COM ports. The “Name” field creates an X-Link “access” file, which is recalled the next time you run the program so that you won’t have to enter this information repeatedly. If you have multiple X-Links in an application, it’s wise to create different access files for each unit. Afterwards, just use File-Open to select the proper access file when connecting to a specific X-Link engine.

The X-Link engine uses a 9-pin, female-to-female “null modem” cable to connect from the PC’s COM port to the engine’s “config” port. Only three wires of this cable are used (pins 2 and 3 for transmit and receive and pin 5 for ground) with the proviso that the interconnecting cable swaps pins 2 and 3. All X-Link engines are sold with an appropriate config cable supplied.



**Figure 6. Specifying the X-Link COM Port and Baud Rate**

X-Link systems require the user to download four files to the engine, they are:

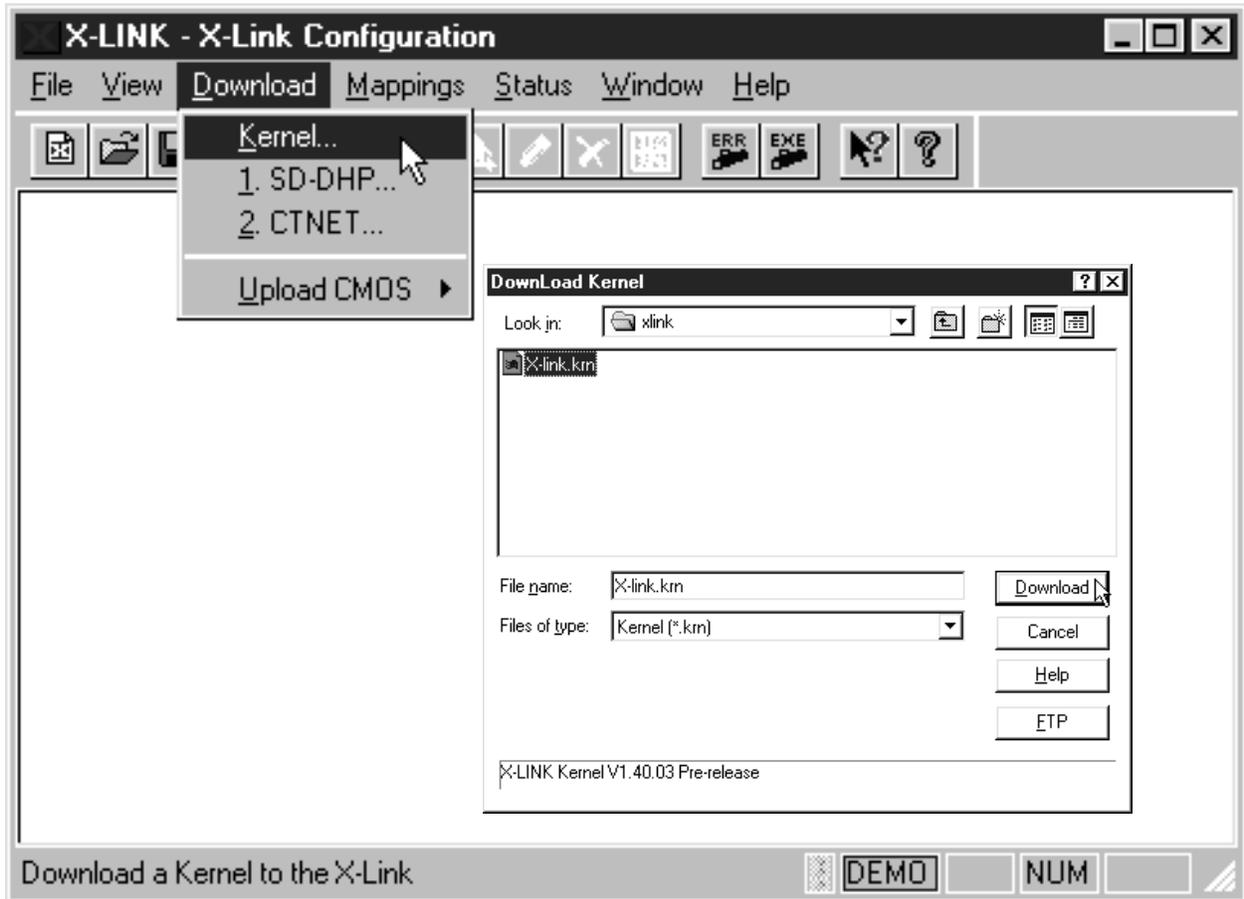
- **Kernel**
- **Driver #1**
- **Driver #2**
- **Mappings**

The Kernel is the SST core software that forms the real-time operating system. The two drivers would be, for this example, the CTNet driver and the SD-DHP driver for the Allen-Bradley Data Highway Plus. The mappings are a database the user creates establishing a link between registers on the Data Highway Plus network and parameters within the CTNet network.

Since we used the “master RESET” operation earlier, we must load each of the above four files. If any of these components are already loaded, the config program will flag this and ask you if you want to reload.

## 2.7 Loading the Kernel

To download the X-Link kernel, pull down the download menu and select “Kernel”.



**Figure 7. Selecting an X-Link Kernel to Download**

After clicking on the selected kernel, in this case **X-LINK.KRN**, the configuration program will take about a minute to download the kernel. It will then open up another dialog box, shown in Figure 8 below, that requires you to specify the Baud Rate used by the kernel and if mapping names are case-sensitive. You might think that this is redundant, but the initial configuration specifies what baud rate the CONFIG program runs at while the download dialog box specifies what baud rate the X-Link engine runs at. These baud rates must match, of course.

A feature that you should avoid selecting is “Timers in Tenths of a Second”. This means that polling rates will be measured in 100 msec. Unfortunately, this option consumes too much CPU time and actually makes the X-Link run slower!

After selecting “OK” in the dialog box shown in Figure 8, the Status LED will blink yellow at a 1 Hz rate to indicate that the kernel has been successfully downloaded.

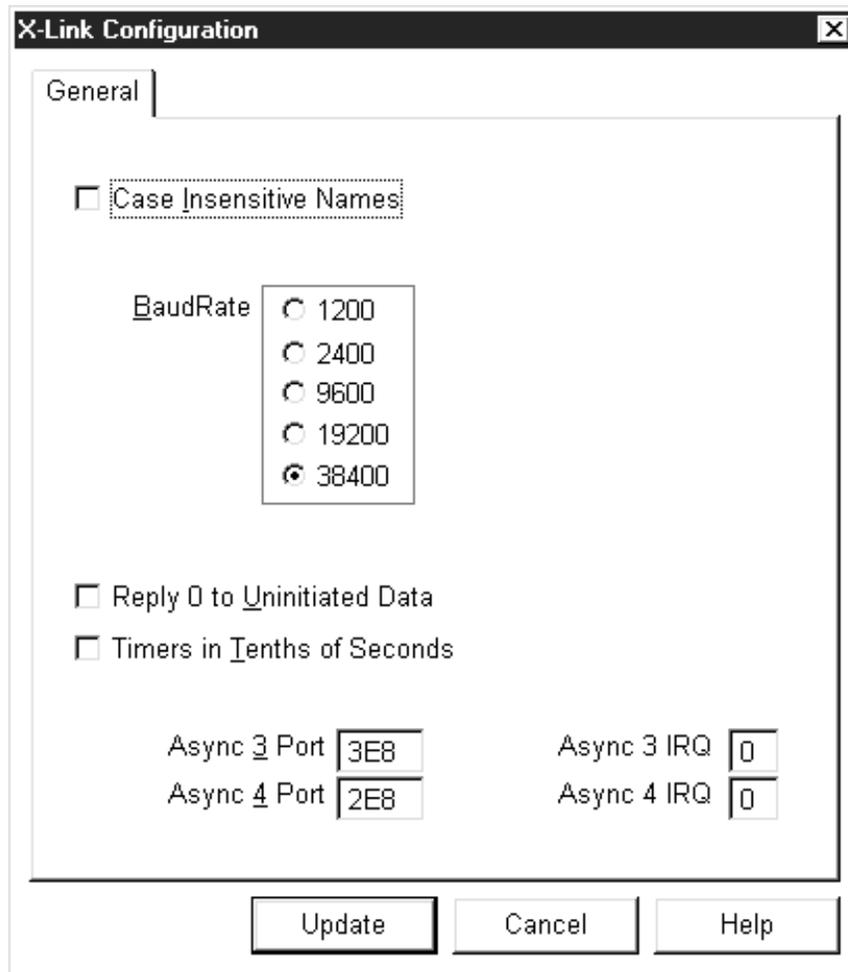
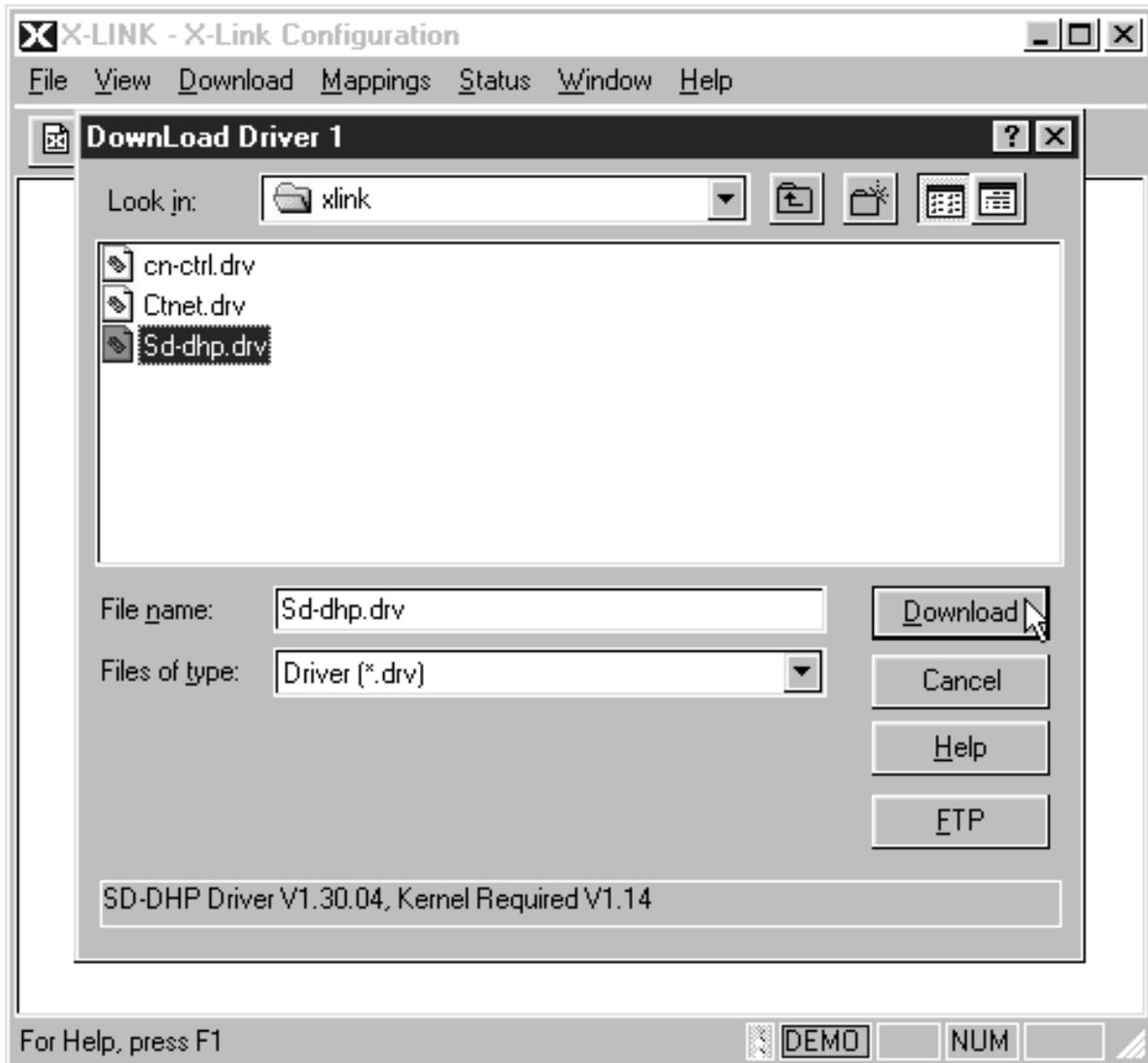


Figure 8. Selecting an X-Link Kernel Baud Rate

## 2.8 Loading the Allen-Bradley Data Highway Plus Driver

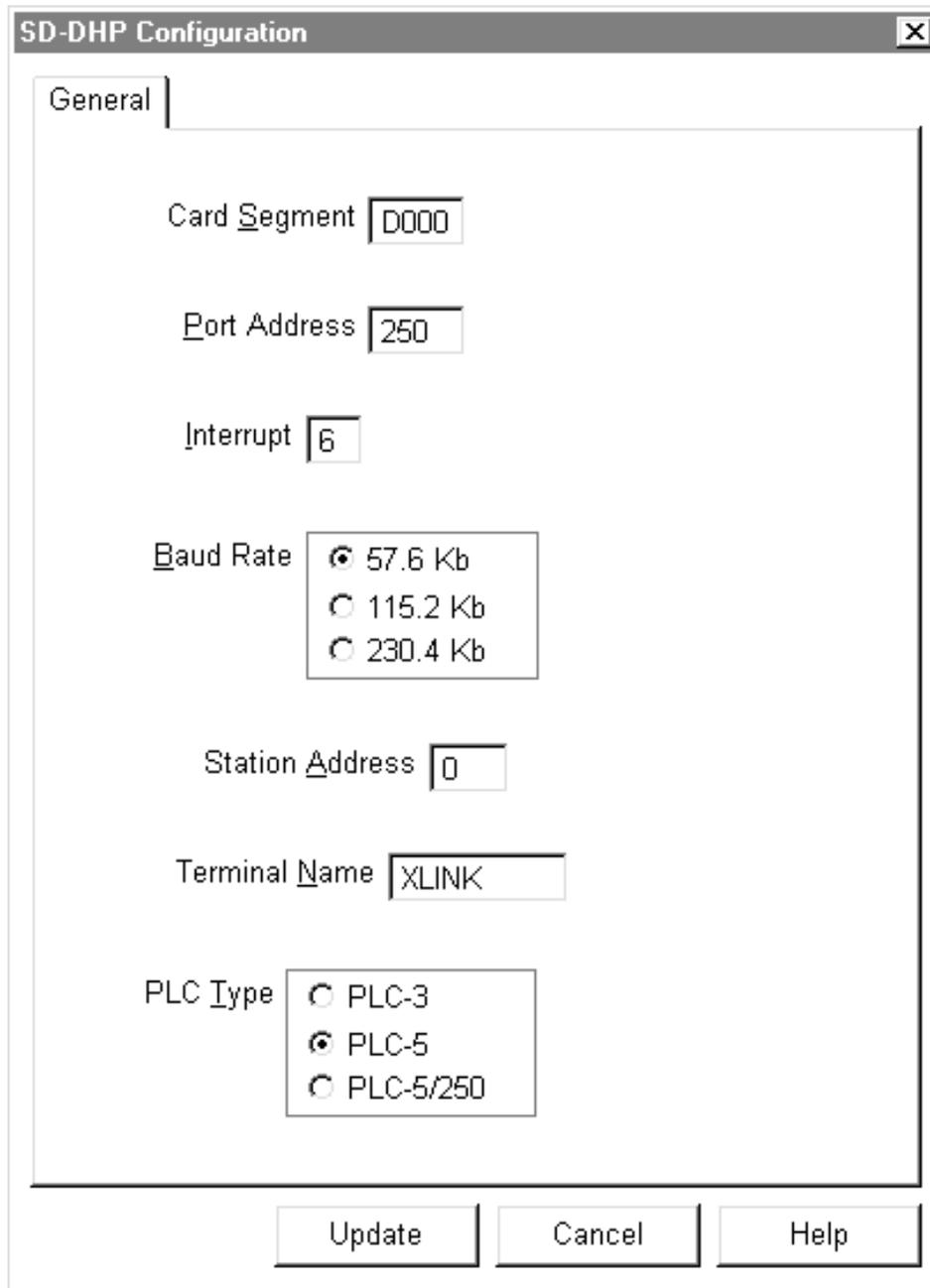
For cost reasons, we recommend using the SST 5136-SD Data Highway Plus interface card. The driver for this card is SD-DHP.DRV.

Pull down the Download menu and select Driver 1. A list of driver files will be presented, select this driver from the list, as shown in Figure 9 below.



**Figure 9. Selecting Driver # 1**

After a few moments, a pop-up dialog window requests some information about the Allen-Bradley driver, as shown in Figure 10 below. We take the default values with two exceptions: IRQ6 is chosen for the interrupt while 0 is chosen for the station address.

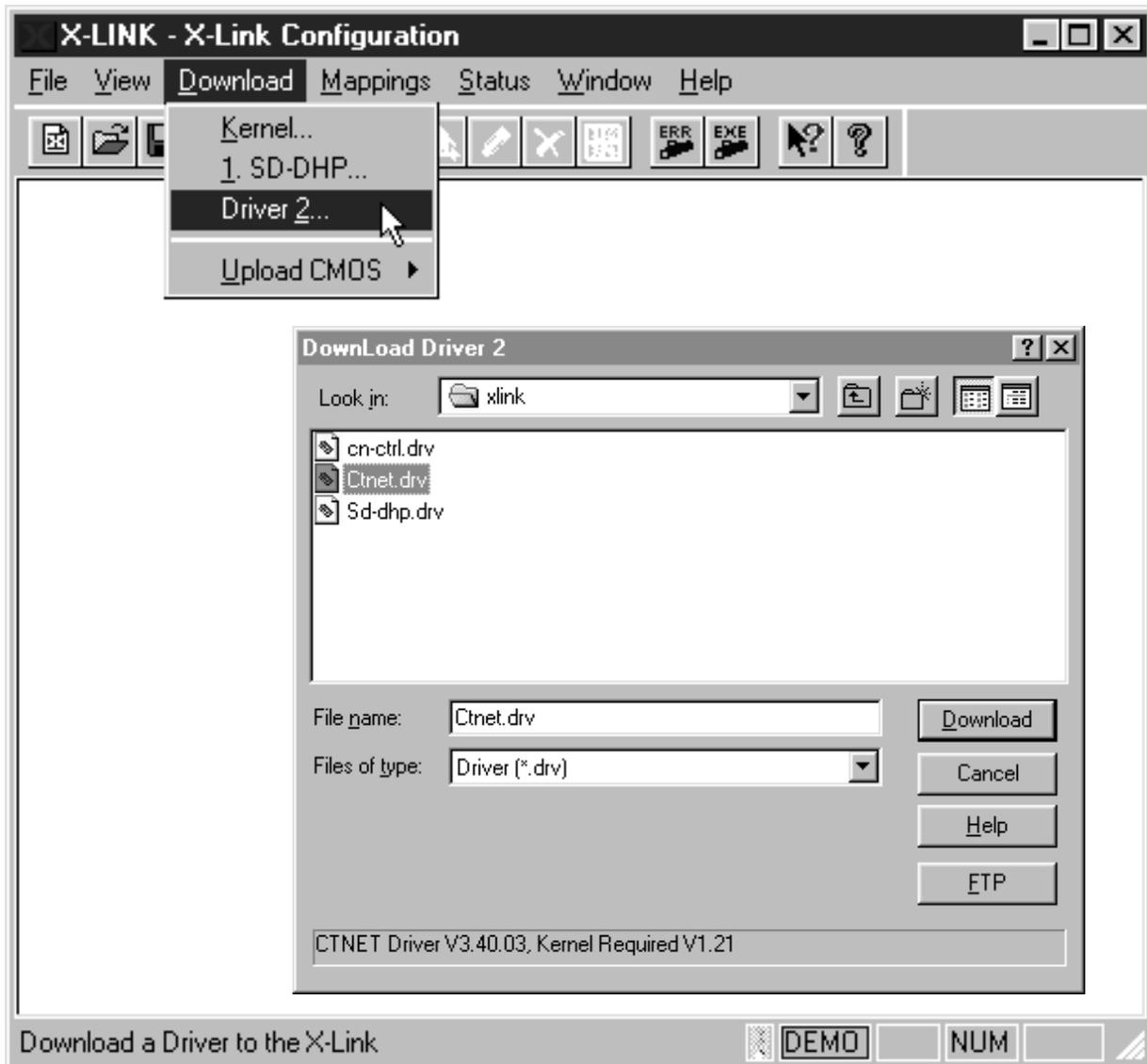


**Figure 10. Configuring Driver # 1**

Be very sure that the settings in the above dialog box conform to the hardware settings on the SD-5136 Data Highway interface card.

## 2.9 Loading the Control Techniques CTNet Driver

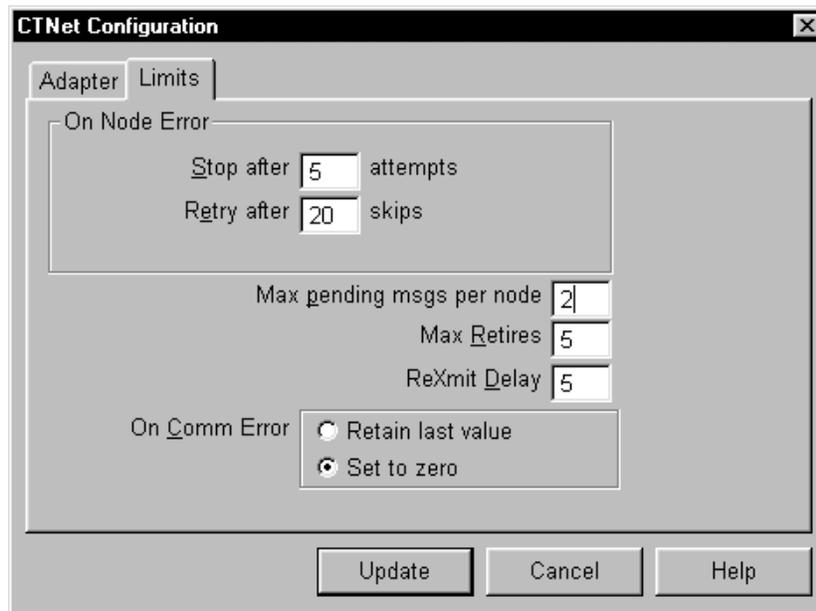
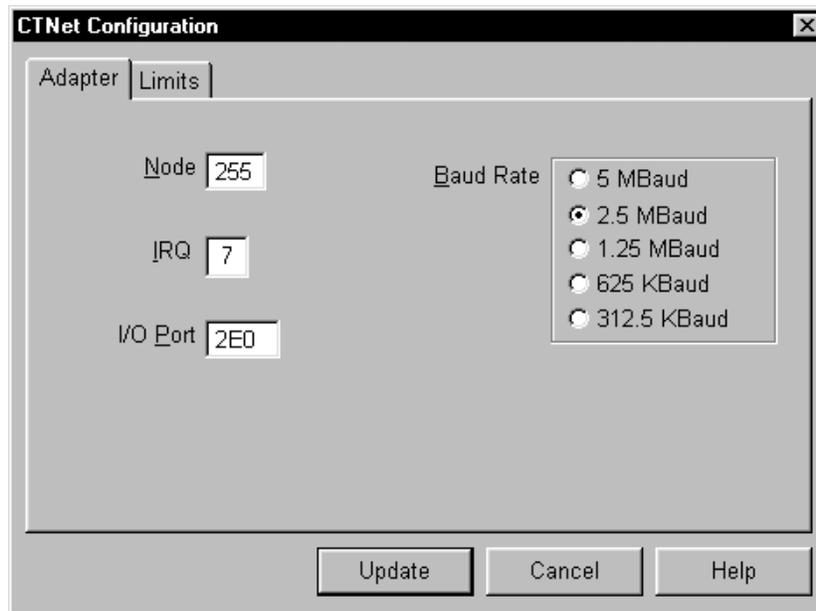
The CTNet driver is loaded by selecting it in the DOWNLOAD pull-down menu. A dialog box appears, as in Figure 11, to allow you to specify the CTNet driver.



**Figure 11. Selecting Driver # 2**

After a few moments, the CTNet driver configuration dialog box appears, as shown in Figure 12 below.

The CTNet driver requires a working CTNet network attached. If this is not the case, the system will return a “hardware error” since the X-Link box will determine that it is the “sole station” on the CTNet token-ring network.



**Figure 12. Configuring Driver # 2**

The error limit field specifies how many attempts will be made to a disconnected node before declaring that node “off-the-air”. The retry delay field specifies how long the X-Link will wait before attempting communication again with an “off-the-air” node. In Figure 12, this retry delay period has been set to one second. Finally, the comm error field specifies what is to be done if a node goes “off-the-air”. The X-Link can return the “last good value” or zero to the requesting network.

After 30 seconds, the CTNet driver download will complete and both drivers will join their respective networks and their status LEDs will blink green.

## 2.10 Loading the Mappings

Once the X-Link kernel and the two drivers have been downloaded, the X-Link engine begins execution. Of course, there's no network activity on either side since no "mappings" have been defined. A mapping must be defined to initiate any communication.

X-Link provides a lot of flexibility in defining mappings; this is described in much more detail in the next chapter. For this simple quick-start example, we will define one simple mapping:

**Allen-Bradley PLC-5 Register n10:0 -----> CTNet Node 12 Menu 20 Parm 48**

To connect these parameters together, we will use the strategy wherein the X-Link Engine will read the "source parameter" by performing a network transaction and then write the data to the "destination parameter" by a second network transaction. The advantage is that the PLC requires no ladder-logic to create a network transaction and the drive needs no DPL code to create a network transaction either. The disadvantage of this approach is that the X-Link must periodically poll the source data and this adds to network traffic.

Mappings are selected from the MAPPINGS pull-down menu. Select "MAPPINGS" and then select "UPLOAD" to read any mappings currently defined in the X-Link box as shown in Figure 13.

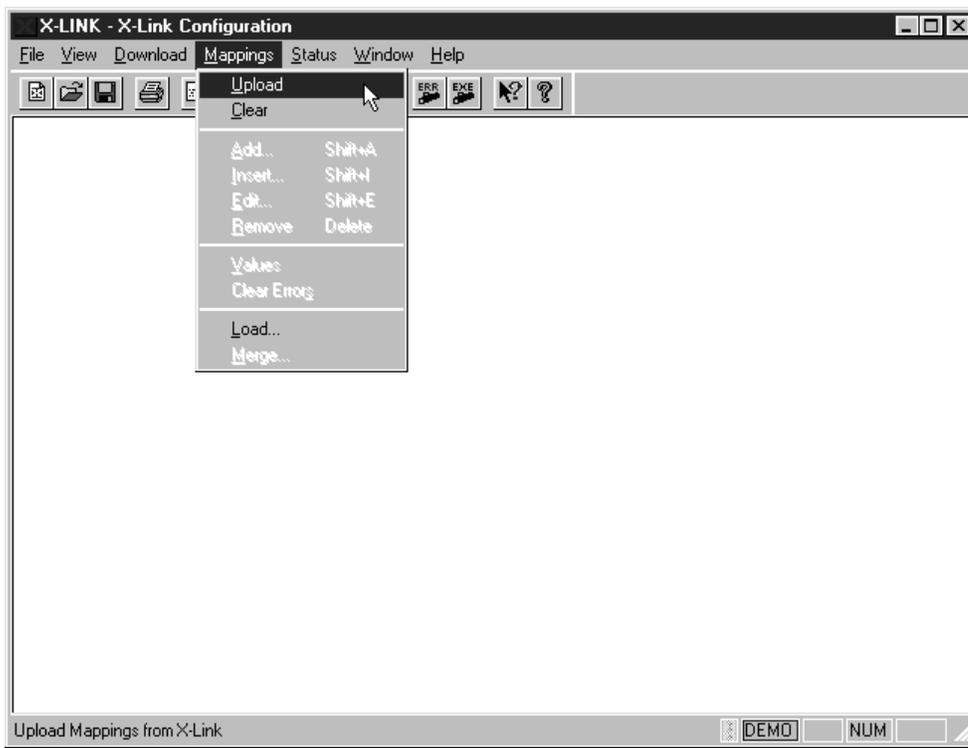
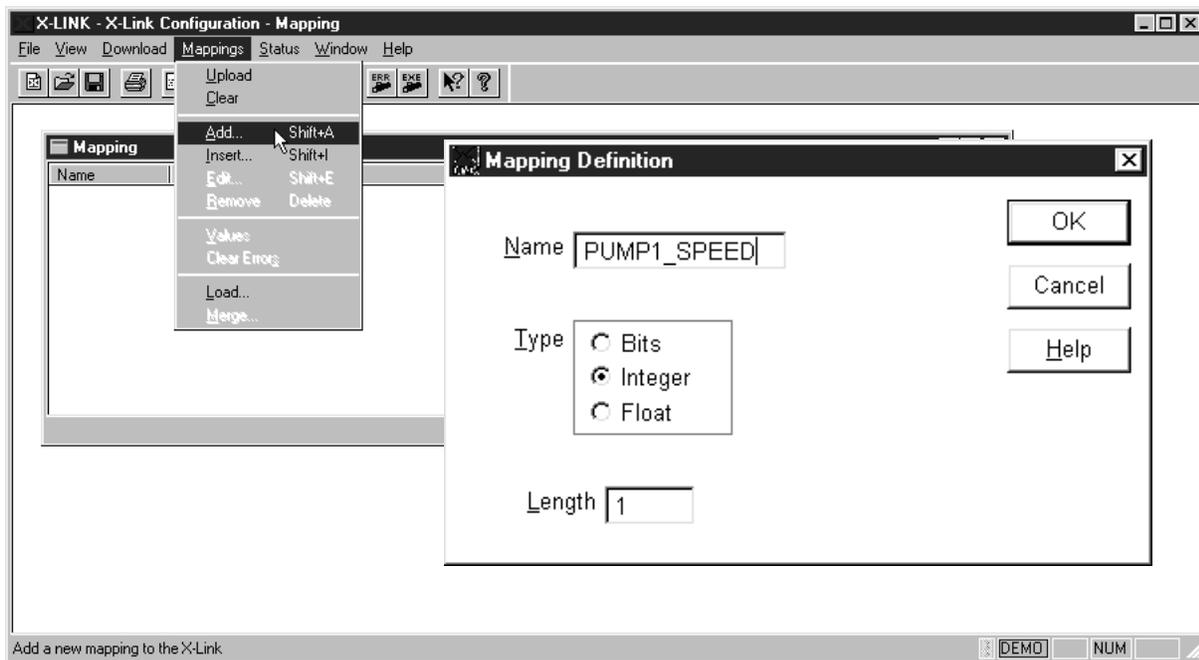


Figure 13. Selecting the Mappings Function

Since we have no mappings defined, we will get a blank mapping screen. From this point on, we create mappings by selecting the ADD option under the MAPPINGS pull-down menu.

When the ADD option is selected under the MAPPINGS pull-down menu, a dialog box appears that allows you to give a name to the mapping. We named this mapping PUMP1\_SPEED. The dialog window also specifies an integer transaction with a length of 1. The mapping description dialog box is shown in Figure 14 below.



**Figure 14. Entering the General Mapping Description**

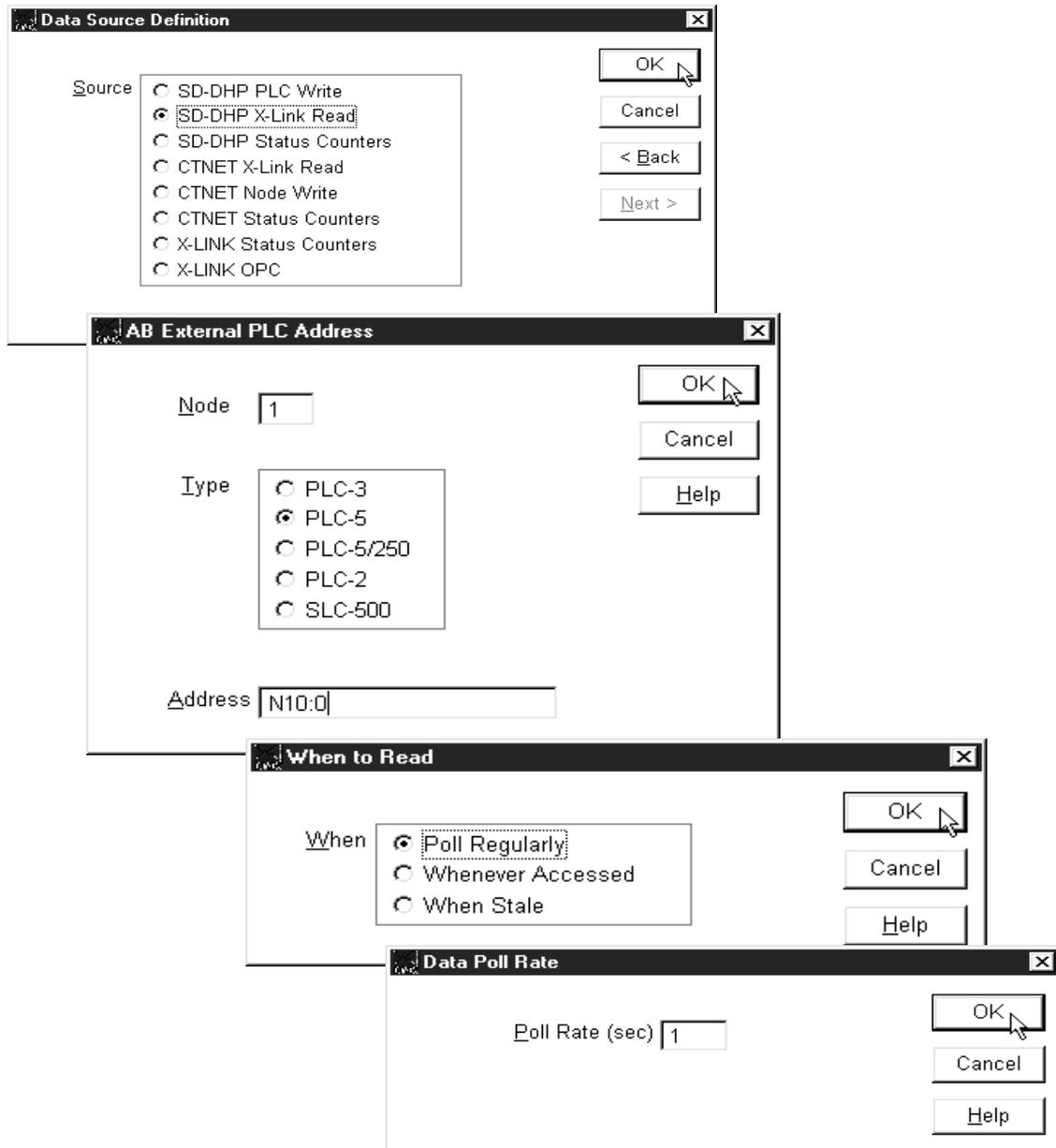
As a reminder, the network mapping is as follows:

**Allen-Bradley PLC-5 Register n10:0 -----> CTNet Node 12 Menu 20 Parm 48**

The source PLC register is n10:0 on the PLC-5. The PLC-5 is node 1 on the Data Highway Plus. However, we want the X-LINK engine to start a Data Highway Plus network transaction to read the data in the register. It will have to do this periodically (polled at 1 sec rate). The destination CTNet parameter is node 12, menu 20, parameter 48. We want the X-LINK engine to write this over the CTNet network only when the value changes. Normally this means that the Data Highway is busy being polled but the CTNet only is used when the value has changed.

After selecting OK in the general dialog window in Figure 14, the configuration program will present several dialog boxes to specify the source of the data (namely the A-B PLC-5 register). This dialog box is shown in Figure 15.

In the sequence of dialog boxes shown in Figure 15, we selected SD-DHP X-Link Read as the Data Source. This means that the X-Link engine will start a Data Highway Plus network transaction to read the PLC register. The PLC address is specified in terms of its Data Highway node address (1), its PLC type (PLC-5) and its register number (n10:0). We specified that the X-Link engine must poll regularly to read the data over the Data Highway. Finally, a poll rate of 1 second was chosen.



**Figure 15. Specifying the Source of the Data**

In a mapping, there is only a single source. There may be more than one destination for the data however. This allows a bit in the Allen Bradley PLC-5 to be a start/stop bit and the X-Link can be used to propagate it to twenty drives if required. In this simple quick-start example, we only specify a single destination.

The destination of the data is a drive parameter: node 12, menu 20, parameter 48. After selecting OK in the final dialog window in Figure 15, the configuration program will present several dialog boxes to specify the destination of the data (namely a Control Techniques drive parameter). This dialog box is shown in Figure 16 below.

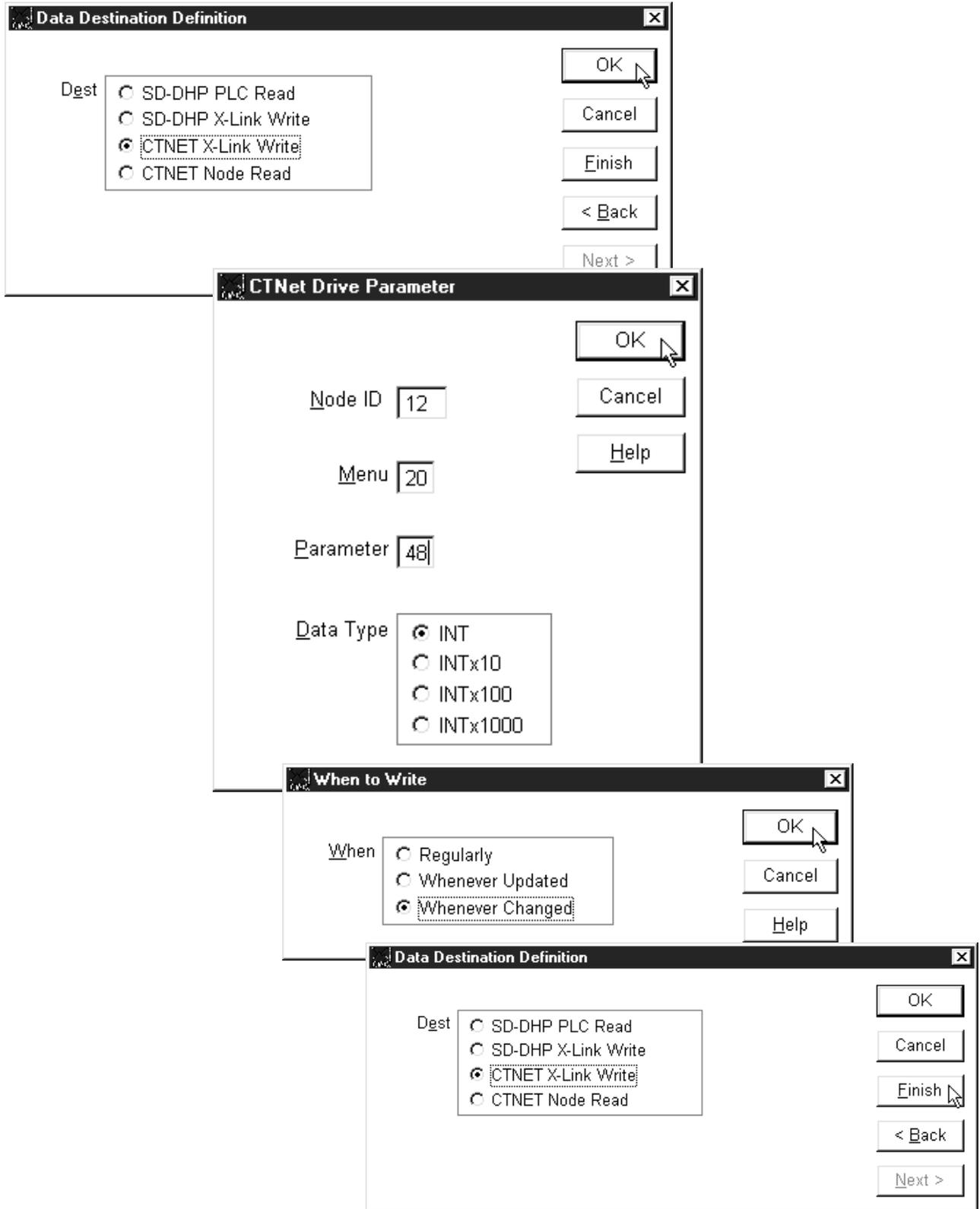
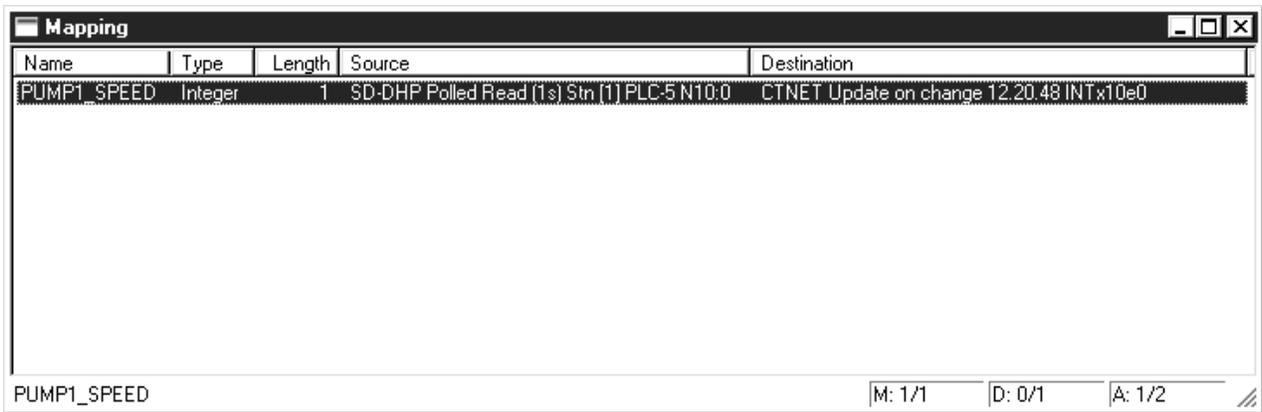


Figure 16. Specifying the Destination of the Data

Note that we specified “CTNet X-Link Write” as our destination. This will cause the X-Link engine to schedule a CTNet network transaction to write the data to the destination CTNet node. We also specified that the X-Link engine is to start a CTNet transaction anytime the PLC-5 register is changed (remember that the X-Link was configured to poll this register via the Data Highway every 1 second). By limiting CTNet transactions to the “whenever changed” condition, we reduce the load on the CTNet network.

The final dialog box in Figure 16 illustrates how the CONFIG program allows specification of additional destinations for the source data. If we had picked an additional destination and clicked on “OK”, then the dialog boxes would repeat and the mapping would show two destinations and so forth. To limit the mapping to one destination, we click on “FINISH” in the final dialog box to complete the mapping.

When the mapping is defined, the config program prints a summary of all mappings as shown in Figure 17.



**Figure 17. Summary Screen Showing One Mapping Defined**

Anytime a mapping is created or modified, there is a pause of several seconds while it is downloaded to the X-Link engine. Then the network communications start and will run as long as power is applied.

The first mapping may be tested by using a PLC programming tool, such as Allen-Bradley’s 6200 Programming Software, to “poke” into integer register n10:0. By browsing the drive’s display to parameter #20.48, these data changes should be visible.

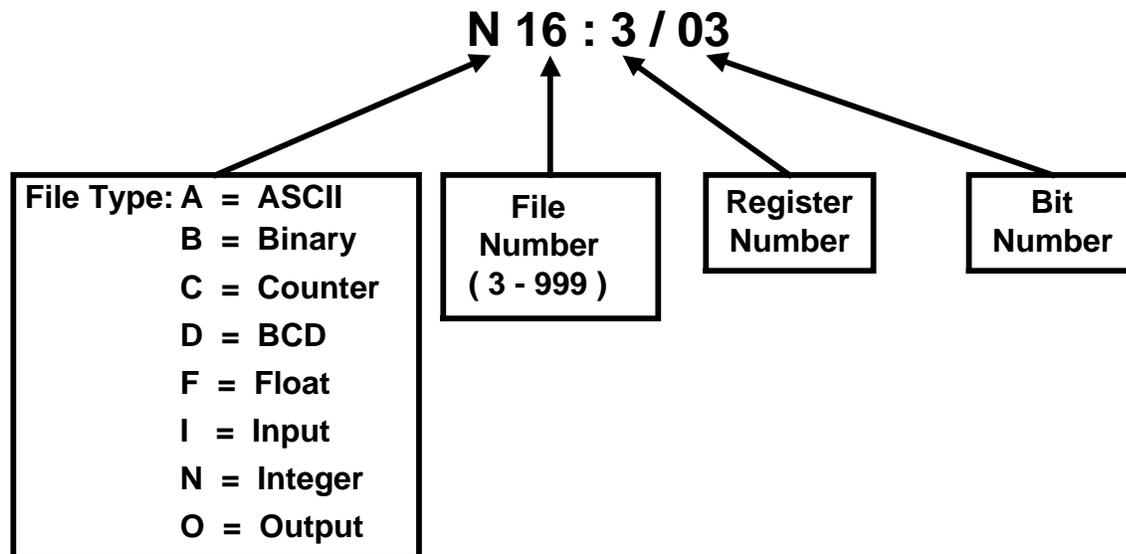
### 3.0 MAPPING FUNDAMENTALS

A mapping is an association of a source data register on one network with one or more destination data registers on another network. This implies a network transaction to fetch the source register and one or more network transactions to store the data into the destination registers.

This chapter uses Allen-Bradley PLCs as the working examples. The reader will have to mentally port this information into other driver possibilities such as Modbus Plus or ControlNet.

Specification of an X-Link mapping involves two simple issues: **what** register is to be accessed and **who** actually initiates the network transaction to access it. Selecting the register is straightforward; you select a source register and deposit it into a destination register. You must be familiar with PLC register naming conventions and CTNet addressing to properly specify a source or destination register. Selecting who initiates the network transaction is more complex but essentially you have the choice of letting the X-Link engine initiate the transaction or let the PLC / CTNet Coprocessor initiate the network transaction. If the latter is chosen, then ladder logic must be supplied in the PLC to execute the MSG instruction to initiate a Data Highway Plus transaction or DPL code must be programmed to execute the RDNET or WRNET instructions to initiate a CTNet transaction.

Allen Bradley PLC's organize their data into a "file" motif. Addressing a register using this nomenclature is as shown in Figure 18.



**Figure 18. Addressing Registers in an Allen-Bradley PLC**

Allen-Bradley Input and Output registers have fixed file numbers (0 for output, 1 for input) and thus the file number needn't be specified for them. The Register Number for Input and Output registers includes a two octal digit rack number and a one digit octal I/O group number. The bit number is specified as octal also. For example, the output bit in rack 1, I/O group 2, bit 3 would be specified as **O:012/3**.

You must also know the Data Highway node address of the PLC you are accessing. Unfortunately, the X-Link Allen-Bradley driver doesn't support bit addressing.

CTNet identifies parameters with a “menu.parameter” designation. There are 99 possible menus and 99 possible parameters within each menu.

**Notation: menu.parameter where: menu = 0 .. 99  
parameter = 0 .. 99**

This is the way Control Techniques MENTOR drives, Unidrives and Beckhoff I/O are organized.

Examples: 07.06 = menu 7, parameter 6 on Unidrive ( analogue input 1 )  
05.09 = menu 5, parameter 9 on Unidrive ( motor rated voltage )

You must also know the CTNet node address of the drive you are accessing.

Disregarding for the moment that the Data Highway, CTNet and X-Link status counters can be the source of a mapping, there are four possible sources for a mapping and four possible destinations, as shown in Figure 19.

<b>SOURCE</b>		
<b>DRIVER</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
<b>SD-DHP</b>	<b>PLC Write</b>	PLC will initiate Data Highway Plus transaction ( ladder logic required )
<b>SD-DHP</b>	<b>X-Link Read</b>	X-Link will initiate the Data Highway Plus transaction ( must periodically poll for the data )
<b>CTNet</b>	<b>Node Write</b>	CTNet will initiate a network transaction ( DPL code required )
<b>CTNet</b>	<b>X-Link Read</b>	X-Link will initiate the CTNet transaction ( must periodically poll for the data )
<b>DESTINATION</b>		
<b>SD-DHP</b>	<b>PLC Read</b>	PLC will initiate Data Highway Plus transaction ( ladder logic required )
<b>SD-DHP</b>	<b>X-Link Write</b>	X-Link will initiate the Data Highway Plus transaction ( write the data when changed, when accessed or ) ( at a specified poll rate )
<b>CTNet</b>	<b>Node Read</b>	CTNet will initiate a network transaction ( DPL code required )
<b>CTNet</b>	<b>X-Link Write</b>	X-Link will initiate the CTNet transaction ( write the data when changed, when accessed or ) ( at a specified poll rate )

**Figure 19. Mapping Sources and Destinations**

### 3.1 Selecting the Mapping Source

There can be only one source specification. Certain drivers, such as the Allen-Bradley Data Highway Plus driver, permit multi-word block transfers. This would permit moving a block of data in a single Data Highway transaction. Currently, the current CTNet driver supports only single 32-bit word transfers.

#### 3.1.1 Allen-Bradley SD-DHP PLC Write

Selecting this option means that ladder logic in the PLC will initiate a Data Highway network transfer to the X-Link engine in order to supply the source data. Allen Bradley PLCs use a MSG instruction to initiate network transfers. When the MSG instruction is placed on a rung, a “control block” address is entered and a data entry screen will appear allowing the programmer to specify all the communication parameters. For example, to send 1 register to the X-Link, use the procedure suggested in Figure 20.

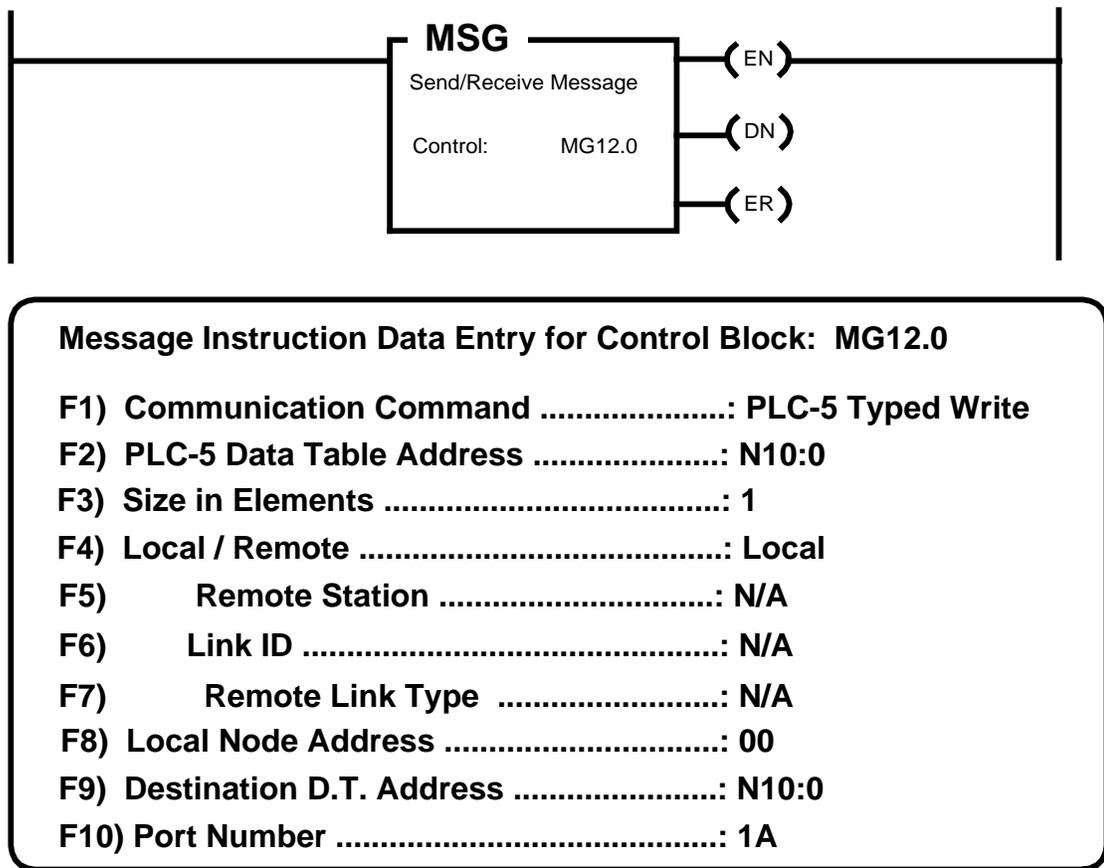
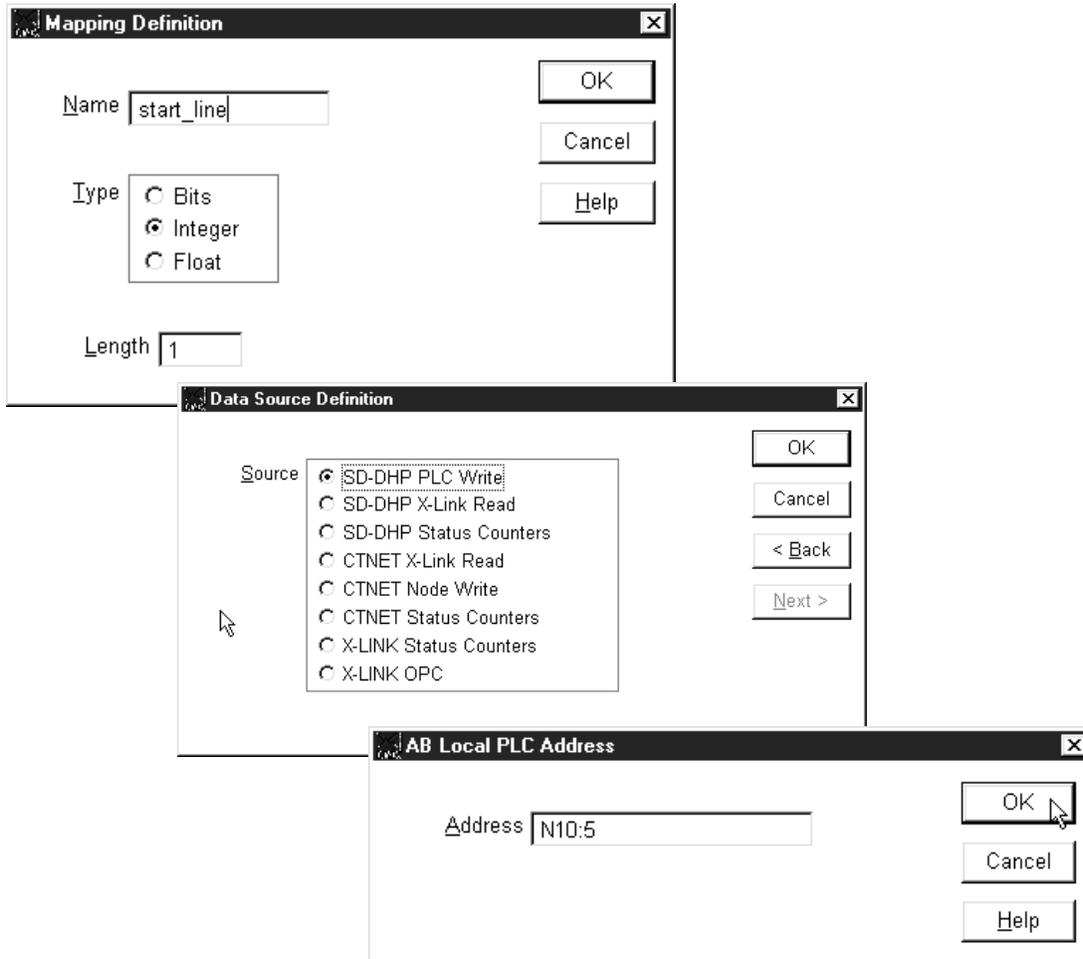


Figure 20. Ladder logic to send one integer register to the X-Link

When this mapping source is selected with the X-Link configuration program, you will be asked to identify a Data Highway register inside the X-Link engine that the incoming PLC data will be deposited into. This can be any legal DH+ address, the X-Link creates a pseudo-register to hold the data. Note that it's a good procedure to write a block of ten registers using the PLC's MSG instruction and then create ten mappings to distribute the individual registers to CTNet

destination parameters. This is very efficient since it still uses only one Data Highway block transfer to move the data to the X-Link.

Figure 21 below shows the sequence of dialogs presented to specify PLC Write as the source of the data.



**Figure 21. Dialogs required to specify PLC Write as the source**

Use of this option is extremely efficient in terms of network bandwidth and responsiveness. The ladder logic can be fashioned to only transmit a Data Highway packet when data, such as a switch or reference, has actually changed. This helps reduce the network traffic. Also, the X-Link engine will act on the incoming data as soon as it comes in; it may instantly schedule a CTNet transaction to relay the data to the destination register.

### 3.1.2 Allen-Bradley SD-DHP X-Link Read

Selecting this option means that the X-Link engine itself will query for the source data by initiating a Data Highway Plus transaction on a regular basis. The X-Link configuration program will require specification of the DH+ address of the data as well as a polling rate to do this. Refer to Figure 15 for an example of specifying "SD-DHP X-Link Read" as the source data.

The big advantage of this method is that no additional ladder logic is required. The X-Link system can get data from a PLC that has already been programmed.

### 3.1.3 CTNet Node Write

Selecting this option means that DPL code in the UD70 / MD29AN coprocessor will initiate a CTNet network transfer to the X-Link engine in order to supply the source data. Figure 22 shows typical DPL code needed to send a start/stop switch (one parameter) connected to the UD70's Digital Input #0 to the X-Link engine via the CTNet.

```
$define NET_OK 1

background {
old_start_stop% = #86.01

top:
// if start/stop switch has changed state, do a CTNet write to the X-Link
if #86.01 <> old_start_run% then

    // write start / stop bit to X-Link node
    // 255    = destination node ID ( X-Link )
    // 86     = menu number on destination node ( X-Link )
    // 1      = parameter number on destination node ( X-Link )
    // #86.01 = value to write
    // 0      = decimal point (not used)
    // 100    = timeout in msec
    status% = WRNET (255, 86, 1, #86.01, 0, 100)

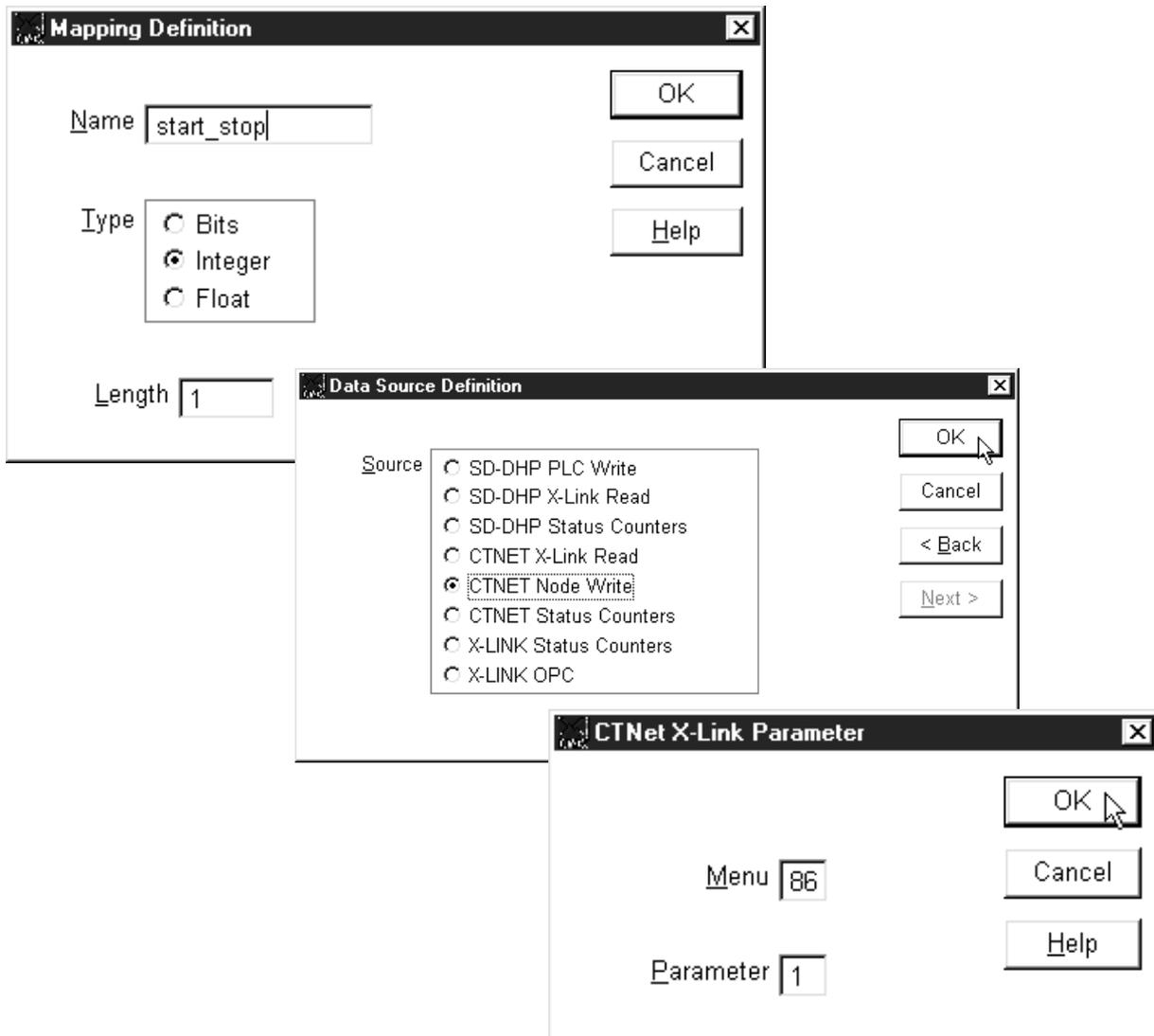
    // Did we get a response or time out ?
    if status% = NET_OK then
        // message was accepted, update previous state
        old_start_stop% = #86.01
    else
        // timed out, force the driver to NET_IDLE state
        // by not updating previous state, CTNet write will be resubmitted
        status% = NETREPLY(0)
    endif
endif

goto top:
}
```

Figure 22. DPL Code to write one parameter to the X-Link to act as the source

Figure 23 below shows the sequence of dialogs presented to specify CTNet Node Write as the source of the data. We labeled the mapping CTNET-write and specified that we wanted the incoming data to be deposited into X-Link local CTNet parameter #86.01. This is just a pseudo-parameter within the X-Link that temporarily holds the incoming parameter before it is relayed to the destination network.

Use of this option is extremely efficient in terms of network bandwidth and responsiveness. The DPL code can be fashioned to only transmit a CTNet packet when data, such as a switch or reference, has actually changed, as shown in the example above. This helps reduce the network traffic. Also, the X-Link engine will act on the incoming data as soon as it comes in; it may instantly schedule a Data Highway Plus transaction to relay the data to the destination register.



**Figure 23. Dialogs required to specify CTNet Node Write as the source**

### 3.1.4 CTNet X-Link Read

Selecting this option means that the X-Link engine itself will query for the source data by initiating a CTNet transaction on a regular basis. The X-Link configuration program will require specification of the CTNet address of the data as well as a polling rate to do this. Refer to Figure 24 below for an example of specifying “CTNet X-Link Read” as the source data.

The big advantage of this method is that no DPL code is required. The X-Link system can get data from a drive coprocessor that has already been programmed.

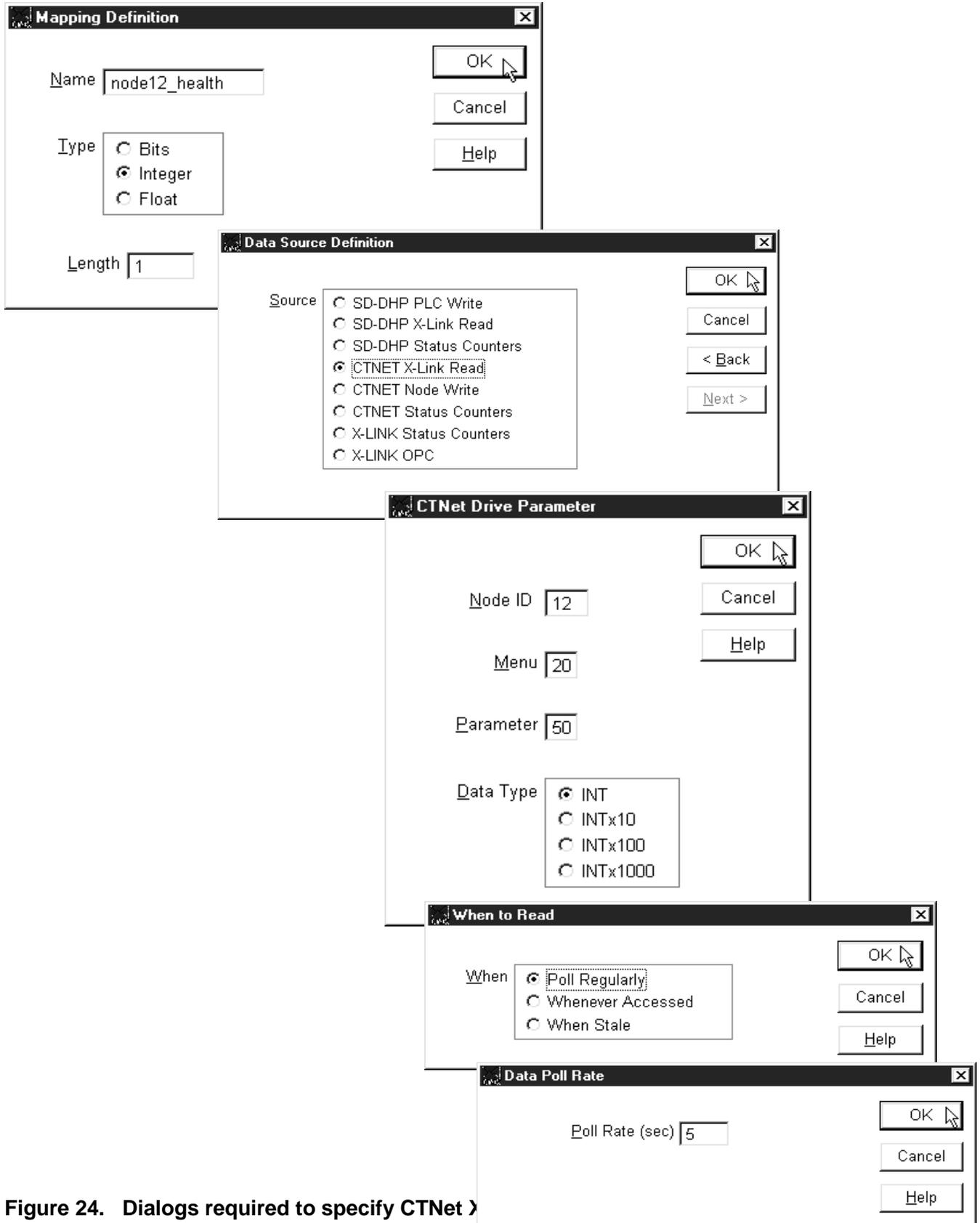


Figure 24. Dialogs required to specify CTDNet X-Link Read

## 3.2 Selecting the Mapping Destination

There can be multiple destination specifications. This can be very useful in commanding twenty drives to start from a single I/O bit in a PLC.

### 3.2.1 CNet X-Link Write

Selecting this option means that the X-Link engine itself will write the source data to the CNet destination parameter by initiating a CNet transaction. The X-Link configuration program will require specification of the CNet address as well as when to perform the write. Refer to Figure 25 for an example of specifying “CNet X-Link Write” as the destination data.

In specifying when the CNet X-Link Write is to be initiated, you have three choices:

<b>Poll Regularly</b>	<b>writes the data periodically, user specifies a poll rate</b>
<b>Whenever Updated</b>	<b>writes the data whenever the source network writes to X-Link</b>
<b>Whenever Changed</b>	<b>writes the data whenever source network changes its value</b>

The big advantage of this method is that no DPL code is required. The X-Link system can write data to any drive or I/O BOX parameter. It’s also extremely efficient if the “whenever changed” option is selected.

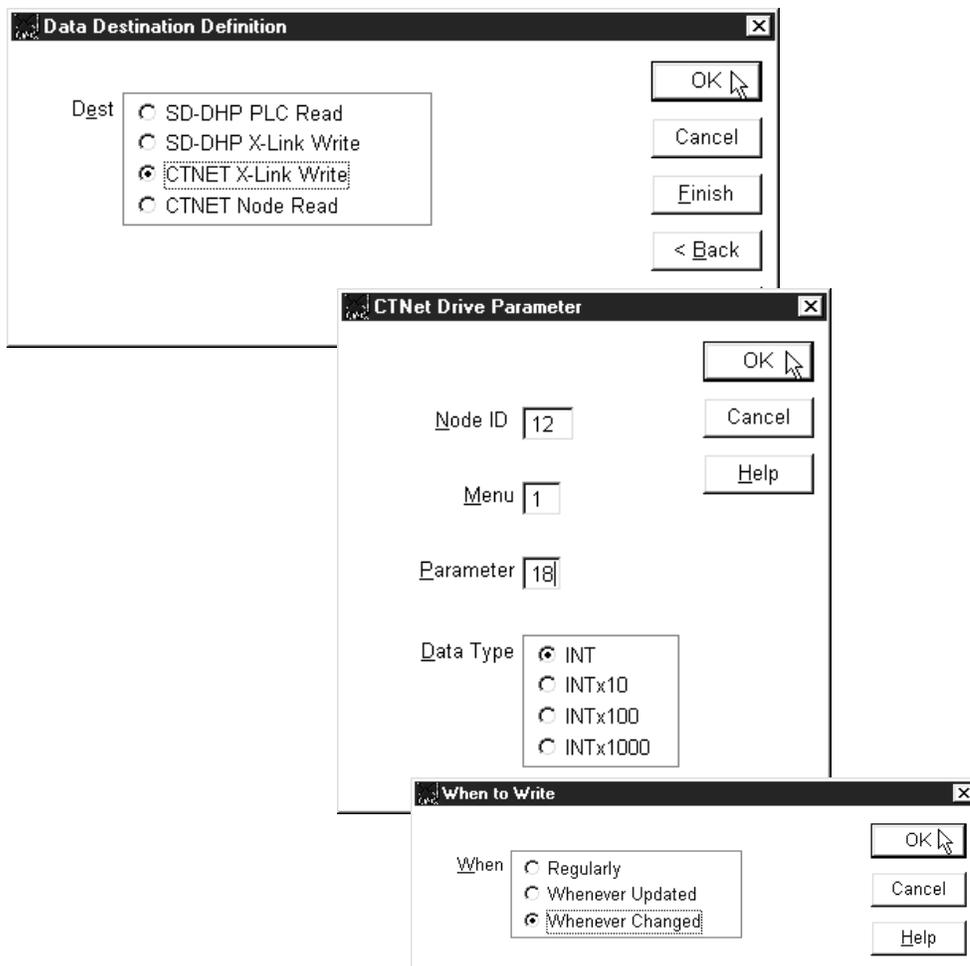


Figure 25. Dialogs required to specify CNet X-Link Write as the destination

### 3.2.2 CTNet Node Read

Selecting this option means that the CTNet coprocessor node will initiate a network READ to fetch the data; this requires DPL code on the coprocessor. Whatever it reads will be the most recent value propagated over the source network to the X-Link. Figure 26 shows the dialog boxes required to set this up.

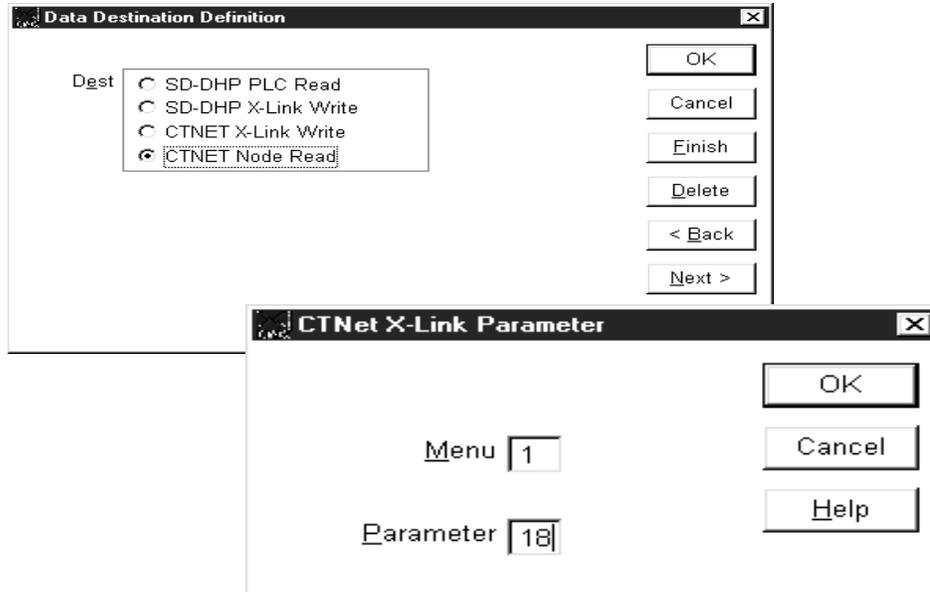


Figure 26. Dialogs required to specify CTNet Node Read as the destination

Typical DPL code to read this parameter via CTNet is shown in Figure 27 below:

```
$define NET_OK 1

background {
top:

// Read 32-bit reference from X-Link node
// 255 = X-Link node ID ( X-Link )
// 73 = menu number on X-Link
// 00 = parameter number on X-Link
// 100 = timeout in msec
status% = RDNET (255, 73, 0, 100)

// Did we get a response or time out ?
if status% = NET_OK then
// message was accepted, update the reference
reference% = NETREPLY(0)
else
// timed out, force the driver to NET_IDLE state
status% = NETREPLY(0)
endif

goto top:
}
```

Figure 27. Typical DPL code to read X-Link data

### 3.2.3 SD-DHP PLC Read

Selecting this destination option means that the PLC will initiate the network transaction to read the source data from the X-Link and deposit it into a PLC register. Obviously, ladder logic is required in the PLC to initiate this transfer, an example is shown in Figure 27.

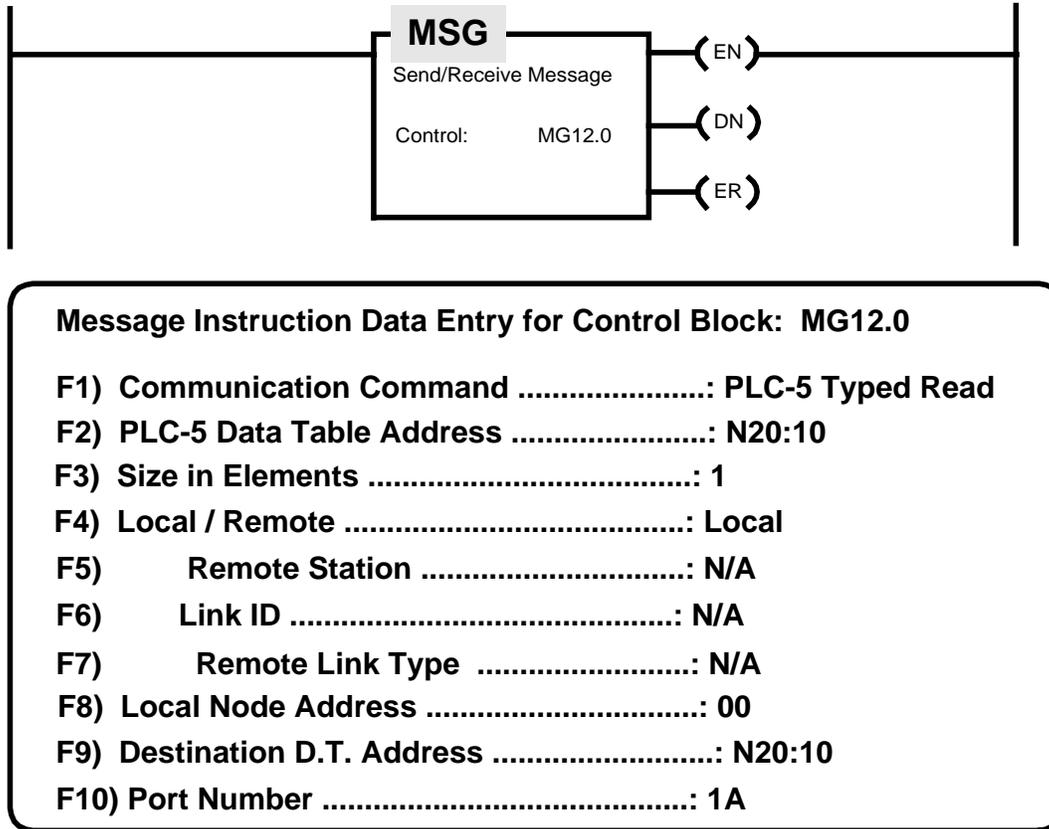


Figure 27. Ladder logic to read one integer register from the X-Link

The dialogs required to set up this transfer in the X-Link are shown in Figure 28.

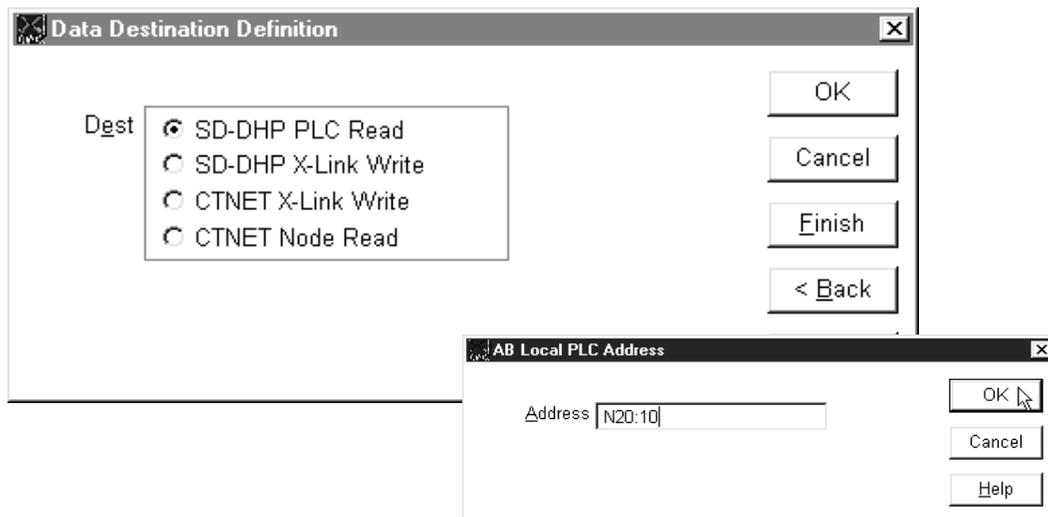


Figure 28. Dialogs required to specify SD-DHP PLC Read as the destination

### 3.2.4 SD-DHP X-Link Write

Selecting this option means that the X-Link engine itself will write the source data to the PLC destination register by initiating a Data Highway Plus transaction. The X-Link configuration program will require specification of the Data Highway address as well as when to perform the write. Refer to Figure 28 for an example of specifying “SD-DHP X-Link Write” as the destination data.

In specifying when the SD-DHP X-Link Write is to be initiated, you have three choices:

<b>Poll Regularly</b>	<b>writes the data periodically, user specifies a poll rate</b>
<b>Whenever Updated</b>	<b>writes the data whenever the source network writes to X-Link</b>
<b>Whenever Changed</b>	<b>writes the data whenever source network changes its value</b>

The big advantage of this method is that no ladder logic is required. The X-Link system can write data to any PLC register. It's also extremely efficient if the “whenever changed” option is selected.

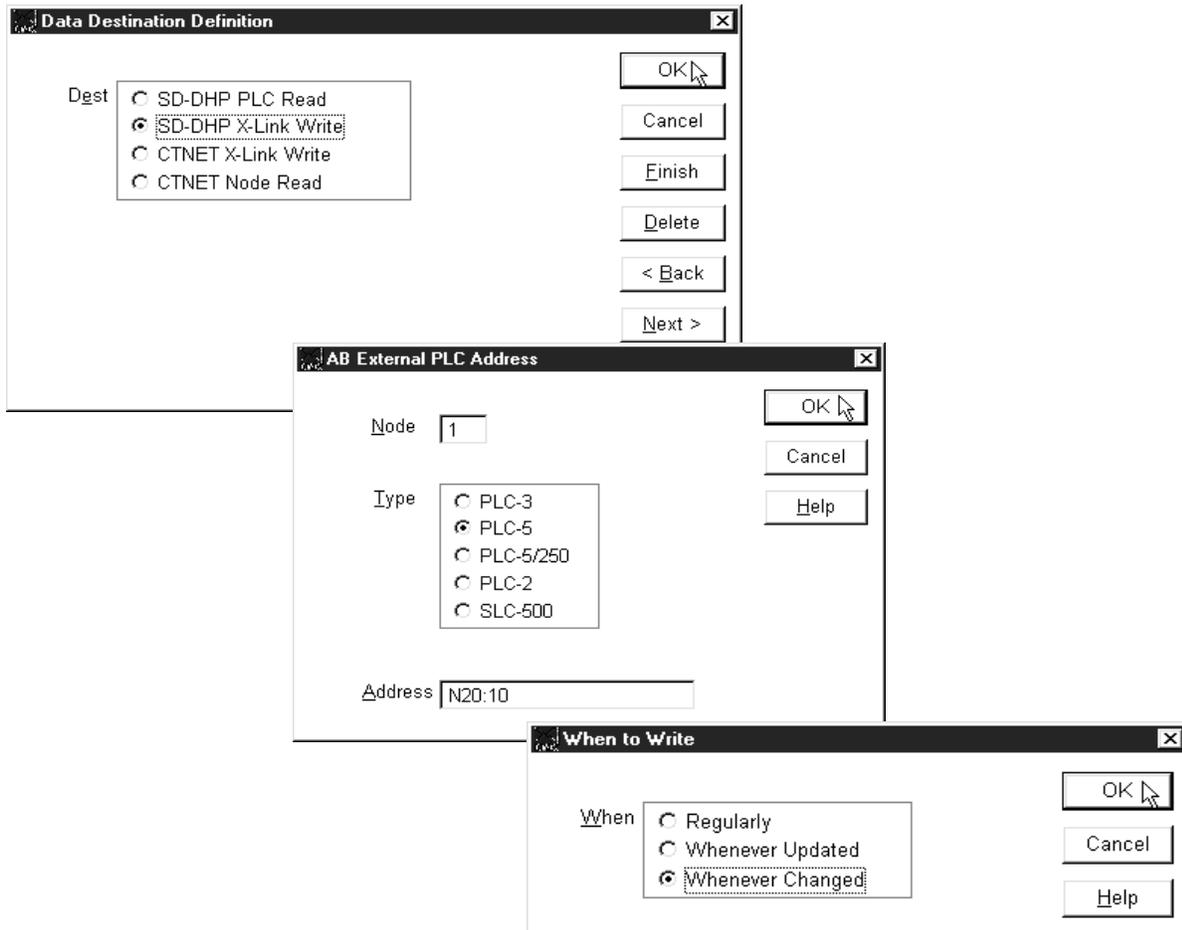


Figure 33. Dialogs required to specify SD-DHP X-Link Write as the destination

## 4.0 MAPPING STRATEGY

In Chapter 3, we learned how to properly specify a mapping. It's quite another thing to intelligently specify all the mappings to get a desired level of throughput. Blindly setting a large number of mappings to a zero poll rate only guarantees that no mapping will get through very fast. To understand why this is the case and how to cleverly avoid these traps, the system designer must be aware of a few facts about the X-Link box.

### 4.1 Unsolicited Transactions are Faster

A crucial consideration in the design of any mapping is “who initiates the transaction?”

If the PLC or CTNet coprocessor initiates the transaction, this request to read or write can come at any moment and is thus totally asynchronous. This type of message is called an unsolicited transaction and is interrupt-driven. It is serviced at the highest priority by the X-Link kernel.

If the X-Link initiates the transaction by polling, then the expiration of a timer causes the X-Link kernel to service the transaction. This operation has lower priority than the unsolicited type.

### 4.2 All X-Link Transactions are Queued

Mappings always result in one or two network transactions. These transactions are dumped into an internal queue within the X-Link kernel. Queue transactions are handled on a “first-in, first-out” basis. If there are 200 mappings all scheduled to occur at one second polling intervals, the elapse of a one second timer within the X-Link kernel will cause 200 transactions to be loaded into the queue.

These queued transactions **must run to completion**. There is no way to force a more important transaction that has arrived afterwards to the top of the queue. The more important transaction must wait until all 200 transactions have completed.

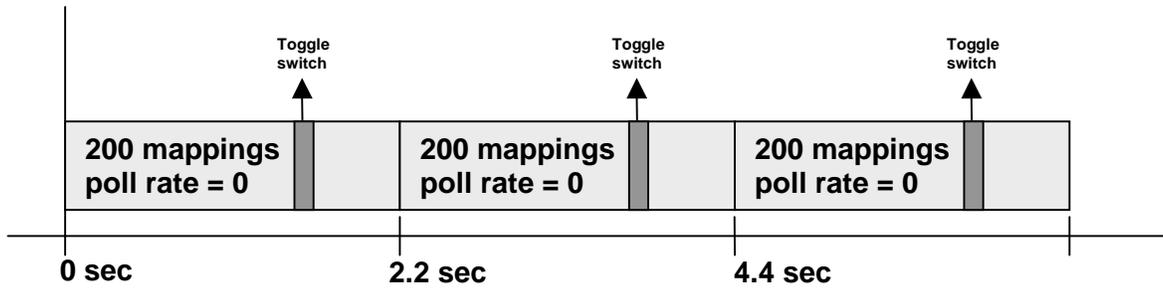
### 4.3 The X-LINK Box can do 91 CTNet transactions per Second

X-Link CTNet operations use the “one shot” message system in which all messages have a response. The UD75 Coprocessor is programmed to process these “one shot” messages every 8 milliseconds and this ultimately limits the throughput on the CTNet side. Currently, the X-Link is limited to 91 transactions per second on the CTNet side.

### 4.4 The Danger of Setting Polled Mappings to Zero Poll Rate

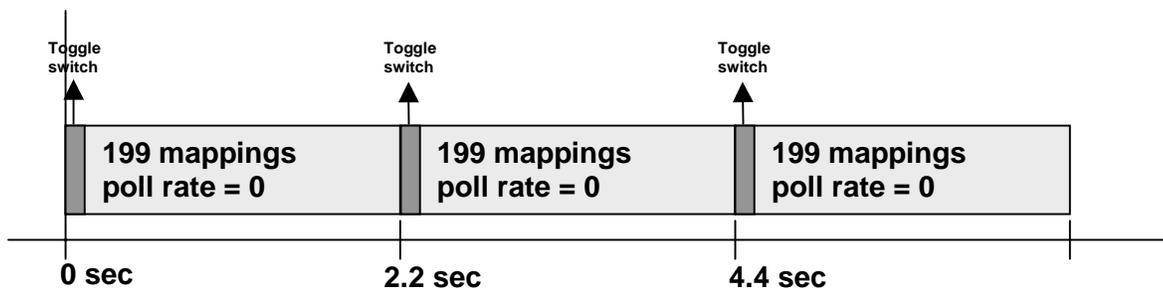
Zero polling rate means “poll as fast as possible”. Assume there are 200 CTNet X-Link Read mappings defined, all defined at a zero poll rate. At the outset 200 transactions will be dumped into the queue. They must run to completion and this will take a little over 2 seconds at 91 transactions/sec. As soon as these 200 transactions complete, they will be immediately re-inserted into the queue since we specified “zero poll rate”. If one of these mappings is a “line start” switch from a Beckhoff I/O system, the two-second delay may be intolerable.

This situation is illustrated below in Figure 34.



**Figure 34. Throughput with all 200 mappings set for zero poll rate**

The obvious solution is to set the “toggle switch” mapping at a zero poll rate (fastest) and set all the other mappings to a one-second poll rate. This doesn’t really help. The 199 one-second poll rate mappings must still run to completion before the next zero poll rate mapping can be processed, as shown in Figure 35.



**Figure 35. Throughput with 199 mappings set for one-second poll rate**

Two factors are working against us here. First, all mappings set at the same poll rate are loaded into the queue simultaneously when the timer expires and must run to completion. Second, we have too many mappings for a system that can only process 91 mappings per second.

#### 4.5 Staggered Polling Allows Important Data to Get Through

One trick is to break the mappings up into groups of forty mappings, each group with a different poll rate. Another trick is to pick poll rates that are prime numbers. This lessens the occurrences of spatial mapping coincidences (those time instants when more than one group comes due).

Breaking the mappings up into smaller groups with differing polling rates is really quite reasonable. There is no need to sample a “fault word” at a high speed – it is only important when there’s a failure. Information for SCADA displays can be sampled at moderate speeds since the update rate of these systems is typically slow. Toggle switches and trip status are the type of thing that must be sampled as fast as possible.

In the above example, let’s break the mappings up into five groups. Twenty-four mappings will run at the zero poll rate. The other four groups of forty-four mappings will be run at 2 seconds, 3 seconds, 5 seconds and 7 seconds respectively. Figure 36 below shows the first 40 seconds of operation of such an arrangement.

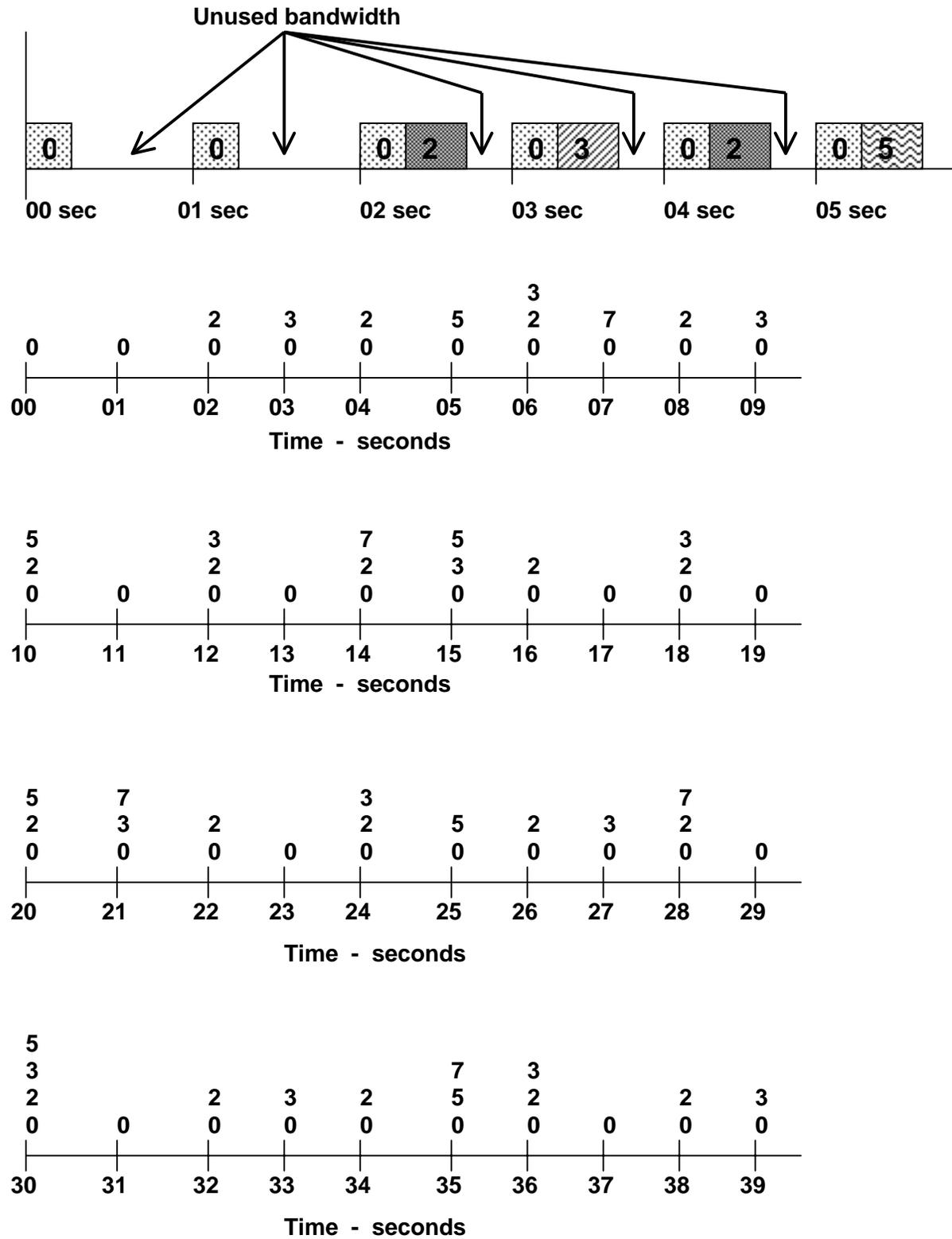


Figure 36. Operation of Staggered Mappings

Figure 36 above shows what poll-rate groups are scheduled each second. It illustrates how the zero poll rate group (24 mappings) **almost always gets through** every second. Often only a single non-zero poll rate group is scheduled and it always completes within one second. Where there are multiple non-zero poll rate groups scheduled, the last non-zero mapping will fall behind but the system will catch up within one or two seconds. A triple coincidence occurs every thirty seconds while the rare quadruple coincidence occurs every 210 seconds. The solution above is not perfect but is a significant improvement on the “all zero poll rate” version shown earlier.

#### 4.6 Polling Can Interfere with Unsolicited Transfers

Unsolicited transactions are high priority and are scheduled as soon as they arrive, effectively placing them ahead of the polled mappings. However, once polled mappings are scheduled, they must run to completion before any other mappings can be processed.

If three hundred polled mappings are placed in the queue, any unsolicited messages from the PLC or Drive will have to wait until the polled messages have cleared.

#### 4.7 Interleaving Mappings is More Efficient

Never bombard a single node with more than two mappings at a time. Assigning more than two mappings contiguously to a single node may result in some transactions being missed since the drive's CTNet processor can handle only two at a time.

In the example directly below, two mappings for CTNet node 12 are specified first. Then mappings for other nodes are interleaved before any additional mappings for CTNet node 12 are specified. Mappings are placed in a queue and having too many contiguous mappings to a single node will bombard it with too many transactions.

Map00	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:0	CTNet Update on change 12.6.30 INTx1e0
Map01	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:1	CTNet Update on change 12.6.31 INTx1e0
Map02	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:2	CTNet Update on change 13.6.30 INTx1e0
Map03	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:3	CTNet Update on change 13.6.31 INTx1e0
Map04	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:4	CTNet Update on change 14.6.30 INTx1e0
Map05	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:5	CTNet Update on change 14.6.31 INTx1e0
Map06	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:6	CTNet Update on change 12.6.32 INTx1e0
Map07	1	SD-DHP Polled Read (1S) Stn(1) PLC-5 N10:7	CTNet Update on change 12.6.33 INTx1e0

## 4.8 Mapping Strategies to Try

Choosing the appropriate mapping strategy is more difficult when there's a large number of mappings and some of the mappings must be fast. Here are some suggestions to try.

- **Make all mappings unsolicited**

If the PLC must write to the drive, use PLC WRITE as the source and CNet X-LINK WRITE as the destination. This requires custom ladder logic on the PLC but the transfer can be programmed to only occur when the data actually changes state. If the PLC must read from the drive, use CNet NODE WRITE as the source and PLC X-LINK WRITE as the destination. This requires custom DPL code, but this code can be fashioned to only transmit when the data actually changes state. This methodology gives the best throughput with the minimal network load.

- **Use staggered mappings**

The staggered mappings described in section 4.5 above will provide adequate throughput in most cases, but the occasional coincidence of mapping groups being scheduled simultaneously will cause some fast mappings to be late once in a while.

- **Use Extra X-Link Boxes**

This is a radical and expensive solution, but in the most difficult cases purchase of additional X-Link boxes is sometimes the only solution.

## 5.0 ADVANCED CONCEPTS

The X-Link system is capable of sophisticated networking setups. The following sections provide some suggestions concerning these advanced capabilities and the reader is invited to experiment with these features. Some of these ideas may be useful in a future application.

### 5.1 Multiple Destinations

It's very common to create a mapping that has multiple destinations. For example, a start/stop button may be propagated to twenty destinations to accomplish a line start. With CNet, each additional destination will go out on successive token rotations. Figure 37 below shows a start/stop mapping that includes 4 destinations.

Name	Type	Length	Source	Destination
start_stop	Integer	1	SD-DHP PLC Write N10:0	CTNET Update on change 12.6.30 INTx10e0 CTNET Update on change 13.6.30 INTx10e0 CTNET Update on change 14.6.30 INTx10e0 CTNET Update on change 15.6.30 INTx10e0

start\_stop M: 1/1 D: 4/4 A: 5/5

### Figure 37. Propagating a Source to Multiple Destinations

Don't forget that the destinations can be on both networks. Figure 38 is an example of a start/stop bit being propagated to a drive and a PLC I/O output to act as an indicator. In fact, the destination PLC is a different PLC from the one sourcing the start/stop bit. In the second destination, the [5] indicates that we selected Data Highway Plus address 5 as the destination. This mapping implies two Data Highway transactions.

Name	Type	Length	Source	Destination
start_stop	Integer	1	SD-DHP PLC Write N10:0	CTNET Update on change 12.6.30 INTx10e0 SD-DHP X-Link Writ on Change Stn [5] PLC-S N10:8

start\_stop M: 1/1 D: 2/2 A: 3/3

Figure 38. Propagating a Bit to Two Destinations on Two Networks

## 5.8 Floating Point Transfers

While the CTNet network specification includes floating point data, none of the current Control Techniques drives actually use floating point data. For example, the power factor parameter, #5.10, looks like a floating point value since it displays as .852 on the Unidrive's display. However, it is transmitted to the Unidrive as an integer in "times 1000" motif (852). The Allen-Bradley Data Highway driver supports floating point data and it is possible to map a A-B floating point register to a CTNet parameter and still have everything work out. Figure 39 shows how this is done.

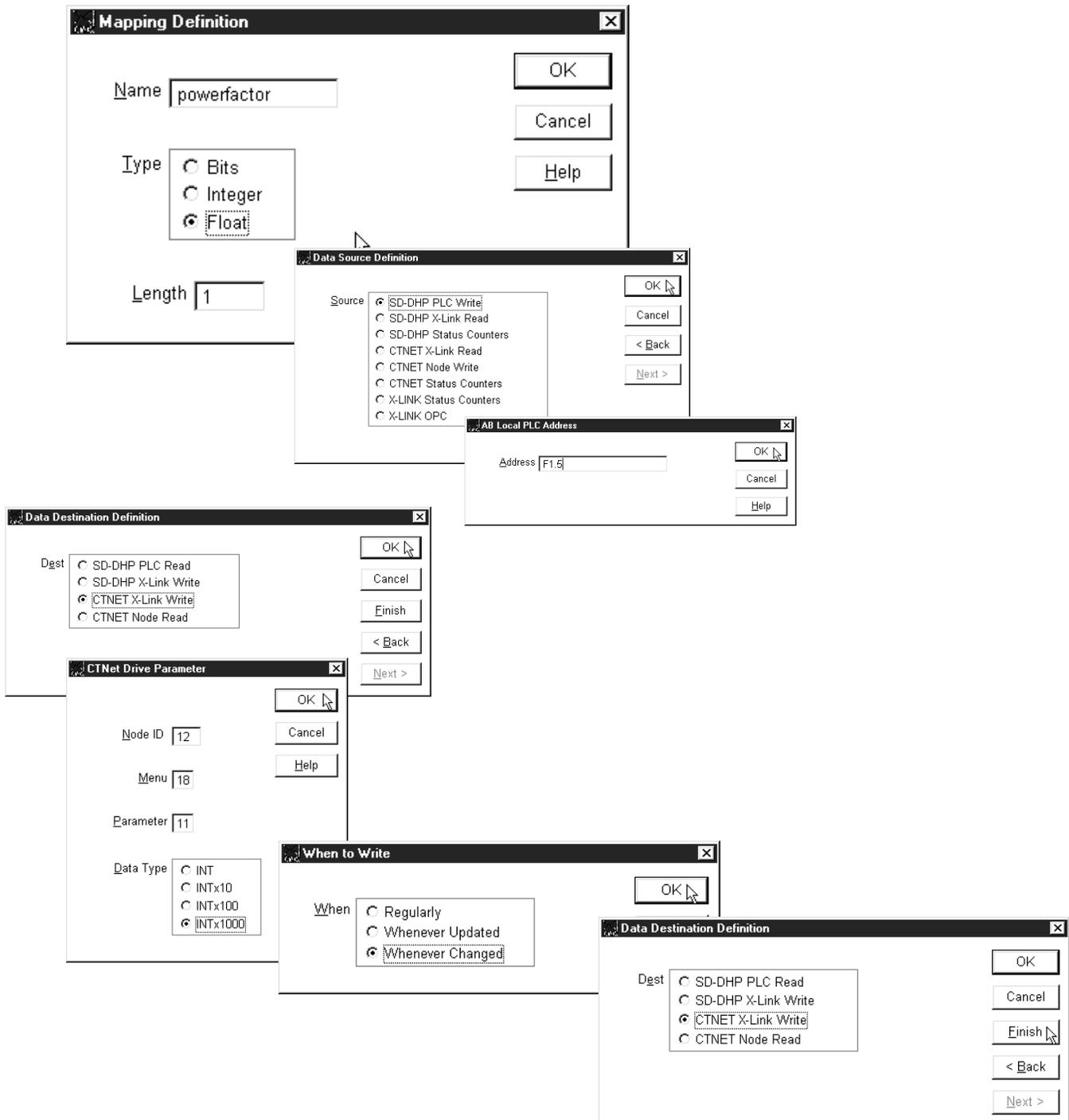
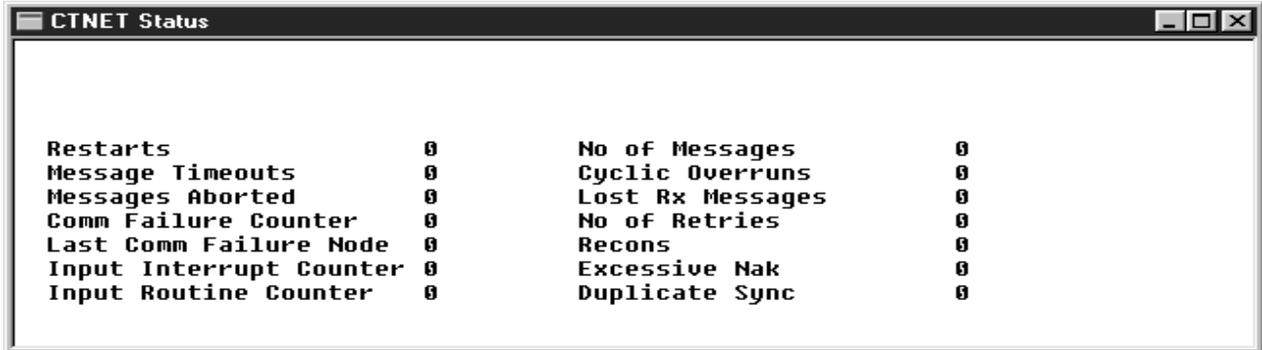


Figure 39. Mapping to Move Floating Point Data

### 5.3 Status Counters

Each driver maintains running diagnostic data called “status counters”. For example, Figure 40 shows the display when you select the STATUS pull-down menu followed by the DRIVER 2 option (in this case, the CTNet driver).



**Figure 40. Inspecting the Status of the CTNet driver**

The meaning of the CTNet driver status counters is as given in the table below.

Status	Offset	Description
Restarts	0	Total number of times the driver has been restarted
Message Timeouts	2	Total number of times a response from a read or write did not arrive before the timeout period elapsed. Could mean that a node has gone off the air.
Messages Aborted	4	Number of times the driver has given up on sending a message
Comm Failure Counter	6	# of times a node has been declared offline
Last Comm Failure Node	8	The last node that was declared offline
Input Interrupt Counter	10	Counts each network comms interrupt ( xmit, receive, etc.) Should be increasing if the node is sending or receiving.
Input Routine Counter	12	Counts number of times the input routine has been entered. Typically one for each transaction.
No of Messages	14	total number of messages sent and received.
Cyclic Overruns	16	# of cyclic overruns ( indicates that cyclic period ended before all cyclic messages were sent out).
Lost Rx Messages	17	# of lost receive messages ( indicates that incoming are coming in faster than the UD70 / CTNet module can unload them ).
No of Retries	18	# of retries ( indicates that outgoing messages are being retransmitted, a high figure would suggest excessive network noise, etc. ).
Recons	19	# of RECONS ( indicates that the CTNet network is re-configuring. It's normal to have a few RECONS during startup, but an increasing count indicates network trouble. ).
Excessive NAKs	20	# of excessive NAKs ( indicates that a receiving station is having trouble accepting messages from this node ).
Duplicate Sync	21	Not applicable for this version of the driver

These status counters can be used as a “source” in an X-Link mapping. For example, an excessive number of CTNet “recons” indicates network problems. A recon is normal when a node comes on or off the air or at initialization, but after that there should be no recons. It’s possible to send this recon count to a PLC so that ladder logic can be created to use a timer to build a recon indicator that would flicker noticeably if recons are occurring at a significant rate. Figure 41 shows a mapping where the recon counter is sent back to a PLC register.

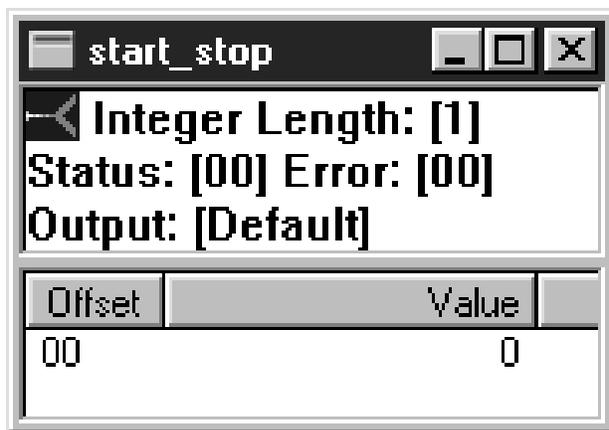


**Figure 41. Mapping the RECON counter to a PLC Register**

When identifying a status counter as a data source, a dialog box will ask for an “offset” to the desired status item. The offsets are given in the table above for the CTNet driver, consult the hardware manuals for the network cards used by other drivers to determine the offset for their status counters.

#### 5.4 X-LINK Real-Time Values Display

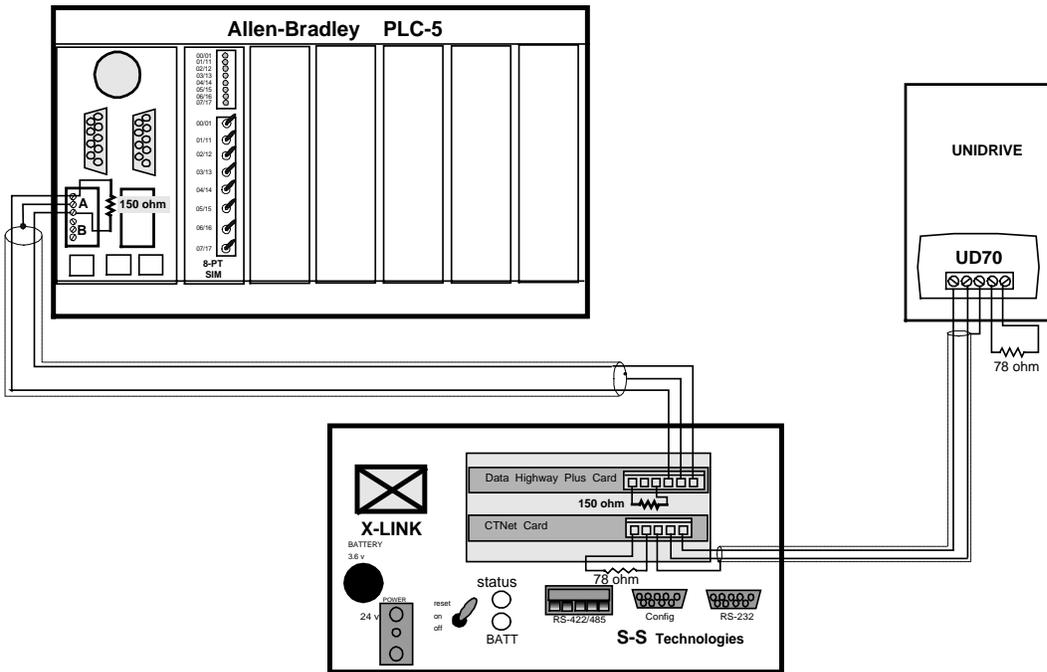
When you “click” on a mapping in the mapping summary, the MAPPING pull-down window has a “VALUES” option that will display real-time the value of the source data for that mapping coming in to the X-Link box. This is an excellent diagnostic tool. Figure 42 shows what this mapping display looks like. The value display is updated continuously.



**Figure 42. Values Diagnostic Display**

## APPENDIX 1 - SAMPLE APPLICATION

In this example, an Allen-Bradley PLC-5 is fitted with an 8-point I/O SIM module. This module has eight switches and 8 LEDs. Switch 00/01 starts and stops the drive by toggling drive parameter #06.30. Switch 01/11 is a reference increase / decrease control that ramps the reference between 0 and 1000 rpm. The reference ramp is written to drive parameter #01.18, which is the precision speed reference.



Application Functions			
Switch	PLC Register	CTNet Parameter	Description
00 / 01	n10:0	#06.30	Start / Stop the Drive 0 = stop the drive 1 = start the drive
01 / 11	n10:1	#01.18	Precision Reference 0 = increase speed ( 0.0 to 1000 rpm ) 1 = decrease speed ( 0.0 to 1000 rpm )

The ladder logic, shown below, is designed to transmit the start/stop bit from register n10:0 to the X-Link and transmit the reference ramp from register n10:1 to the X-Link. By design, the

ladder logic only transmits over the Data Highway Plus when the start/stop bit or the reference actually changes. Both registers are sent to the X-Link as a single block transfer.





