OPTOMUX INTEGRATION KIT USER'S GUIDE

Form 1757-080303—March, 2008

OPTO 22

43044 Business Park Drive • Temecula • CA 92590-3614
Phone: 800-321-OPTO (6786) or 951-695-3000
Fax: 800-832-OPTO (6786) or 951-695-2712
www.opto22.com

Product Support Services
800-TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
Email: support@opto22.com
Web: support.opto22.com

Optomux Integration Kit User's Guide
Form 1757-080303—March, 2008

Copyright © 2008 Opto 22.
All rights reserved.
Printed in the United States of America.

The information in this manual has been checked carefully and is believed to be accurate; however, Opto 22 assumes no responsibility for possible inaccuracies or omissions. Specifications are subject to change without notice.

Opto 22 warrants all of its products to be free from defects in material or workmanship for 30 months from the manufacturing date code. This warranty is limited to the original cost of the unit only and does not cover installation, labor, or any other contingent costs. Opto 22 I/O modules and solid-state relays with date codes of 1/96 or later are guaranteed for life. This lifetime warranty excludes reed relay, SNAP serial communication modules, SNAP PID modules, and modules that contain mechanical contacts or switches. Opto 22 does not warrant any product, components, or parts not manufactured by Opto 22; for these items, the warranty from the original manufacturer applies. These products include, but are not limited to, OptoTerminal-G70, OptoTerminal-G75, and Sony Ericsson GT-48; see the product data sheet for specific warranty information. Refer to Opto 22 form number 1042 for complete warranty information.

Cyrano, Opto 22 FactoryFloor, Optomux, and Pamux are registered trademarks of Opto 22. Generation 4, ioControl, ioDisplay, ioManager, ioProject, ioUtilities, *mistic*, Nvio, Nvio.net Web Portal, OptoConnect, OptoControl, OptoDataLink, OptoDisplay, OptoOPCServer, OptoScript, OptoServer, OptoTerminal, OptoUtilities, PAC Control, PAC Display, PAC Manager, PAC Project, SNAP Ethernet I/O, SNAP I/O, SNAP OEM I/O, SNAP PAC System, SNAP Simple I/O, SNAP Ultimate I/O, and SNAP Wireless LAN I/O are trademarks of Opto 22.

ActiveX, JScript, Microsoft, MS-DOS, VBScript, Visual Basic, Visual C++, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Linux is a registered trademark of Linus Torvalds. Unicenter is a registered trademark of Computer Associates International, Inc. ARCNET is a registered trademark of Datapoint Corporation. Modbus is a registered trademark of Schneider Electric. Wiegand is a registered trademark of Sensor Engineering Corporation. Nokia, Nokia M2M Platform, Nokia M2M Gateway Software, and Nokia 31 GSM Connectivity Terminal are trademarks or registered trademarks of Nokia Corporation. Sony is a trademark of Sony Corporation. Ericsson is a trademark of Telefonaktiebolaget LM Ericsson.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Contents

Using the Integration Kit 1

Overview 1

What is Required 1

Installing the Integration Kit 2

Setting Up Optomux Communication 2

Adding the Subroutines 2

Exporting Charts from the Example 2

Importing the Charts 4

Calling the OPTOMUX_CONFIG Chart 5

Creating the I/O Point String Table 5

I/O Point Definition Guidelines 6

Creating and Using Communication Handles 9

Choosing the Communication Handle Type 9

Optomux Serial Brain Network (chOptomuxPort0) 10

Optomux Serial Brain Network with an SCM (chOptomucSerial10A) 10

Optomux over Ethernet (chOptomuxUDP98 and chOptomuxUDP99) 11

Defining Communication Handles 11

Assigning Communication Handle Index Values 12

Reconfiguring after a Power Failure 13

Subroutine Error Codes 14

Brain Board Error Codes 14

Subroutine Error Codes 14

PAC Control Communication Driver Error Codes 15

Using the Integration Kit

Overview

The Optomux Integration Kit for PAC Project (Part # PAC-INT-OMUX) provides PAC Control™ subroutines that make it easy to talk to Optomux I/O (B1, B2, E1, and E2 brain boards) from a SNAP PAC controller using standard PAC Control commands.

Using the subroutines, communication with Optomux I/O is similar to communicating with PAC I/O. All data passed to or from Optomux analog I/O is in user-defined engineering units. All time values to or from digital or analog I/O are in seconds. No HEX is used. For analog, 0-4095 is not used unless specifically desired. The subroutines support serial communication on all four brain boards as well as Ethernet communication on the E1 and E2 brain boards.

The integration kit contains:

- A set of PAC Control subroutines you add to a strategy to enable an Opto 22 controller to communicate with Optomux I/O
- An example strategy to help you get started

This manual assumes that you fully understand how to use PAC Control, Optomux, and the Optomux devices to be used. For more information on PAC Control and Optomux, see the following documents:

- *PAC Control User's Guide*, form 1700
- *PAC Control Command Reference*, form 1701
- *Optomux Protocol Guide*, form 1572

What is Required

Before including the subroutines in your strategy, you will need:

- A PC running PAC Control Professional 8.1a or later
- The Optomux Integration Kit for PAC Project

You can obtain both PAC Control and the Optomux Integration Kit on the Opto 22 website, http://www.opto22.com/.

Installing the Integration Kit

To install the Optomux Integration Kit, extract the zip file to a directory on your computer. The example strategy is set up to be used in the default directory for PAC Control:

C:\Program Files\Opto22

Therefore, it is recommended that you install the Optomux Integration Kit in this directory.

Setting Up Optomux Communication

As detailed in the sections below, in order to establish communication with your Optomux I/O, you need to do the following things:

- Add the Optomux subroutines to your strategy
- Export the Powerup chart from the example strategy, and rename it OPTOMUX_CONFIG
- Export the OPTOMUX_SEND_RECV chart from the example strategy
- Import the OPTOMUX_CONFIG and OPTOMUX_SEND_RECV charts into your existing strategy
- Add a command to your Powerup chart to start the new OPTOMUX_CONFIG chart
- Edit the Load Optomux Table block to include your I/O point definitions

Adding the Subroutines

You need to add the entire set of PAC Control subroutines to your strategy.

1. Start PAC Control Pro in Configure Mode and open the strategy that you want to use the Optomux protocol.
2. Select Configure→Subroutines Included to open the Subroutine Files dialog box.
3. Click the Add button, browse to the folder containing the Optomux subroutines, and select all of the subroutine files (.ISB extension).
4. Click OK.

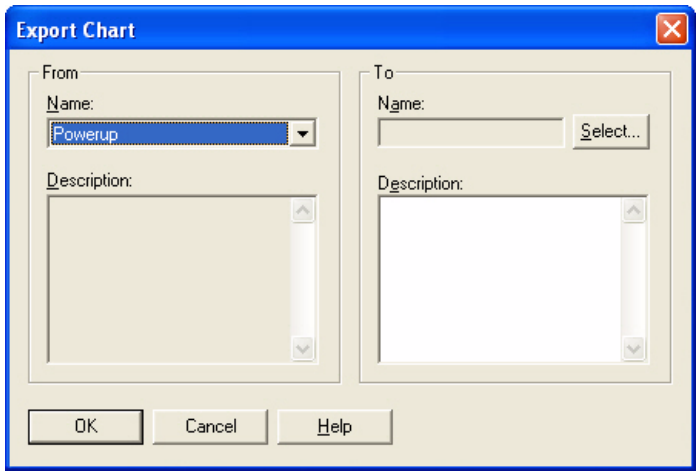
   The subroutines appear in the Subroutines Included folder and are ready to be used in your strategy.
5. Save your strategy.

For more information about subroutines, see form 1700, the *PAC Control User's Guide*.

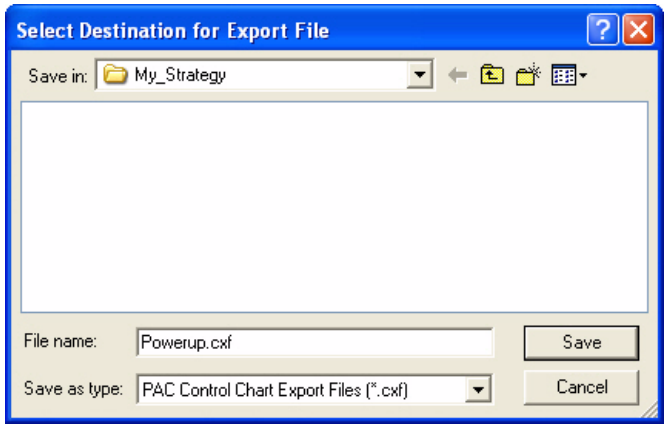
Exporting Charts from the Example

1. In PAC Control Pro, open the example strategy file, PACOptomux.idb, located in the PACControl folder.
2. In Configure mode, choose Chart→Export.

   The Export Chart dialog box appears.



3. In the From section of the dialog box, select Powerup from the Name drop-down list.
4. In the To section of the dialog box, click Select.



5. Navigate to the folder where you want the exported chart file to be saved. In the File name field, change the name to OPTOMUX_CONFIG. Click Save.

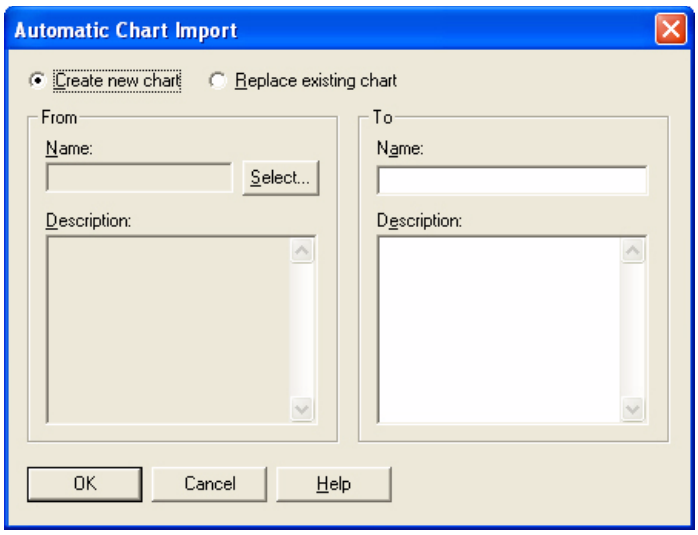
   CAUTION: In order to prevent overwriting the existing Powerup chart, make sure to save the Powerup chart with a new name.

   The Export Chart dialog box now shows the path and file name in the To section.
6. (Optional) Enter a description for the new chart.
7. Click OK.
8. Repeat steps 2-7 to export the chart named OPTOMUX_SEND_RECV, but use the same name, OPTOMUX_SEND_RECV, for the exported file.
9. Close the example strategy

Importing the Charts

1. Open your strategy again. In Configure mode, choose Chart→Import.

   The Automatic Chart Import dialog box appears.



2. At the top of the dialog box, click Create new chart.

   CAUTION: *If you choose Replace existing chart, the old chart will be completely overwritten with the chart you are importing.*

3. Click Select. Navigate to the OPTOMUX_SEND_RECVchart. Click OK.
4. In the To section of the dialog box, enter the name OPTOMUX_SEND_RECV. If you wish, enter a description. Click OK.

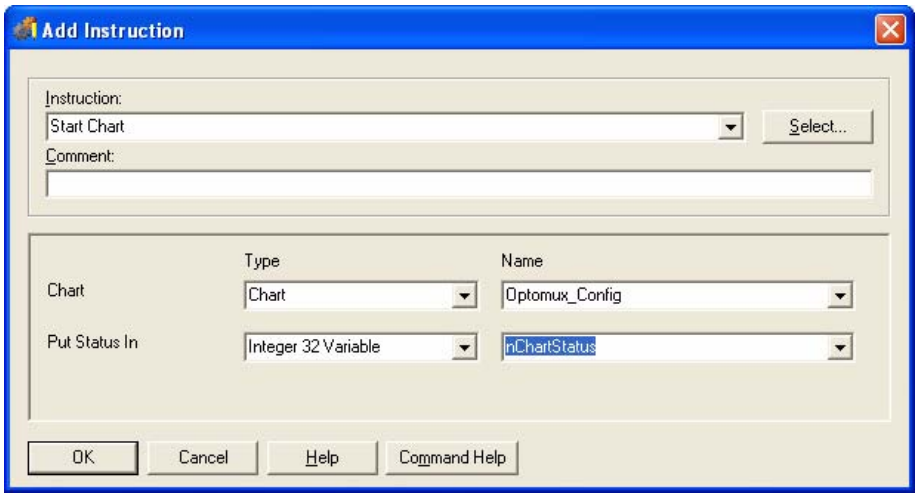
   The chart is imported. A Chart Import Report window shows you how the tags in the chart match with those already in the strategy. Any tags from the chart that do not already exist in the strategy are created and added.

5. Repeat steps 1-4 to import the chart named OPTOMUX_CONFIG. Give the imported chart the same name, OPTOMUX_CONFIG.

Calling the OPTOMUX_CONFIG Chart

To call the OPTOMUX_CONFIG chart in your strategy, you need to use the Start Chart command.

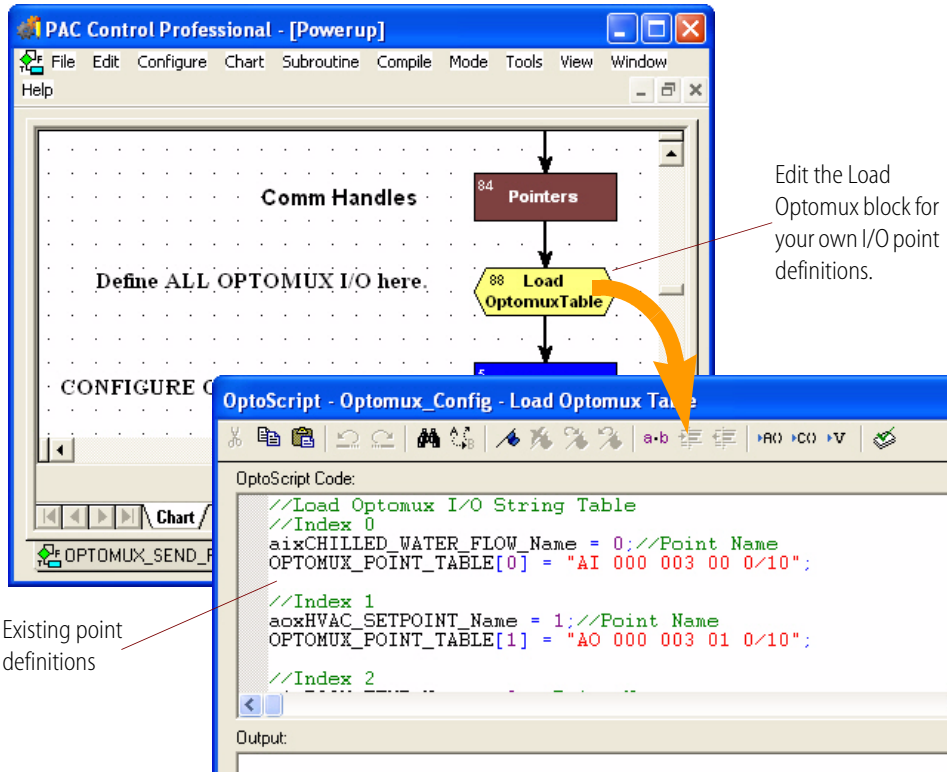
Use the Start Chart command to call the OPTOMUX_CONFIG chart



Creating the I/O Point String Table

To load the Optomux I/O point definitions, do the following:

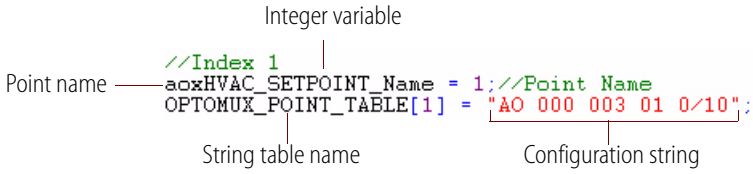
1. Open the OPTOMUX_CONFIG chart in your strategy.
2. Double-click the OptoScript block named Load Optomux Table.



3. Replace the existing I/O point definitions with your own, using the guidelines in the next section.

I/O Point Definition Guidelines

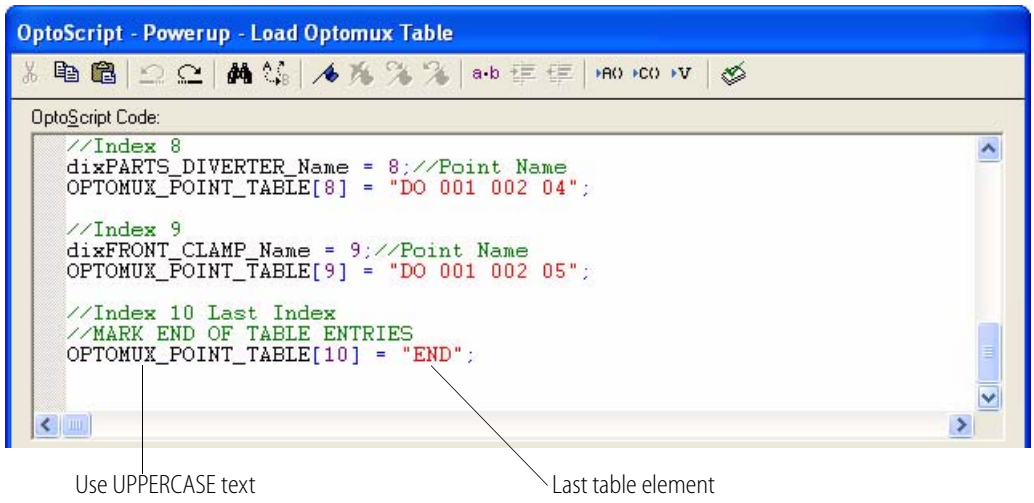
For each I/O point definition in the string table use the format shown here, making sure to use the correct point name, string table name (all CAPS), integer variable, and configuration string.



The order that the points appear in the table is not critical, although it is good practice to put all of a brain board's I/O into one group.

Be sure to remember the following:

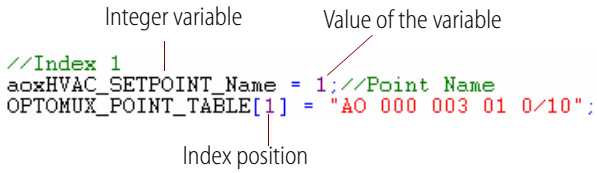
- The string table name, OPTOMUX_POINT_TABLE, must be in UPPERCASE text.
- The *last* table element *must* contain only the word END.



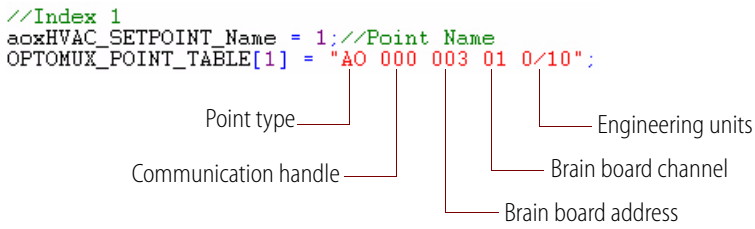
Point Name. Enter the point name defined for your I/O.

String Table Name. Use the name, OPTOMUX_POINT_TABLE, followed by the index number in brackets. Make sure the name is all UPPERCASE, and that there is a unique index number for each entry.

Integer Variable. The integer variable must first be configured or created in the PAC Control strategy tree. For details, see form 1700, the *PAC Control User's Guide*. The value of this variable must match the index position in the string table as shown below.



Configuration String. A configuration string is comprised of the elements shown here.



A typical configuration string is: AI 001 000 00 0/1250

where:

Element	Description
AI	Point type (AI, AO, TI, DI, DO) Type Description AI Analog Input AO Analog Output TI Temperature Input (e.g. AD4 module) DI Digital Input DO Digital Output
001	Communication handle (000-255, *must* use 3 digits)
000	Address of brain board (000-255, *must* use 3 digits)
00	Channel on brain board (00-15, *must* use 2 digits)
0/1250	Engineering units (-99999/999999, *must* use / except for the TI point type) See "Engineering Units" on page 8. See also,"Temperature Input Point Type" below.

See more details below for the *Temperature Input* point type, *communication handle*, and *engineering units*:

- **Temperature Input Point Type**

  If the point type is TI (temperature input), then the engineering units become a single digit (1-9) according to the type of module installed (or, if temperatures are in Fahrenheit rather than Celsius, a single digit followed by the letter F):

Single Digit	Module Type
1	AD4
2	AD14T
3	AD10T2

Single Digit	Module Type
4	AD5T
5	AD8T
6	AD17T 'R'
7	AD17T 'S'
8	AD18T
9	AD19T

  - To read degrees Fahrenheit, follow the single digit with the letter F (for example, 4F). Otherwise °C (degrees Celsius) will be returned.
  - A typical Temperature table entry is: TI 001 005 09 1F
    This specifies an AD4 on port 1, brain board 5, channel 9. Temperature is read in degrees Fahrenheit.

- **Communication Handle**

  Enter a 3-digit value (000-255) to indicate the pointer table index for the correct communication handle for the I/O point's specific brain board. For more information, see "Creating and Using Communication Handles" on page 9.

- **Engineering Units**

  Remember the following things:

  - If omitted, 0-4095 will be returned.
  - Typical entries are: 4/20, -5/+5, 0/50000.
  - Maximum limits are: -99999/999999.
  - The use of the / character is mandatory (except for the TI point type).

Creating and Using Communication Handles

Each communication handle stores the parameters needed to connect to a specific brain board. In order to use a communication handle with your Optomux system, you need to do the following things:

- Choose the type of communication handle for your system.
- Define the communication handle in the Strategy Tree under Communication Handles.
- Enter a 3-digit value (000-255) in the pointer table to indicate the index of the communication handle you want to use.

Choosing the Communication Handle Type

The following table lists the different types of communication handles available in the example strategy, including the index values you need to enter in the configuration string. As part of the example strategy, these communication handles and values are only examples, which you can modify as necessary for your own system.

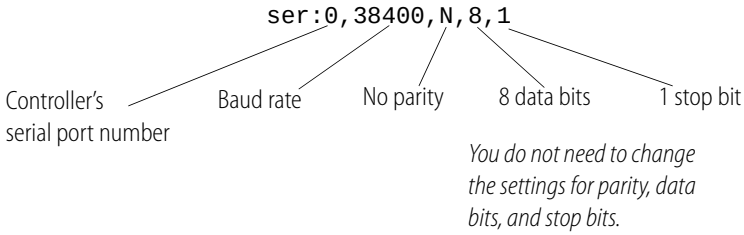
Communication Handle	Initial Value	For Use With:	index Value in Pointer Table*	3-digit Index Value for the Configuration String
chOptomuxUDP98	udp:192.168.1.98:5000	E1 or E2 using Optomux protocol over Ethernet	0	000
chOptomuxUDP99	udp:192.168.1.99:5000	E1 or E2 using Optomux protocol over Ethernet	1	001
chOptomuxPort0	ser:0,38400,n,8,1	Optomux serial brains (B1 and B2, or E1 and E2 in serial mode)	2	002
chOptomucSerial10A	tcp:192.168.1.226:22520	Optomux serial brains (B1 and B2, or E1 and E2 in serial mode) used with a serial communication module (SCM) instead of a serial port built into the controller to connect to the Optomux serial brain network	3	003

* See "Defining Communication Handles" on page 11.

Each type of communication handle is explained in more detail below.

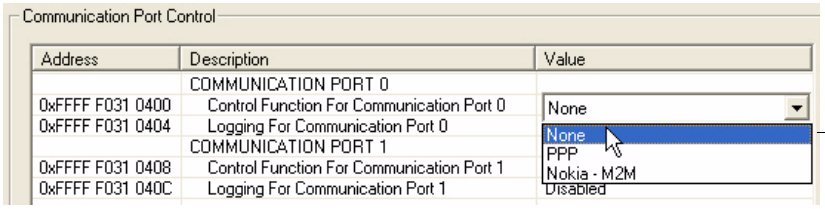
Optomux Serial Brain Network (chOptomuxPort0)

If you are using a B1 or B2, or an E1 or E2 in serial mode, then the communication handle is as follows. You will need to change the port number and baud rate to match your setup.



Using PAC Manager, make sure to set the value for Control for Communication Port to None.

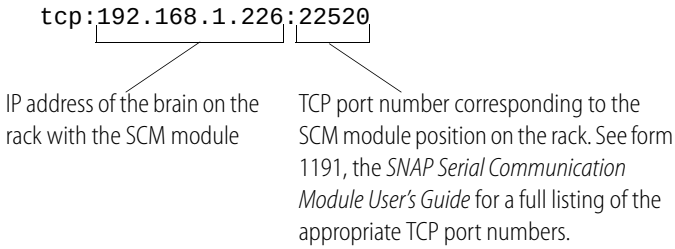
1. In PAC Manager, choose Tools→Inspect→Communications→Communication Port Control.
2. Locate the port that is connected to the Optomux device.
3. Click in the value field for Control Function for Communication Port and choose None from the drop-down list.



Using PAC Manager, set the controller's Control for Communication Port to None.

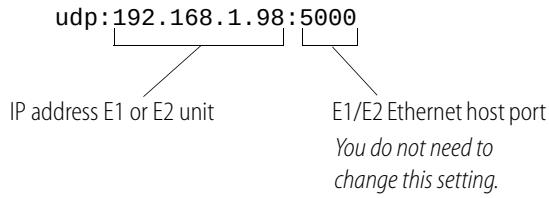
Optomux Serial Brain Network with an SCM (chOptomucSerial10A)

If you are using a B1 or B2, or an E1 or E2 in serial mode and a serial communication module (SCM) (instead of a serial port built into the controller) to connect to the Optomux serial brain network, then the communication handle is as follows:



Optomux over Ethernet (chOptomuxUDP98 and chOptomuxUDP99)

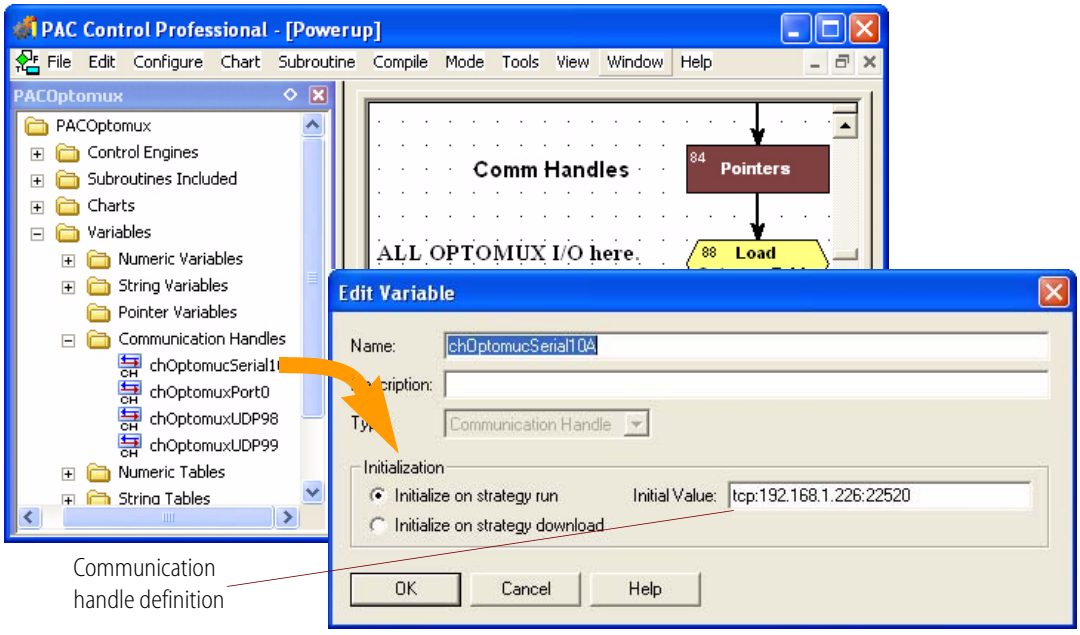
If you are using an E1 or E2 with the Optomux protocol over Ethernet, then the communication handle is as follows:



Use a separate communication handle for each E1 or E2.

Defining Communication Handles

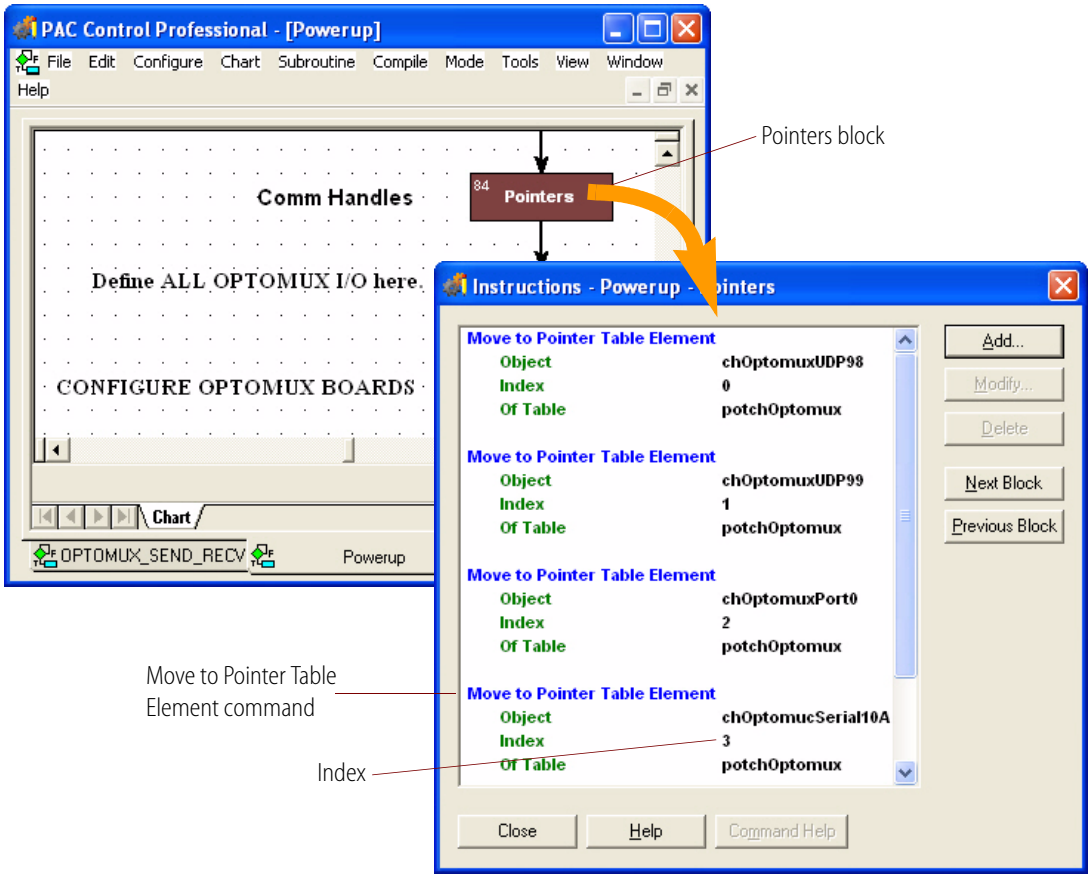
Communication handles are defined in the Strategy Tree under Communication Handles. See the previous sections for descriptions of the various types of communication handles.



For more information on communication handles, see the *PAC Control User's Guide*, form 1700.

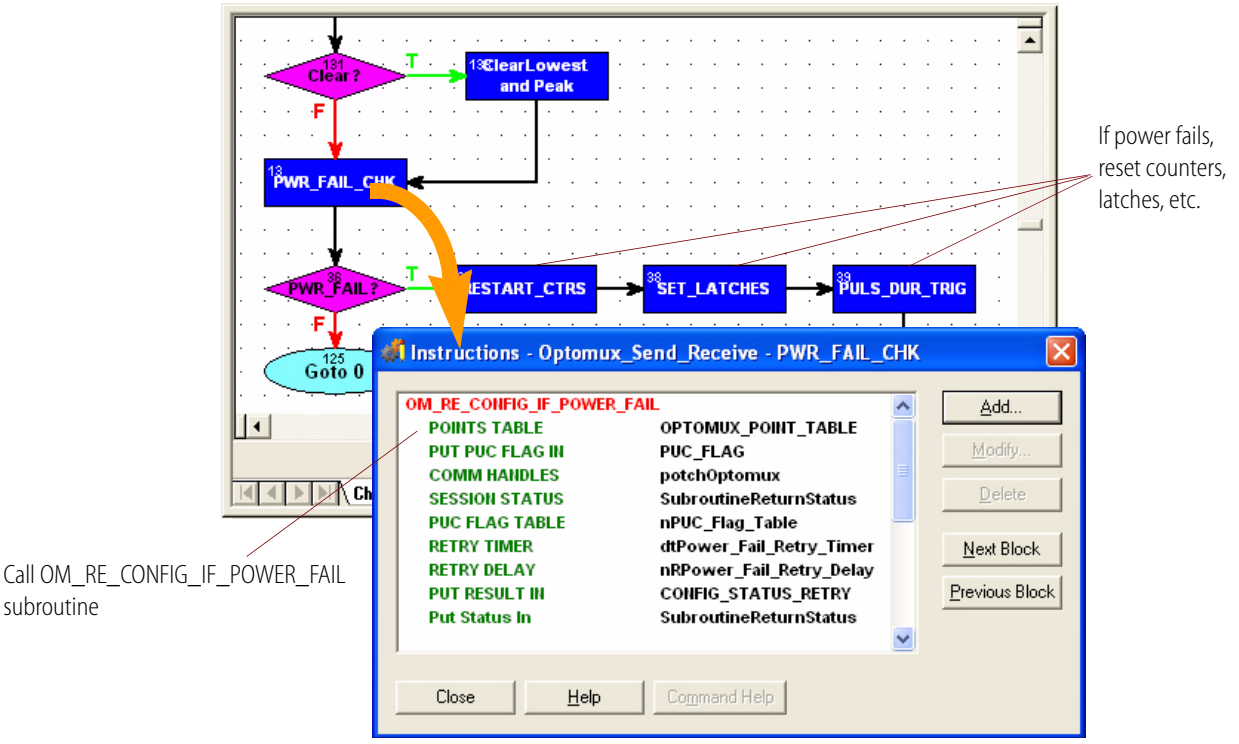
Assigning Communication Handle Index Values

In the example strategy, index values for each communication handle are assigned in the Pointers block in the Powerup chart using the Move to Pointer Table Element Command. This loads the communication handles for each port or IP address in a pointer table called *potchOptomux*. For information on using the Move to Pointer Table Element command, see the *PAC Control Command Reference*, form 1701.



Reconfiguring after a Power Failure

A block in the OPTOMUX_SEND_RECV chart in the example strategy calls the subroutine named OM_RE_CONFIG_IF_POWER_FAIL. This subroutine reconfigures all brain boards that have lost power and redefines all the I/O types.



You can use this subroutine where it is in the OPTOMUX_SEND_RECV chart, or you can place it somewhere else in the strategy where it will be executed periodically. This subroutine will not start counters or configure latches, etc., so these functions must be done separately after calling the OM_RE-CONFIG_IF_POWER_FAIL subroutine (as in the OPTOMUX_SEND_RECV chart).

Each of the command subroutines sets a flag if the command fails. If the subroutine OM_RE-CONFIG_IF_POWER_FAIL sees this flag it will send a powerup clear and the configuration string to the brain board. If the brain responds, it clears the flag. It then sets the PUC_Flag variable to true . As shown in the example strategy, the strategy should look for this flag and start counters, configure latches, and so on.

Subroutine Error Codes

Error codes are returned by most of the subroutines using the last parameter, Put Result In. It is not essential to have your code (strategy logic) check for errors unless you are having problems. When problems occur, knowing what the error codes are will help identify the root of the problem.

The subroutines will automatically take care of any issues that result from a brain powering up (Error -10, PowerUp Clear Expected).

Brain Board Error Codes

Error Code	Meaning
0	OK
-1	Undefined command
-2	Checksum error
-3	Input buffer overrun
-4	Non ASCII character received
- 5	Data field error
-6	Watchdog timeout
-7	Specified limits invalid
-10	Powerup clear expected

Subroutine Error Codes

Error Code	Meaning
-8	Incorrect point type
-9	Couldn't find engineering units (the engineering unit configuration information for that point was not available)

PAC Control Communication Driver Error Codes

Error Code	Meaning
-40	Timeout on locking port
-41	Timeout on sending message
-42	Timeout on receiving reply
-43	Not enough data returned
-44	Invalid data returned
-45	Bad checksum returned
-46	Invalid limit
-47	Received a NAK
-48	String too small
-49	String was empty
-50	Bad port number