

# OPTOMUX PROTOCOL USER'S GUIDE

Form 203-020107 – January, 2002

**OPTO 22**

43044 Business Park Drive, Temecula, CA 92590-3614

Phone: 800/321-OPTO (6786) or 909/695-3000

Fax: 800/832-OPTO (6786) or 909/695-2712

Internet Web site: [www.opto22.com](http://www.opto22.com)

**Product Support Services:**

800/TEK-OPTO (835-6786) or 909/695-3080

Fax: 909/695-3017

E-mail: [support@opto22.com](mailto:support@opto22.com)

Bulletin Board System (BBS): 909/695-1367

<http://bbs.opto22.com>

(24 hours a day, 7 days a week)



---

## **Optomux Protocol User's Guide Form 203-020107—January, 2002**

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. 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 and dry contact modules. Refer to Opto 22 form number 1042 for more details.

Opto 22 FactoryFloor, Cyrano, Optomux, Pamux, SNAP I/O, Generation 4, and Mystic are registered trademarks of Opto 22. OptoConnect, OptoControl, OptoServer, and OptoDisplay are trademarks of Opto 22.

ARCNET is a registered trademark of Datapoint Corporation.

IBM is a registered trademark and IBM PC, XT, AT, and PS/2 are trademarks of International Business Machines Corporation.

Microsoft and MS-DOS are registered trademarks and Windows is a trademark of Microsoft Corporation.

# Table of Contents

<b>Welcome</b> .....	<b>9</b>
Overview .....	9
What's In This Guide?.....	9
Document Conventions .....	10
<b>Chapter 1: Introduction</b> .....	<b>11</b>
Overview .....	11
System Configuration.....	12
Physical Layout .....	12
Communications .....	12
Modular Construction .....	12
Wiring .....	13
Data Transmission.....	13
System Throughput .....	13
Features .....	14
Digital Optomux .....	14
Latched Inputs .....	14
Time Delays.....	15
Pulse Generation.....	15
Event Counting .....	15
Pulse Duration Timers .....	15
Analog Optomux .....	15
Analog Input Averaging .....	15
High/Low Limit Testing .....	15
Output Waveform Generation .....	15
High/Low Value Recording .....	16
Gain and Offset Calculation .....	16
Summary .....	16

<b>Chapter 2: Installation and Wiring .....</b>	<b>17</b>
Physical Installation .....	17
Mounting.....	17
Communications Wiring .....	18
Host to First Optomux.....	18
Optomux to Optomux.....	19
Installing Digital I/O Modules.....	20
Installing Analog I/O Modules.....	21
Power Requirements.....	22
Connect Power Supply .....	22
Third-Party power supply vendors .....	24
Brain Board Mounting .....	25
Selecting The Jumpers .....	26
Information Required.....	26
Group A Jumpers.....	26
Group B Jumpers.....	27
Accessories .....	29
<b>Chapter 3: Programming .....</b>	<b>31</b>
Overview .....	31
Building Command Messages .....	32
Beginning .....	32
Middle .....	32
End .....	32
How Optomux Represents Numbers.....	32
Building The Beginning Of A Command Message .....	33
Building The Middle Of A Command Message .....	33
Building The End Of A Command Message .....	34
Carrying Out Message Transactions.....	35
2-Pass Mode .....	35
4-Pass Mode .....	35
Interpreting the Response .....	36
Error Codes.....	36
Acknowledge .....	37
Acknowledge with Data .....	37
Initializing An Optomux Network.....	38
Serial Port Configuration.....	39
Interpreting Analog Data .....	39
Analog Outputs .....	39
Analog Inputs.....	40
Offset and Gain .....	42

<b>Chapter 4: Command Directory .....</b>	<b>43</b>
Optomux Command Summary .....	43
COMMAND NAME/COMMAND FORMAT .....	43
Operational Differences – Old Optomux Brain boards .....	48
Digital Brain Boards (B1) .....	48
Analog Brain Boards (B2) .....	48
Command Enhancements .....	49
Digital .....	49
Analog .....	49
Command Usage Recommendations .....	50
<b>Chapter 5: Setup Commands .....</b>	<b>51</b>
POWER-UP CLEAR .....	COMMAND A 51
RESET .....	COMMAND B 52
SET TURNAROUND DELAY .....	COMMAND C 53
SET DIGITAL WATCHDOG .....	COMMAND D 54
SET ANALOG WATCHDOG .....	COMMAND D 55
SET PROTOCOL .....	COMMAND E 57
IDENTIFY OPTOMUX TYPE .....	COMMAND F 58
SET ENHANCED DIGITAL WATCHDOG .....	COMMAND m 59
SET ANALOG WATCHDOG USER-DEFINED VALUE .....	COMMAND m 60
SET TIMER RESOLUTION .....	COMMAND n 61
SET TEMPERATURE PROBE TYPE .....	COMMAND k 62
<b>Chapter 6: I/O Configuration Commands .....</b>	<b>65</b>
CONFIGURE POSITIONS .....	COMMAND G 65
CONFIGURE AS INPUTS .....	COMMAND H 66
CONFIGURE AS OUTPUTS .....	COMMAND I 67
READ MODULE CONFIGURATION .....	COMMAND j 68
<b>Chapter 7: Digital Read/Write Commands .....</b>	<b>69</b>
WRITE OUTPUTS .....	COMMAND J 69
ACTIVATE OUTPUTS .....	COMMAND K 70
DEACTIVATE OUTPUTS .....	COMMAND L 71
READ ON/OFF STATUS .....	COMMAND M 72
<b>Chapter 8: Digital Latch Commands .....</b>	<b>73</b>
SET LATCH EDGES .....	COMMAND N 73
SET OFF-TO-ON LATCHES .....	COMMAND O 74
SET ON-TO-OFF LATCHES .....	COMMAND P 75
READ LATCHES .....	COMMAND Q 76
READ AND CLEAR LATCHES .....	COMMAND R 77
CLEAR LATCHES .....	COMMAND S 78

<b>Chapter 9: Digital Counting Commands .....</b>	<b>79</b>
Start/Stop Counters .....	Command T 79
Start Counters .....	Command U 80
Stop Counters .....	Command V 81
Read Counters .....	Command W 82
Read And Clear Counters .....	Command X 84
Clear Counters .....	Command Y 85
<b>Chapter 10: Digital Time Delay/Pulse Commands .....</b>	<b>87</b>
Set Time Delay .....	Command Z 87
Retrigger Time Delay .....	Command h 89
Generate N Pulses .....	Command i 90
Start On Pulse .....	Command k 91
Start Off Pulse .....	Command l 92
<b>Chapter 11: Digital Pulse Duration Measurement Commands.....</b>	<b>93</b>
Set Pulse Trigger Polarity .....	Command a 93
Trigger On Positive Pulse .....	Command b 94
Trigger On Negative Pulse .....	Command c 95
Read Pulse Complete Bits.....	Command d 96
Read Pulse Duration Counters .....	Command e 97
Read And Clear Duration Counters .....	Command f 98
Clear Duration Counters .....	Command g 99
<b>Chapter 12: Analog Read/Write Commands.....</b>	<b>101</b>
Write Analog Outputs.....	Command J 101
Update Analog Outputs .....	Command S 102
Read Analog Outputs.....	Command K 103
Read Analog Inputs.....	Command L 104
Read And Average Inputs.....	Command M 105
Start Input Averaging .....	Command T 106
Read Average Complete Bits .....	Command i 107
Read Input Average Data .....	Command U 108
Read Temperature Inputs .....	Command l 109
Read Average Temperature Inputs.....	Command o 111
<b>Chapter 13: Analog Input Range Commands.....</b>	<b>113</b>
Set Input Range .....	Command N 113
Read Out-Of-Range Latches .....	Command O 114
Read And Clear Out-Of-Range Latches.....	Command P 115
Clear Out-Of-Range Latches .....	Command Q 116
Read Lowest Values .....	Command a 117
Clear Lowest Values .....	Command b 118
Read And Clear Lowest Values .....	Command c 119

READ PEAK VALUES .....	COMMAND d	120
CLEAR PEAK VALUES .....	COMMAND e	121
READ AND CLEAR PEAK VALUES .....	COMMAND f	122
<b>Chapter 14: Analog Gain/Offset Commands.....</b>		<b>123</b>
CALCULATE OFFSETS .....	COMMAND g	123
SET OFFSETS .....	COMMAND W	125
CALCULATE AND SET OFFSETS .....	COMMAND h	127
CALCULATE GAIN COEFFICIENTS .....	COMMAND X	129
SET GAIN COEFFICIENTS .....	COMMAND Y	131
CALCULATE AND SET GAIN COEFFICIENTS .....	COMMAND Z	132
<b>Chapter 15: Analog Waveform Commands .....</b>		<b>135</b>
SET OUTPUT WAVEFORM .....	COMMAND R	135
IMPROVED OUTPUT WAVEFORMS .....	COMMAND V	137
<b>Chapter 16: Revision Identification Commands.....</b>		<b>139</b>
CHECKSUM ON ROM .....	COMMAND _	139
DATE OF FIRMWARE.....	COMMAND '	140
<b>Appendix A.....</b>		<b>141</b>
HEX - ASCII Character Table .....		141
HEX To Binary Conversion Table.....		142
<b>Appendix B.....</b>		<b>143</b>
Implementing Drivers for Optomux.....		143
<b>Appendix C.....</b>		<b>145</b>
Surge Protection For RS-422/485 Communication Links .....		145
<b>Appendix D.....</b>		<b>147</b>
Troubleshooting Questions and Answers .....		147
<b>Product Support.....</b>		<b>151</b>





# Welcome

## Overview

Opto 22's Optomux system provides low-cost distributed access to both digital and analog I/O. Using a simple serial port with an RS-485 converter up to 4095 I/O points can be accessed over a distance of several thousand feet at moderate speeds with high reliability.

This manual describes the protocol used by Optomux to aid in implementing Optomux driver or custom hardware that uses Optomux protocol. Commands supported by Optomux digital and analog I/O are described.

C source code called 'Generic Optomux Driver' is available in the OptoDriver Toolkit. This driver is helpful for those developing their own Optomux driver. Optomux drivers are available in the OptoDriver Toolkit for DOS, Win16 and Win32.

*NOTE:* Information in this manual relating to Pamux brain boards applies only to the revisions of the brain boards indicated below:

- AC28: Revision C or later. For revision B or earlier, request Opto 22 form #218.
- B4: Revision L or later. For revision K or earlier, request Opto 22 form #127.
- B5: Revision J or later. For revision I or earlier, request Opto 22 form #145.
- B6: Revision G or later. For revision F or earlier, request Opto 22 form #154.

## What's In This Guide?

This guide includes the following sections:

**Chapter 1, "Introduction"**—overview of Optomux Protocol and hardware.

**Chapter 2, "System Setup"**—Optomux communication wiring, power wiring, installation, and jumpers.

**Chapter 3, "System Setup"**—description of protocol.

**Chapter 4, "System Setup"**—Command directory — Index of Optomux commands.

**Chapter 5, "Setup Commands"**

**Chapter 6, "I/O Configuration Commands"**

- Chapter 7, “Digital Read/Write Commands”**
- Chapter 8, “Digital Latch Commands”**
- Chapter 9, “Digital Counting Commands”**
- Chapter 10, “Digital Time Delay/Pulse Commands”**
- Chapter 11, “Digital Pulse Duration Measurement Commands”**
- Chapter 12, “Analog Read/Write Commands”**
- Chapter 13, “Analog Input Range Commands”**
- Chapter 14, “Analog Gain/Offset Commands”**
- Chapter 15, “Analog Waveform Commands”**
- Appendix A, “Hex/Binary Conversion Tables”**
- Appendix B, “Implementing Drivers for Optomux”**
- Appendix C, “Surge Protection For RS-422/485 Communication Links”**
- Appendix D, “Troubleshooting Questions and Answers”**
- Appendix E, “Product Support”**—information on how to get help from Opto 22.

## Document Conventions

- **Bold** typeface indicates text to be typed. Unless otherwise noted, such text may be entered in upper or lower case. (Example: “At the DOS prompt, type **cd windows.**”)
- *Italic* typeface indicates emphasis and is used for book titles. (Example: “See the *OptoControl User’s Guide* for details.”)
- File names appear in all capital letters. (Example: “Open the file TEST1.TXT.”)
- Key names appear in small capital letters. (Example: “Press SHIFT.”)
- Key press combinations are indicated by hyphens between two or more key names. For example, SHIFT-F1 is the result of holding down the SHIFT key, then pressing and releasing the F1 key. Similarly, CTRL-ALT-DELETE is the result of pressing and holding the CTRL and ALT keys, then pressing and releasing the DELETE key.
- “Press” (or “click”) means press and release when used in reference to a mouse button.
- Menu commands are sometimes referred to with the Menu → Command convention. For example, “Select File → Run” means to select the Run command from the File menu.
- Numbered lists indicate procedures to be followed sequentially. Bulleted lists (such as this one) provide general information.

# Introduction

## Overview

Optomux is a protocol used by a family of intelligent digital and analog I/O (input/output) units that operate as slave devices to a host computer.

An Optomux I/O unit is a 4-, 8-, or 16-point assembly that accommodates optically-isolated analog or digital I/O modules. Each Optomux I/O unit consists of a removable brain board and an I/O mounting rack. The removable brain board contains a microprocessor which communicates with the host computer and controls the plug-in I/O modules located on the I/O mounting rack.

There are two types of Optomux units: analog and digital. Any combination of analog I/O modules may be plugged into an analog Optomux unit and likewise, any combination of digital I/O modules may be plugged into a digital rack. Analog and digital Optomux units from the SNAP, brick, and Classic rack families can then be combined on the same serial link providing endless combinations of analog and digital I/O points.

Optomux I/O units communicate with the host computer over an RS-485 serial communications link. The serial data link is composed of two twisted pairs and a ground (5 wires) that connect to each Optomux I/O unit. It is possible to communicate with up to 256 individual units on a single serial data link for a total of 4,096 digital and analog I/O. This requires configuring all units (B1's and B2's) in Repeat mode. However, this can also be accomplished with all units (B1's, B2's, and B3000's) in Multidrop mode, using a repeater (AC30) after every 32 nodes. Note: the B3000 configured for Optomux only supports Multidrop mode.

The RS-485 serial data link offers excellent noise immunity and long cable lengths. This can dramatically reduce wiring costs by eliminating the need for bringing large bundles of field wiring back to a central control location. Optomux units can be located where the actual control is needed.

This manual describes format of the actual data sent to and received from the Optomux protocol brain board. Knowledge of the details of the Optomux data transmission can be helpful in analyzing the data created by the Optoware software driver for direct communication.

# System Configuration

## Physical Layout

The high cost of electrical wiring and the noise susceptibility of analog signals make it desirable to place the control or monitoring point as close to the controlled device as possible. Optomux offers design flexibility, with as few as 4 points or as many as 4,096 points in one physical location. Each Optomux I/O unit must be assigned its own unique address, 0 through 255.

Considerable installation savings and improved system performance can be realized by placing the control point (Optomux) close to the controlled device. The serial data link between adjacent Optomux I/O units consists of two twisted pairs and a common which makes it practical to have an Optomux located at each machine on a factory floor, at each heating or air conditioning unit in an energy management application, or at each instrument in a data acquisition environment.

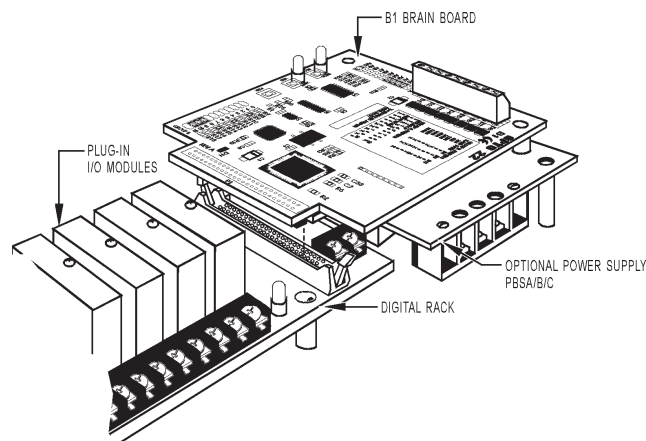
## Communications

Optomux I/O units can be configured to operate in multidrop or repeat mode via jumpers located on each Optomux protocol brain board. For more information on communications, wiring, and system layout, please refer to the individual data sheets for Optomux protocol brain boards.

- Form #463            B1 Digital Brain Board Data Sheet
- Form #464            B2 Analog Brain Board Data Sheet

## Modular Construction

Each Optomux unit is composed of two components, a brain board and a module mounting rack. Optomux I/O units using the open mounting rack technology use a B1 brain board. The B1 connects to the rack via a 50-pin header connector. The following figure shows a typical B1/mounting rack combination assembly.



**Figure 1-1: Brain Board and Rack Assembly**

## Wiring

Inputs and outputs are connected to the Optomux mounting rack via an industrial barrier strip with integral cable clamp for each terminal. Spade lugs, other wire terminations, bare wires, and tinned wires are all readily accepted. Serial link connections and power connections are also made via screw terminals.

## Data Transmission

Optomux supports 8 baud rates (300, 600, 1,200, 2,400, 4,800, 9,600, 19.2K, and 38.4K) which are selectable via jumpers located on the Optomux unit. Selection of baud rate often depends upon the capabilities of the host port. Many host computers are limited to 9,600. Modems and radio links usually operate at either 300 or 1,200 baud. System throughput is increased by using the fastest available baud rate.

Optomux is capable of using two types of message protocols. The 2-pass protocol is intended for use during normal operation. This protocol requires the transmission of two messages on the serial link every time a command is executed. The host sends a command to an Optomux unit and then receives a response message acknowledging successful execution of the command along with any requested data, or an error message indicating that the Optomux detected an error in the command message and was unable to execute it.

A 4-pass protocol is also available. This protocol is sometimes useful during initial setup and installation because it allows the host to examine and display the command message that Optomux received. The command message is echoed back to the host by Optomux for verification, before it can be executed. After the host verifies that the two messages are identical, an execute command gives Optomux the go ahead, the required action is performed, and any requested data is returned.

To ensure secure data transmission, every Optomux command message, and every response from Optomux which contains data, includes a calculated checksum. Optomux never executes a command containing a checksum error.

## System Throughput

In most control applications, some points need to be updated more frequently than others. This section will help you estimate the minimum timing between successive events or transactions.

The total time required to complete an Optomux instruction is approximately equal to the transmission time of all the characters in the instruction, plus all the characters in the response. The time required for digital Optomux units to respond to a command is typically less than 1 millisecond. Typical analog Optomux response times are from 3 to 7 milliseconds.

A total of 15 characters are required to read the ON/OFF status of all 16 positions on a digital Optomux unit. The command sent from the host requesting the status consist of 7 characters; the response from Optomux consists of 8 characters. The following table shows system throughput assuming that it takes 1 millisecond for Optomux to respond to the host command.

**Table 1-1: Optomux System Throughput**

Baud Rate	ms/ Character	Char/ Second	Digital Positions/ Second	Analog Positions/ Second
300	33.3	30	32	6
1,200	8.5	120	124	24
9,600	1.0	1,000	1,000	186
19.2K	0.5	2,000	1,882	344
38.4K	0.25	4,000	3,368	598

Reading the values of 16 analog inputs on an analog Optomux unit requires a total of 79 command/response characters. The command sent by the host consists of 11 characters; the response from Optomux consists of 68 characters. If we assume a response time of 7 milliseconds, the following table shows system throughput based upon reading all 16 positions per transaction.

## Features

System performance and throughput are increased by allowing the Optomux I/O unit to do as much processing as possible, thereby reducing serial line activity and host computer processing. The following summary of Optomux capabilities will help you determine which tasks can be performed by Optomux.

# Digital Optomux

In addition to ON/OFF control, digital Optomux units provide:

## Latched Inputs

Any or all input positions can be used to record momentary events by functioning as latches. Each input position can be configured by host command to latch on either OFF-to-ON or ON-to-OFF transitions. The host computer can recall and/or clear the status of these latches.

## Time Delays

Any or all output positions can function in time delay mode. Outputs can be set to operate with four types of delays.

- Delay before turning OFF
- Delay before turning ON
- Pulse ON
- Pulse OFF

Time delays are programmable with a resolution of 10 milliseconds.

## Pulse Generation

Optomux can be instructed to output a specific number of pulses (with programmable period) at any output position. Continuous square waves can also be generated.

## Event Counting

Any or all input positions can function as event counters to return a count of external events. The count ranges from 0 to 65,535. Each of the event counters can be individually read, stopped, and cleared. Frequencies of up to 400 Hz with a minimum duty cycle of 50 percent can be counted.

## Pulse Duration Timers

Any or all of the input positions can function as pulse duration timers. Either ON or OFF pulses can be timed with a resolution of 10 milliseconds.

# Analog Optomux

In addition to simple input and output, analog Optomux units perform:

## Analog Input Averaging

Optomux can be instructed to average the values of successive readings.

## High/Low Limit Testing

Input ranges can be established for input positions causing flags to be set when either high or low limits are exceeded. These flags can be read and/or cleared by the host computer at any time.

## Output Waveform Generation

Square waves, triangle waves, and ramps can be generated at any of the 16 module positions with programmable rates.

## High/Low Value Recording

Optomux can be instructed to remember the highest and lowest readings at input positions for recall by the host computer.

## Gain and Offset Calculation

Optomux can be instructed to calculate and set offsets and gain coefficients for input positions allowing for easy calibration under software control.

## Summary

In laying out your installation, plan to route the RS-422 data link cable to all points where you need to install Optomux I/O units now and to all points where you may need control in the future.

Make provision to supply +5 VDC to each Optomux location, preferably with a local power supply. For repeat mode systems, make it difficult for anyone to accidentally remove the power or data cables from the repeat mode Optomux controllers. For analog Optomux units, make provisions to supply +15 and -15 VDC to power the analog I/O modules.

Choose a baud rate and protocol compatible with your host computer and make this information available to the hardware installer. They will need it to set the jumpers on each controller.

Assign a unique address to each controller. Addresses may appear in any order from beginning to end of the data link, however every address must be unique. No two controllers may share the same address. There is no requirement for consecutive addresses. The entire range of addresses may be used. Try to relate the address to a location or function.

Outline the functions of the application software. Take advantage of Optomux's processing capability to off-load host computer processing time and to reduce data link activity.



# Installation and Wiring

## Physical Installation

### Mounting

The Optomux controller can be mounted in any attitude on any flat surface. The mounting rack and removable brain board portion of the Optomux unit are supplied with permanently-attached standoffs. All of the standoffs should be secured for maximum physical strength. Be sure to leave sufficient space between adjacent controllers for the I/O wiring.

To ensure reliable and trouble-free communications, the following is recommended:

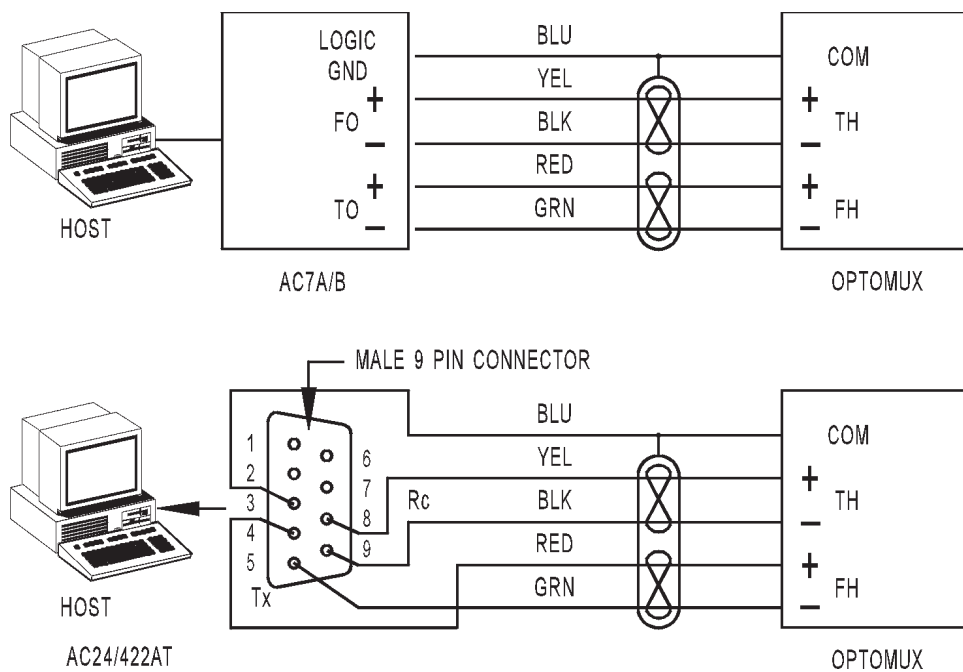
- Twisted pair wires must be used for the communications wiring. Typical wire types are Belden p/n #8,162, #9,729 (2 pair) or Belden p/n #8,164, #9,728 (4 pair) or equivalent.
- The communication and DC power wiring should be routed or kept separate from any high voltage field wiring.

# Communications Wiring

A complete data link connection at each Optomux controller (except the last in a string) consists of 10 wires; two twisted pairs (4 wires) and a common coming from the computer or previous Optomux, and two twisted pairs and a common going to the next Optomux.

## Host to First Optomux

The following figures show two possible connections from the host to the first Optomux on the serial link. A suggested color code is shown to help avoid wiring errors.

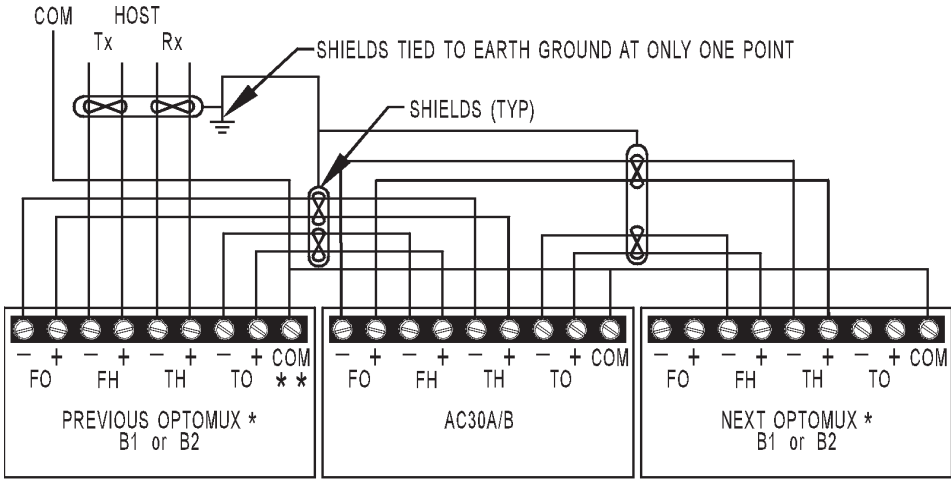


**Figure 2-1: Communication Wiring Host to Optomux**

## Optomux to Optomux

The following figure shows the connection between adjacent Optomux controllers on the serial link. The Optomux unit at the end of the data link has only 5 wires — “To Host,” “From Host,” and common.

*HINT:* Always think of the previous Optomux as the host.



\* B2 COMMUNICATION CONNECTIONS ARE MADE TO THE RACK (PB4AH, PA8AH OR PB16AH), NOT TO THE BRAIN BOARD.  
\*\* DO NOT CONNECT ANY "COM" POINT TO EARTH GROUND.

**Figure 2-2: Optomux to Optomux Communications Wiring**

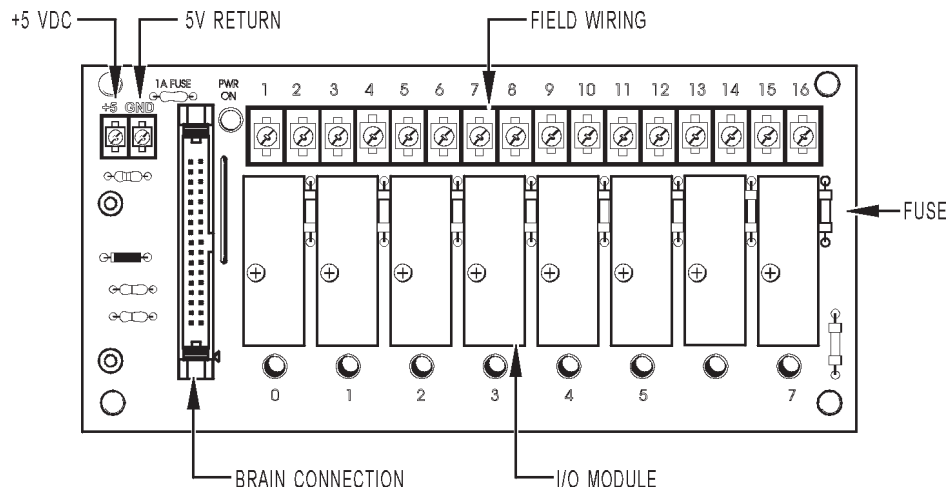
## Install Digital I/O Modules

**CAUTION:** Be sure that all Optomux power and power to the controlled devices is removed before installing or removing I/O modules. Each position on a digital mounting rack can accept either input or output modules. Install the color-coded power I/O modules which perform the required functions on the I/O mounting rack by inserting the module pins in the sockets on the rack. Secure the modules with the captive Phillips-head screw.

Each power I/O module is connected to two barrier strip terminals on the I/O mounting rack. When connecting DC loads or input, the lower-numbered terminal is always more positive.

For detailed information concerning module specifications and wiring, refer to the Opto 22 Optomux and Optomux Support Products Data Book (Form #524). Digital Optomux units only accept modules with 5-volt logic (i.e. IDC5).

Each digital I/O mounting rack is supplied with an individual 5-amp fuse for each module position. The fuse is installed in a pair of sockets, and may be removed with needle-nose pliers.

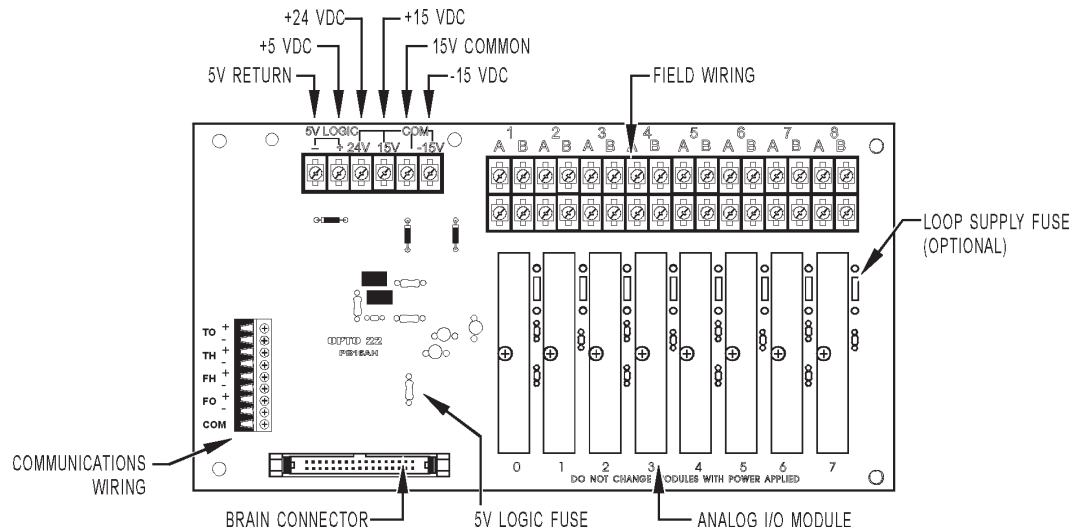


**Figure 2-3: Digital Mounting Rack**

# Install Analog I/O Modules

**CAUTION:** Be sure that all Optomux power and power to the controlled devices is removed before installing or removing I/O modules. Installing or removing modules with power applied can destroy the module.

Each position on an analog mounting rack can accept either an input or an output module. Install the I/O module which performs the required function. Secure the module with the captive Phillips-head screw. Refer to the Opto 22 Optomux and Optomux Support Products Data Book (Form #524) for detailed specifications and wiring diagrams for each of the analog I/O modules.



**Figure 2-4: Analog Mounting Rack**

## Power Requirements

DC power must be provided to each Optomux unit. All Optomux boards require +5 volts DC ( $\pm 1$  VDC) at 0.5 amps.)

Analog controllers also require +15VDC and -15 VDC ( $\pm 25$  VDC). The amount of power required is dependent upon the type and number of analog I/O modules that are plugged into the Optomux unit. The analog Optomux board requires +15 and -15 volts at 10 milliamps. Power requirements for each of the analog modules are included in the module specifications. To determine what size power supply is needed add the requirements for each module to the 10 milliamps required by the Optomux mounting rack.

Analog racks also provide terminals for a separate +24 volt supply to be used when powering a 4 - 20 mA loop using 4 - 20 mA analog I/O modules. For this type of application, the +24 volt supply is required in addition to the supplies mentioned above. Refer to the Optomux and Optomux Support Products Data Book (Form #524) for information on wiring AD3 and DA3 modules with a loop supply.

The current requirements given for the output modules are only for the modules themselves. Load requirements must be added to these to determine total power supply requirements.

Although it is possible to distribute DC from a common power supply to several locations, better noise immunity is obtained by having separate power supplies at each physical location where Optomux is installed.

A +5 VDC power supply option can be used to provide power to digital Optomux units. The power supply attaches to the I/O mounting rack beneath the removable brain board portion of the Optomux unit.

### Connect Power Supply

Connect the 5-volt power to the barrier strip connectors marked "+ 5V" and "GND" on the mounting rack. If the + 5 volt supply is to be used by more than one Optomux or other devices, make sure there is 5 VDC ( $\pm 1$  V) at each set of terminals on each rack. The communications wiring and the +5-volt and  $\pm 15$  volt wires should be routed away from any high voltage field wires. There should only be one "earth" ground connection per network, typically connected at the host site. If the "earth" ground connection is at the host site, make sure none of the supplies are connected to "earth" ground. This method of grounding prevents ground loop problems due to offset voltages appearing between multiple ground points.

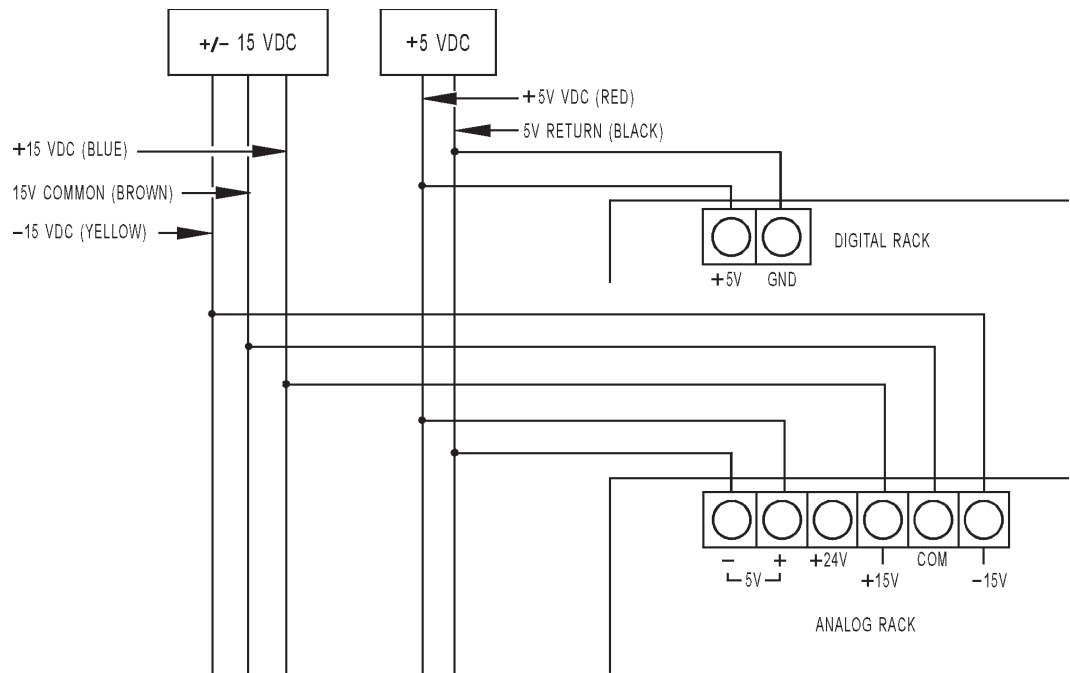
If an Opto 22 PBSA/B/C power supply is used with the digital racks, the + 5 VDC logic connection is made by the supply when it is screwed to the rack. In this case, the only connection to be made is the 120 VAC (220 VAC or 10-28 VDC depending on supply type) connection to the PBSA (PBSB or PBSC) supply.

Separate or combined + and - 15 VDC ( $\pm 25$  VDC) supplies can be used to provide power to analog I/O modules. If using combined supplies, make sure that the 5 VDC RETURN line is separate from the 15-volt COMMON line. Otherwise, the isolation of the analog modules will be defeated.

**CAUTION:** Check polarities of all power supply connections before applying power to the Optomux controller. Incorrect polarity may cause damage to the Optomux controller.

Use a consistent color code from power supply to all Optomux controllers to prevent wiring errors. Size 18 AWG is recommended for power supply wiring. Optomux demo equipment uses the following colors:

- Red +5 Volts
- Black GND
- Blue +15 Volts
- Yellow -15 Volts
- Brown GND



**Figure 2-5: Power Supply Wiring**

**WARNING:** If brain boards are powered up and serial lines are left floating, or are connected to RS-485 adapter card without proper biasing, the RX and TX LED lights may float to an unknown state. This may cause confusion as to the actual state of the communication lines and the proper operation of the brain board. We recommend checking to make sure that the communication lines have the proper biasing and termination resistors installed at the two ends of the RS-485 network.

The following is a list of a few of the many power supply vendors:

**COMPUTER PRODUCTS INC.**

Power Conversion Group  
2900 Gateway Dr.  
Pompano Beach, FL 33069  
(305)974-5500

**ELPAC POWER SYSTEMS**

3131 S. Standard Ave.  
Santa Ana, CA 92705  
(714)979-4440

**POWER-ONE**

740 Calle Plano  
Camarillo, CA 93010  
(805)987-8741

**POWER GENERAL**

152 Will Drive  
P.O. Box 189  
Canton, MA 02021  
(617)828-6216

**SOLA**

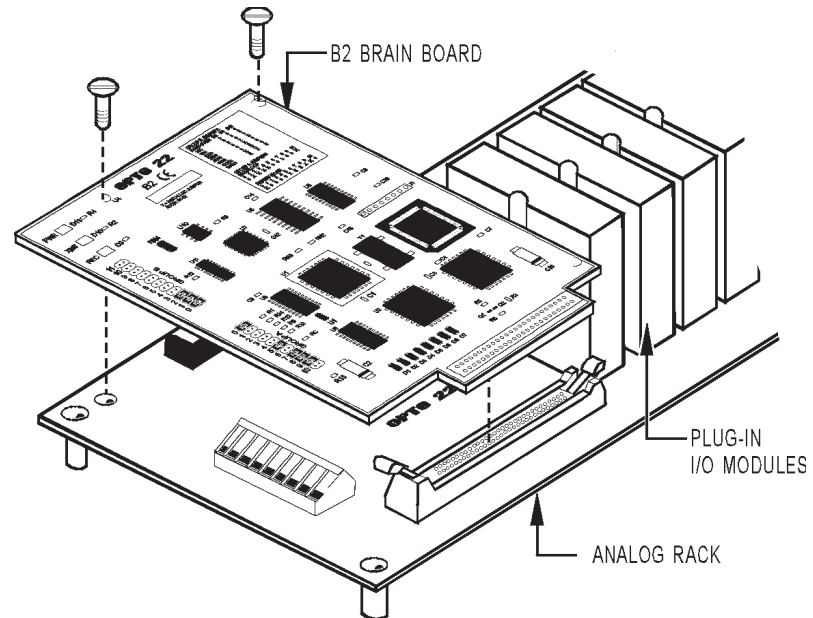
1717 Busse Hwy.  
Elk Grove Village, IL 60007  
(312)439-2800

*NOTE:* When specifying power supplies for powering Optomux analog units, some manufacturer's triple supplies have the +5 volts and the  $\pm 15$  volts commons connected; thereby defeating the isolation. To ensure complete isolation, use separate power supplies for the +5 volts and the  $\pm 15$  volts.



## Brain Board Mounting

The brain board mates with the header connector on the rack. The orientation of the digital brain board (B1) should be such that the brain board extends away from the rack. The orientation of the analog brain board (B2) is such that the brain board covers up the communications and power wiring on the analog rack when it is plugged in.



**Figure 2-6: Analog Brain Board and Rack Assembly**

# Selecting The Jumpers

*TIP:* Use the “OptoScan” utility as an aid in setting jumpers.

## Information Required

There are two groups of jumpers located on each Optomux brain board. The jumpers are labeled Group A and Group B.

### Group A Jumpers:

These jumpers route wiring for repeat-mode or multidrop communications and also provide the proper termination and biasing of the RS-422/485 network. All Optomuxes on the same network should be configured for either multidrop or repeat mode, but not both.

### Group B Jumpers:

This group of jumpers selects the Optomux address and baud rate. Each Optomux on the same link must have a unique address (different from all others). All Optomuxes on the same network should be set for the same baud rate.

The following information is required to set the jumpers on a single Optomux unit.

Optomux Address (0 - 255) \_\_\_\_\_

What is serial baud rate? \_\_\_\_\_  
(300; 1,200; 9,600; 19.2K; 38.4K)

Multidrop or Repeat mode \_\_\_\_\_

The following pages describe each individual jumper setting and should be used as a guide for configuring an Optomux unit as specified above.

### Group A Jumpers

For multidrop operation, verify that the jumpers in group A are installed as follows:

X = INSTALLED

:	X	X	X	X	:	:	:	:	:	X
0	1	2	3	4	5	6	7	8	9	10

Install jumper A0 and A6 if the Optomux is the last controller in the serial data link.

For repeat mode settings all Optomux bus should be jumpered as follows:

X	:	:	:	:	X	X	X	X	X	:
0	1	2	3	4	5	6	7	8	9	10

## Group B Jumpers

### Baud Rate Selection

Jumpers B8, B9, and B11 select the baud rate. Possible configurations are shown below.

**Table 2-1: Baud Rate Jumper Settings**

Baud Rate	B 8	B 9	B 11
300	Not Installed	Not Installed	Not Installed
600	Not Installed	Not Installed	Installed
1200	Installed	Not Installed	Not Installed
2400	Installed	Not Installed	Installed
4800	Not Installed	Installed	Installed
9600	Not Installed	Installed	Not Installed
19200	Installed	Installed	Not Installed
38400	Installed	Installed	Installed

### Protocol Selection

Jumper B10 selects between 2-pass and 4-pass communications protocol. The 2-pass protocol is the recommended protocol because it provides faster and more efficient communications. The 4-pass protocol is mainly for use as a troubleshooting aid. When jumper B10 is installed, 2-pass protocol is selected. Optomux can be put into a 4-pass mode, by removing the B10 jumper or using the Optomux software command "E."

### Address Selection

Jumpers B0 through B7 select the Optomux address. Each Optomux on the same network must have a unique address (different from all the others). Addresses are arbitrary and do not need to be sequential (although they must be unique). The following table lists the possible addresses with corresponding jumper configurations. To use the table, find the address you wish to use (0 to 255), to the right of the address is the sequence of jumpers in the following order: B7, B6, B5, B4, B3, B2, B1, and B0.

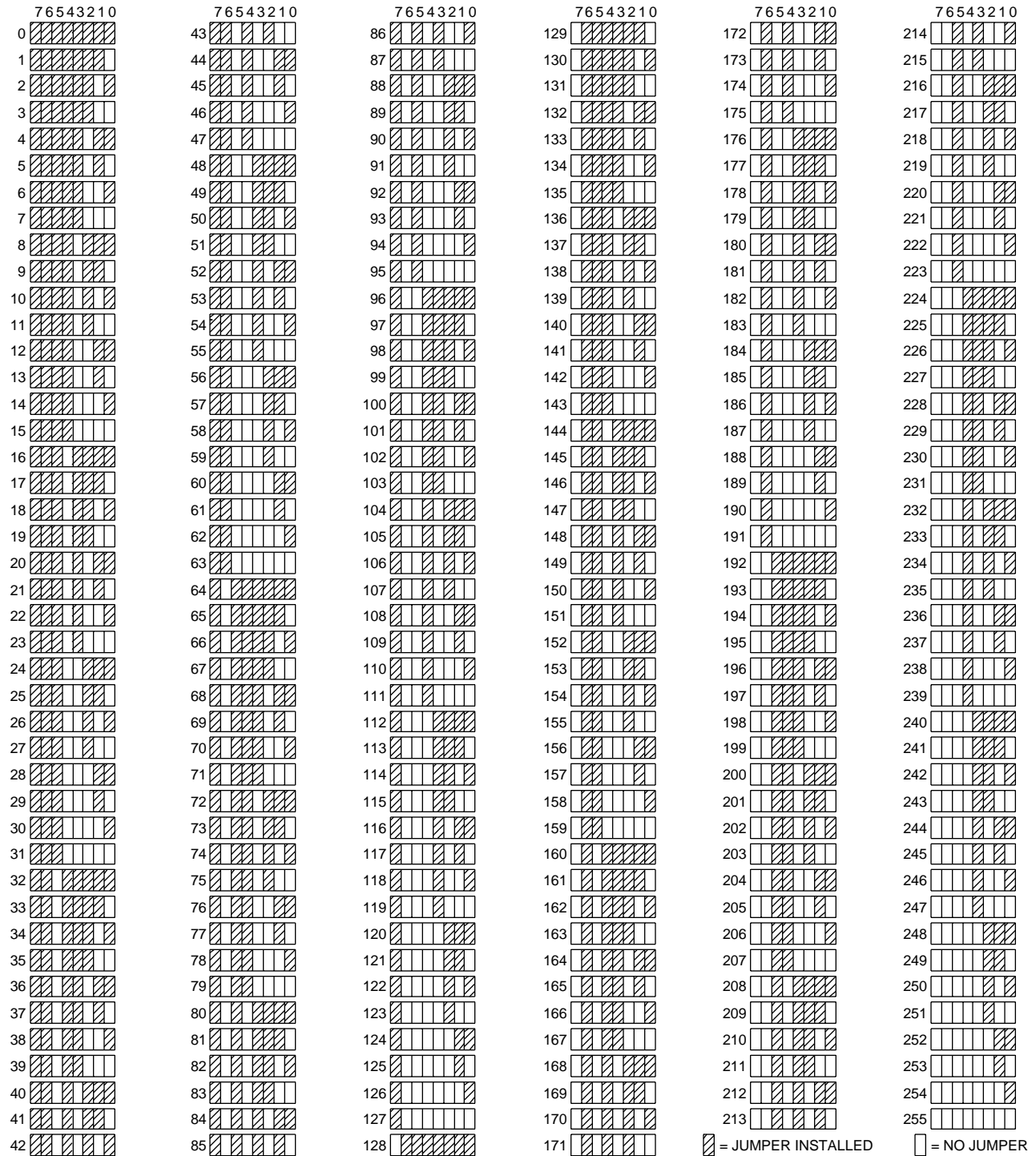


Figure 2-7: Jumper Addresses

# Accessories

There are several accessories to the Optomux family, ranging from power supplies to adapter cards. Several are listed below. Please refer to the Opto 22 Optomux and Optomux Support Products Data Book (Form #524) for detailed information on the following products.

**PBSA, PBSB, PBSC:** The PBSX series of devices are 5 VDC power supplies that mount directly to digital I/O racks that have a header connector. The power supply mounting allows an Optomux brain board to be mounted above the supply, thereby taking up no extra mounting space. The PBSA is for 120 VAC operation, the PBSB is for 240 VAC and the PBSC is for 12/24 VDC operation.

**AC7A, AC7B:** This adapter card converts any full duplex RS-232 port to an RS-422/485 port, allowing any computer with a serial RS-232 port to communicate to an Optomux network. The AC7A is for 120 VAC operation and the AC7B is for 240 VAC operation.

**AC24, AC24AT:** The AC24 is an optically-isolated RS-422/485 adapter that plugs directly into the IBM PC Bus backplane. This card can be configured as COM1, COM2, COM3, or COM4. The AC24AT version is for use with PC/AT type machines or compatibles.

**AC422, AC422AT:** This adapter is similar to the AC24 except that it is smaller in size and is not optically isolated.

**AC8:** This adapter converts a half duplex RS-232 serial port to a full duplex RS-422/485 port. The adapter is used for connecting half duplex radio modems to the Optomux network.

**AC30:** This adapter is an RS-422/485 repeater, used to extend the 5,000 foot limit of an Optomux multidrop network.

**AC31:** This adapter is an intelligent network interface for the Optomux network. It allows serial devices (RS-232 or RS-422/485) that are not addressable to reside on the Optomux network as slaves to a host computer. Devices may include printers, terminals, barcode readers, counters, motion controllers, etc.

**AC32:** This adapter is a dual RS-422/485 interface card specifically intended for the IBM PS/2 Micro Channel Bus. The AC32 can be configured as COM1 or COM2 on the IBM Model 50, 60, or 80 computers.

**AC34:** This adapter is an optically-isolated single-channel RS-422/485 plug-in card specifically intended for the IBM PS/2 Micro Channel Bus.

**LC2/LC4:** The LC2 and LC4 Local Controllers are single board computers that can be used to communicate and control an Optomux network. These controllers can act as a standalone replacement for the host computer, or can be connected in a multidrop network as slaves to a supervisory host computer. The LC2 can be programmed in BASIC or FORTH, and the LC4 can be programmed in BASIC, FORTH, or PARAGON LC.

*NOTE:* Micro Channel PC, PC/XT, PC/AT are trademarks of International Business Machines Corporation. Optomux, LC2, LC4, and PARAGON LC are trademarks of Opto 22.

## Factory Support

Opto 22 maintains a staff of application engineers at our Temecula, California facility to provide “instant information” on configuration and operation of Optomux.

Call our toll-free number weekdays between 8:00 a.m. and 5:00 p.m. Pacific time.

1-800-TEK-OPTO (800/835-6786)

# Programming

## Overview

An Optomux I/O unit is an intelligent device that acts as a slave device to a host computer. The host computer issues instructions to Optomux by sending command messages over the serial communications link. Optomux responds to the host by sending messages back.

All messages between the host computer and Optomux are made up of ASCII (American Standard Code for Information Interchange) characters. Appendix A.1 will help you become familiar with this code. The ASCII code numbers which represent upper and lowercase letters and punctuation are standardized throughout the world and are the only characters Optomux recognizes.

In normal operation, Optomux uses a 2-pass protocol with a checksum for message transactions. This protocol requires 2 distinct messages between the host computer and the specified Optomux. The host computer initiates the transaction by sending a command to one of the Optomux units in the network. Although all Optomux I/O units receive the message, only the addressed Optomux will verify that the command is valid and then execute the command. Upon command completion, Optomux returns an acknowledgment along with any requested data. A 4-pass mode is available for diagnostics.

Command messages and returned messages from Optomux containing data contain a message checksum to ensure secure communications. Each ASCII character has an associated number value (the decimal number 65 stands for the letter A, for example). The message checksum is calculated by adding up the number values that represent all the characters in a message. This sum is converted into 2 ASCII characters and is appended to the end of the message. A simple "Acknowledge" does not use a checksum.

The host computer calculates the checksum and sends it along as part of the Command Message to Optomux. When Optomux receives the message, it calculates its own checksum and compares that value with the transmitted checksum. If they match, Optomux can be sure the message was received correctly.

The same procedure is repeated whenever Optomux returns data to the host computer. When a message is received from Optomux, the host computer calculates the message checksum and compares against the checksum that was transmitted as part of the message.

*NOTE:* For those who will be using a high level language such as BASIC, C, or PASCAL on an IBM-PC to communicate with Optomux, an Optomux communications driver is available from Opto 22. The communications driver is called Optoware and includes an assembly language driver with source code, a command tutorial, and a rich set of utilities with examples and diagnostics. The driver takes care of building command messages, calculating the checksums, and processing the response.

# Building Command Messages

All Optomux command messages consist of three main parts as described below:

## Beginning

Always contains the Start Of Message character (>) followed by 2 characters representing the address of the intended Optomux I/O unit.

## Middle

Contains from 1 to 4 fields as shown below. The fields shown in square brackets are not required by some commands.

```
cmd [positions] [modifier] [data]
```

### **cmd**

This field always contains a single character which specifies the command to be executed.

### **positions**

This field is required by some commands to specify which positions are to be affected.

### **modifier**

This 1 or 2 character field is required by some commands to further specify command execution (tells what type of time delay to set).

### **data**

This 1 to 64 character field is required by some commands. It is used to specify analog output values, etc.

## End

Always contains three characters. Two characters representing the checksum followed by a carriage return.

## How Optomux Represents Numbers

Optomux uses the hexadecimal (hex) numbering system to represent numbers in commands and responses. If you are not familiar with hex, Appendix B provides an explanation of binary and hex numbering systems.

Optomux command messages are transmitted as a series of ASCII characters. Numbers are transmitted as the ASCII number characters 0 through 9 and the uppercase ASCII characters A through F. We will refer to an ASCII character representing a hex number as "ASCII-hex."

*EXAMPLE: 15 (decimal) = F (hex) and is transmitted as the ASCII "F" character.*



## Building The Beginning Of A Command Message

All Optomux command messages start with the START OF COMMAND CHARACTER ">", followed by two ASCII-hex digits representing the address of the intended Optomux I/O unit (00-FF). If devices other than Optomux are to communicate on the same serial communications link along with Optomux, the beginning of message character ">" cannot appear in any communication between the host computer and the other device(s).

## Building The Middle Of A Command Message

The middle section of Optomux command messages contains from 1 to 4 possible fields. The first field always contains the command character. This character specifies which command is to be executed. Optomux Analog uses the uppercase ASCII characters A through Z and lower case a through i.

The remainder of the middle section differs depending upon the command. There are 3 possible fields.

### Positions Field

Most commands require the positions field to specify the module positions that are to be effected. The positions field contains from 1 to 4 ASCII-hex digits. Each of these digits affects a group of four module positions.

A single hexadecimal digit can represent a 4-bit binary number (0000 through 1111); we can specify all the possible combinations of 0s and 1s for 4-module positions with 1 hex digit.

Please refer to page 140 for a list of Hex-ASCII to binary conversions.

Optomux converts each hex character appearing in the positions field to its 4-bit binary equivalent and uses each bit to specify one particular module position.

Each ASCII-hex character within the positions field corresponds to four module positions on the Optomux unit. The following diagram illustrates the relationships between each positions field character and the module effected.

*NOTE:* The least significant bit corresponds to the least significant channel; i.e., bit 0 corresponds to channel 0.

#### Example #1 Positions Field Contains A5F0

*15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	0
A				5				F				0			

\* Optomux channel number

With four characters in the Positions field, the entire range of 16 module positions will be specified.

**Example #2**  
**Positions Field Contains 1E**

*15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0
								1				E			

\* Optomux channel number

With only two characters in the positions field, only module positions in the range 0–7 will be affected by the command.

### Modifier Field

The next possible field in the middle section of the command is the modifier field. This field is required by commands that have many possible execution features.

For example, the Set Time Delay command requires a modifier field to indicate what type of time delay is to be used. If required, this field contains a single ASCII character.

### Data Field

The last field in the middle section of the command message is the DATA field. Some of the Optomux commands require that a value or list of values be specified. This information is contained in the DATA field.

## Building The End Of A Command Message

All Optomux command messages end with 2 ASCII-hex digits representing the message checksum, followed by a carriage return. These digits are appended to the command message and a carriage return character is put on as the last character in the command message. The "." character can be used in place of the carriage return (cr). For debugging purposes, when using a terminal, 2 "?" characters can be used in place of the checksum characters. The ?? is a wildcard checksum and should not be used in the final application because it defeats the purpose of checksum verification for message integrity.

The message checksum is computed by adding the decimal values of all the ASCII characters in the message EXCLUDING the START OF COMMAND CHARACTER ">". This sum is then divided by 256 and the integer remainder is converted to 2 ASCII-hex digits.

### Example:

```
>08KC01289cr is a valid command.
08KC012 is the part used to calculate the checksum.
```

*NOTE:* The Start of Command character is NOT part of the checksum calculation.

The checksum is calculated by summing the decimal values of the ASCII characters that make up the command.

ASCII characters:	O	8	K	C	O	1	2
value of characters:	48	56	75	67	48	49	50

$$48 + 56 + 75 + 67 + 48 + 49 + 50 = 393$$

$$393/256 = 1 \text{ remainder } 137$$

$$137 \text{ decimal} = 89 \text{ Hex}$$

The complete command then becomes ">08KC01289cr." Alternate forms of this message include ">08KC01289" or ">08KC012??cr."

See Appendix B for tips on implementing a driver.

## Carrying Out Message Transactions

The type of message transaction used between the host computer and Optomux is a 2-pass transaction. For proper operation of the Optomux unit, make sure jumper B10 is installed on the brain board.

### 2-Pass Mode

In 2-pass mode, the host computer transmits a command via the serial link (the first pass) and Optomux returns a complete response on the second pass. This is described in the following steps.

1. The host sends a command message to Optomux (such as >86K1004Acr).
2. If the command was executed successfully, Optomux will respond with an A followed by a carriage return, or if data is to be returned, an A followed by the data characters, a checksum, and ending with a carriage return. (In the example above, a typical response may be B2EB9cr.) If the command was not executed for a particular reason, Optomux will respond with an N followed by a 2-digit error code and a carriage return (such as N03cr.) Note that no checksum is returned if the response was an error condition.

### 4-Pass Mode

The 4-pass mode is activated by either removing the B10 jumper before powering up the Optomux or by using the "E" command to instruct Optomux to use either 2-pass or 4-pass protocol. For more efficient and faster communications, it is highly recommended that Optomux be used in the 2-pass mode as shown above. The 4-pass mode is good for troubleshooting a network. The 4-pass mode works as follows:

1. The host sends a command message (such as >FFACDcr).
2. Optomux echoes the message, substituting an A for the, if there was no error (such as AFFCDcr). If an error occurred, Optomux will respond with an N followed by an error code and carriage return (such as N02cr). In either case, the command is not executed.

3. The host must now send Optomux an E character followed by a carriage return to instruct Optomux to execute the command.
4. If the command was executed successfully, Optomux will respond with an A followed by a carriage return, or if data is to be returned, an A followed by the data characters, a checksum, and ending with a carriage return. If the command was not executed for a particular reason, Optomux will respond with an N followed by an error code and a carriage return (such as N03cr).

## Interpreting the Response

Optomux responses can be divided into three types; an error response, an acknowledgment response, and an acknowledgment with data response.

The error response is a message consisting of the letter N followed by a 2-digit error code and a carriage return (example: N06cr). This type of response does not return a checksum. Following is a list of possible error codes and their descriptions.

## Error Codes

<b>Code</b>	<b>Meaning</b>
<b>00</b>	<b>Power-Up Clear Expected — Command Ignored</b> A command other than "A" (Power-Up Clear) was attempted after power-up or power failure. Once the error is received, it is unnecessary to execute Power-Up Clear. The next command will be executed normally.
<b>Important:</b>	If this error message is received, it means that Optomux has gone through its power-up sequence and has reset all characteristics to defaults. It will be necessary to reinitialize the Optomux I/O unit.
<b>01</b>	<b>Undefined Command</b> The command character was not a legal command character.
<b>02</b>	<b>Checksum Error</b> The checksum received by Optomux did not match the sum of the characters in the command.
<b>03</b>	<b>Input Buffer Overrun</b> The received command contained more than 71 characters for analog or 16 characters for digital boards. The command was ignored.
<b>04</b>	<b>Non-printable ASCII Character Received — Command Ignored</b> Only characters from 21 hex to 7F hex are permitted within commands.

- 05            Data Field Error**  
Not enough characters received.
- 06            Communications Link Watchdog Time-out Error**
- 07            Specified Limits Invalid**

## Acknowledge

An acknowledgment response is a message that consists of the ASCII letter A followed by a carriage return (such as `Acr`). This response is typical of commands that instruct Optomux to perform a function that returns no data. This type of response does not return a checksum.

## Acknowledge with Data

The third type of response is the acknowledgment message followed by data, a checksum, and a carriage return (such as `A130110010FFA100D59cr`). The data component of the message can be of variable length and is composed of fields which may be three characters each when using the `READ ANALOG OUTPUTS` command or four characters each when using all other commands that return data. The data is returned with the most significant position's field first in sequence with the least significant field last.

For instance, suppose a command was sent to a digital Optomux at address 23 hex to read the counter values for positions 0, 2, 4, 6, 8, and 10. The host message would look like the following:

```
>23W5555Bcr
```

Notice that the positions field sent to Optomux is only three characters (555) and the checksum is always two characters (5B). If the response is as follows:

```
A123405671111????ABCD000127cr
```

The return message can be interpreted as follows:

```
Checksum=27
```

**Table 3-1: Meaning of Data in the Response**

Position	Hex Data	Decimal Value
0	0001	1
2	ABCD	43,981
4	????	position 4 is an output
6	1111	4,369
8	0567	1,383
10	1234	4,660

With commands that look at status bits of positions, the data component of the message will be a 4-character data field representing a 16-bit integer value with a 1-to-1 correspondence between bit position and module position (bit 5 corresponds to position 5).

# Initializing An Optomux Network

Many of the operating characteristics of Optomux can be selected by the host computer. On power-up or after a RESET command, Optomux initializes itself to a set of default characteristics.

The RESET defaults are as follows:

**Turnaround delay is set to 0**

**Message protocol is set according to jumper B10**

**Watchdog timer is disabled**

**Analog Optomux** - All module positions are configured to function as inputs with gain set to 1.000 and offsets set to zero.

**Digital Optomux** - All module positions are configured to function as inputs.

All counters set to 0.

All time delays set to 0.

All latches cleared, trigger set for OFF-to-ON.

Pulse width measurement set for ON pulses.

The host computer is responsible for sending each Optomux I/O unit in the network all the commands necessary to select the desired characteristics.

This initialization usually involves sending a POWER-UP CLEAR and a CONFIGURE POSITIONS command to each Optomux. The characteristics for watchdog, timer resolution, and temperature probe type (if used) should also be set at this time. If using the Offset and Gain capability on the analog Optomux, gain coefficients and offsets should be set during initialization for each input position.

*NOTE:* Optomux initialization should be always be performed when Optomux goes through a reset condition. The reset condition can occur whenever a RESET command is sent OR when power is lost and returns. Optomux will respond with a N00cr error whenever power was cycled. The purpose of this error is to warn you that Optomux has lost its configuration and needs to be initialized. A power-up condition can also be caused by a momentary dip on the +5 VDC line, which causes Optomux's processor to reset.

# Serial Port Configuration

Before the host computer can communicate with Optomux, the port characteristics must be set to match those of Optomux. Optomux requires a 10-bit data word with the following format:

## 1 Start Bit 8 Data Bits (no parity) 1 Stop Bit

Example: ASCII "3" (Hex 33)

	<b>start bit</b>									<b>stop bit</b>
	0	1	1	0	0	1	1	0	0	1
	LSB				MSB (always 0)					

Optomux can communicate at eight different baud rates; 300, 600, 1,200, 2,400, 4,800, 9,600, 19,200, and 38,400. The baud rate is selected by jumpers B8, B9, and B11 on the Optomux board. The baud rate at the host computer serial port must match the baud rate that has been selected on the Optomux board.

The MSB is used by some equipment as a parity bit, however Optomux ignores this bit in Host-to-Optomux transmissions.

# Interpreting Analog Data

This section applies only to analog brain board, B2.

## Analog Outputs

All analog output values exchanged between the host computer and Optomux are represented by three ASCII hex digits. Analog output modules are scaled such that when the host computer instructs Optomux to write a value of zero scale (000 hex) to the module, the module will go to its most negative output. When the host computer instructs Optomux to write a value of full-scale to the module (FFF hex), the module will go to its most positive output.

zero scale	000 Hex = 0 decimal
full-scale	FFF Hex = 4095 decimal

When the host computer instructs Optomux to return the current value of an analog output position, Optomux will return three hex digits representing the value. No offset is added to the value and it is returned in the same form that it was sent to Optomux. Therefore, there is no need to store the current values of analog outputs in your host applications program because they can be retrieved from Optomux at any time.

## Example

If we wish to output 2.32 volts from a DA4 (0 to 5 volt output) we must do the following:

```

divide desired value by 5-volts
2.32-volts / 5-volts = .464
multiply result by 4,095 (full-scale in decimal)
1,900 = .464 * 4,095
convert to Hex
1,900 decimal = 76C Hex

```

Therefore, we would instruct Optomux to write the value 76C hex to the output module.

*NOTE:* Refer to the Optomux Family Data Book, form #524, for specific information on each module. The data book contains the necessary scaling and linearization formulas for each module.

## Analog Inputs

When the host computer instructs Optomux to return the value of an analog input module, Optomux returns four ASCII-hex digits representing the value. Analog input modules are scaled such that when the module is receiving its zero scale input, Optomux will return 1,000 hex and when the module is receiving its full-scale input, Optomux will return 1FFF hex. This does NOT apply to the READ TEMPERATURE INPUTS command. Please refer to the READ TEMPERATURE INPUTS command in Chapter 12 for information on handling direct temperature data.

```

zero scale 1,000 Hex = 4,096 decimal
full-scale 1FFF Hex = 8,191 decimal

```

The first of the four hex digits is normally 1, which represents a 1,000 hex (4,096 decimal) offset that Optomux adds to the value of the input (zero scale is 4,096 not 0). If we subtract this offset (1,000 hex) from the value returned by Optomux, we end up with the following zero and full-scale readings:

### Zero Scale

```
(1,000 Hex - 1,000 Hex) = 0 Hex = 0 decimal
```

### Full-Scale

```
(1FFF Hex - 1,000 Hex) = FFF Hex = 4,095 decimal
```

By subtracting 1,000 hex (4,096 decimal) from the value returned by Optomux, zero scale works out to be 0 decimal and full-scale turns out to be 4,095 decimal. If an input module is receiving an input slightly less than zero scale, Optomux will return a value less than 1,000 hex so that if we convert the value to decimal and subtract 4,096; we will end up with a negative number indicating that the input is less than zero scale. If the module is receiving an input more negative than 2.5 percent below zero scale, Optomux will return a value of 0000 hex so subtraction of the offset will yield -4,096. This 2.5 percent (approx.) under range limit is determined by the Optomux hardware and cannot be altered. Optomux may return values which are up to 100 percent over-range with some modules.



Therefore, by always subtracting 4,096 from analog input values, we have the following:

no module installed	value is -4,096
input is more negative than 2.5% below zero-scale	value is -4,096
module is below zero-scale	value is less than 0
module is at zero-scale	value is 0
module is at full-scale	value is 4,095
module is above full-scale	value is greater than 4,095

*NOTE:* In the following examples dealing with analog inputs, it is assumed that the value returned by Optomux has been converted to decimal and the 4,096 offset has been subtracted. This value will be referred to as the DECIMAL VALUE.

### Example #1

If we read the value of a 0 to 5 volt input module (AD6) that is receiving -.122-volt, Optomux will return the value 0F9C. To interpret the return data, do the following:

```
convert the value to decimal
0F9C Hex = 3,996 decimal
subtract 4,096 offset
3,996-4,096 = -100
```

The value -100 indicates that the input is under scale. To determine the value in VOLTS, we do the following:

$$5 \text{ VOLTS} / 4,096 * (-100) = -0.122 \text{ VOLTS}$$

## Example #2

If we read the value of a 0 to 5 volt input module (AD6) that is receiving 3 volts, Optomux will return the value 199A. To interpret the return data do the following:

```
convert the value to decimal
199A Hex = 6,554 decimal
subtract 4,096 offset
6,554-4,096 = 2,458
```

To convert 2,458 to VOLTS, we do the following:

```
5 VOLTS/4,096 * 2,458 = 3 VOLTS
```

## Offset and Gain

### Offset

Optomux can be instructed to set offsets for input module positions. Each time Optomux is instructed to return a value for a particular input position, the offset for that position is subtracted from the value of the input and the result is returned to the host. This is an easy method of correcting minor variations. For example, if we are using a 0 to 5 volt input module (AD6), and the lowest possible input to the module is 0.01 volts, it may be desirable to consider this 0.01 volts as zero scale in your host program. This can be accomplished by instructing Optomux to calculate and set the offset for the module while it is receiving 0.01 volts. After this has been done, Optomux will return zero scale (1,000 hex) when the module is receiving 0.01 volts.

### Gain

Optomux can be instructed to set gain coefficients for input module positions. Each time Optomux is instructed to return a value for a particular input position, the gain coefficient for that position is multiplied by the value of the input and the result is returned to the host. For example, if we are using a 0 to 5 volt input module (AD6), and the highest possible input to the module is 4.5 volts, it may be desirable to consider this 4.5 volts as full-scale in your host program. This can be accomplished by instructing Optomux to calculate and set the gain for the module when it is at 4.5 volts. After this has been done, Optomux will return full-scale (1FFF hex) when the module is receiving 4.5 volts.

*NOTE:* The Offset and Gain features should only be used to compensate for small deviations in sensor range. Use of these commands for scaling a module with a wide range to a scale with a smaller range form resolution; instead, the values will tend to make large jumps for small increments of input.

# Command Directory

## Optomux Command Summary

This section provides a quick reference and index of all the analog and digital Optomux commands. The remaining chapters cover the commands in detail.

The description of each command in the following chapters is formatted as follows:

### Command Name/Command Format

#### PURPOSE

Tells briefly what the command does.

#### VERSIONS

Indicates which Optomux units support the command.

#### FORMAT

Shows the correct format for the middle part of the command message. Brackets “[ ]” are used for clarity to separate fields in the manual. Do not include these brackets in the actual message.

#### REMARKS

Describes in detail how the command is constructed and used.

#### EXAMPLE

Shows actual command messages and responses from Optomux that demonstrate the use of the command.

The following summary lists all the commands. The version indicates whether the command is an A(nalog) or D(igital) command. The digital commands are used with the B1 and the Optomux digital brick, and the analog commands are used with the B2 as well as the analog Optomux brick. The B3000 SNAP processor recognizes all commands in its Optomux mode.

**Table 4-1: Chapter 5: Setup Commands**

Command Name	Command Format	Version
POWER UP CLEAR	A	A,D
RESET	B	A,D
SET TURN-AROUND DELAY	C [data]	A,D
SET WATCHDOG DELAY (Digital)	D [data]	D
SET WATCHDOG DELAY (Analog)	D [positions][data]	A
SET PROTOCOL	E [data]	A,D
IDENTIFY Optomux TYPE	F	A,D
ENHANCED DIGITAL WATCHDOG	<sup>m</sup> [positions][data]	D
ENHANCED ANALOG WATCHDOG	<sup>m</sup> [positions][data]	A
SET TIMER RESOLUTION	n [data]	D
SET TEMPERATURE PROBE TYPE	k [positions][data]	A

**Table 4-2: Chapter 6: I/O Configuration Commands**

Command Name	Command Format	Version
CONFIGURE POSITIONS	<sup>G</sup> [positions]	A,D
CONFIGURE AS INPUTS	H [positions]	A,D
CONFIGURE AS OUTPUTS	I [positions]	A,D
READ MODULE CONFIGURATION	j	A,D

**Table 4-3: Chapter 7: Digital Read/Write Commands**

Command Name	Command Format	Version
WRITE OUTPUTS	J [positions]	D
ACTIVATE OUTPUTS	K [positions]	D
DEACTIVATE OUTPUTS	L [positions]	D
READ STATUS	M	D

**Table 4-4: Chapter 8: Digital Latch Commands**

Command Name	Command Format	Version
SET LATCH EDGES	N [positions]	D
SET LATCH OFF TO ON	O [positions]	D
SET LATCH ON TO OFF	P [positions]	D
READ LATCHES	Q	D
READ AND CLEAR LATCHES	R [positions]	D
CLEAR LATCHES	S [positions]	D

**Table 4-5: Chapter 9: Digital Counting Commands**

Command Name	Command Format	Version
START/STOP COUNTERS	T [positions]	D
START COUNTERS	U [positions]	D
STOP COUNTERS	V [positions]	D
READ COUNTERS	W [positions]	D
READ AND CLEAR COUNTERS	X [positions]	D
CLEAR COUNTERS	Y [positions]	D

**Table 4-6: Chapter 10: Digital Time Delay/Pulse Commands**

Command Name	Command Format	Version
SET TIME DELAY	Z [positions][type][delay]	D
INITIATE SQUARE WAVE	Z [positions] L [data]	D
HIGH RESOLUTION SQUARE WAVE	Z [positions] M [data]	D
RETRIGGER TIME DELAY	h [positions]	D
GENERATE N PULSES	[# of pulses]	D
START ON PULSE	k [positions][data]	D
START OFF PULSE	l [positions][data]	D

**Table 4-7: Chapter 11: Digital Pulse Duration Measurement Commands**

Command Name	Command Format	Version
SET TRIGGER POLARITY	a [positions]	D
TRIGGER ON POSITIVE	b [positions]	D
TRIGGER ON NEGATIVE	c [positions]	D
READ PULSE COMPLETE BITS	d	D
READ DURATION COUNTERS	e [positions]	D
READ AND CLEAR DURATION COUNTERS	f [positions]	D
CLEAR DURATION COUNTERS	g [positions]	D

**Table 4-8: Chapter 12: Analog Read/Write Commands**

Command Name	Command Format	Version
WRITE ANALOG OUTPUTS	J [positions][data]	A
READ ANALOG OUTPUTS	K [positions]	A
UPDATE ANALOG OUTPUTS	S [positions][data]	A
READ ANALOG INPUTS	L [positions]	A
READ AND AVERAGE INPUT	M [positions][data]	A
START INPUT AVERAGING	T [positions][data]	A
READ AVERAGE COMPLETE BITS	i	A
READ INPUT AVERAGE DATA	U [positions]	A
READ TEMPERATURE INPUTS	l [positions]	A
READ AVERAGE TEMPERATURE INPUTS	o [positions]	A

**Table 4-9: Chapter 13: Analog Input Range Commands**

Command Name	Command Format	Version
SET INPUT RANGE	N [positions][data]	A
READ OUT-OF-RANGE LATCHES	O	A
READ AND CLEAR RANGE LATCHES	P [positions]	A
CLEAR OUT-OF-RANGE LATCHES	Q [positions]	A
READ LOWEST VALUES	a [positions]	A
CLEAR LOWEST VALUES	b [positions]	A
READ AND CLEAR LOWEST VALUES	c [positions]	A
READ PEAK VALUES	d [positions]	A
CLEAR PEAK VALUES	e [positions]	A
READ AND CLEAR PEAK VALUES	f [positions]	A

**Table 4-10: Chapter 14: Analog Gain/Offset Commands**

Command Name	Command Format	Version
CALCULATE OFFSETS	g [positions]	A
SET OFFSETS	W [positions][data]	A
CALCULATE AND SET OFFSETS	h [positions]	A
CALCULATE GAIN COEFFICIENTS	X [positions]	A
SET GAIN COEFFICIENTS	Y [positions][data]	A
CALCULATE AND SET GAIN	Z [positions]	A

**Table 4-11: Chapter 15: Analog Waveform Commands**

Command Name	Command Format	Version
SET OUTPUT WAVEFORM	R [positions][type][data]	A
IMPROVED OUTPUT WAVEFORMS	V [positions][type][data]	A

**Table 4-12: Chapter 16: Revision Identification Commands**

Command Name	Command Format	Version
CHECKSUM ON ROM	_	D
DATE OF FIRMWARE	'	D

# Operational Differences – Old Optomux Brain Boards

This protocol manual applies to Optomux units labeled “REV 1” and above. If the Optomux brain board (applies to B1 and B2 only) does not have a revision label and is not surface mount, you have a very old Optomux unit and should be aware of the following differences. Brain board model B3000 is newer than “REV 1”. The brains with surface mount technology are newer than the “REV 1.”

## Digital Brain Boards (B1)

- The “REV 1” (or greater) Optomux units will notify the host that a serial watchdog has timed out by responding to the first command after a serial watchdog time-out with an N06cr error code. The command will not be executed. This is similar to a power-up operation. This error code will not be sent in response to a “POWER UP CLEAR” command.
- Pulse duration counters are edge triggered with the newer Optomux brain boards. The previous versions were level triggered.
- Maximum counter frequency is now 400 Hz, 50 percent duty cycle (minimum ON or OFF pulse time is 1.25 msec). Use of the “GENERATE n PULSES” command will degrade counter frequency to 350 Hz maximum.
- The baud rate may now be changed without cycling power to the Optomux. Additional baud rates of 600, 2,400, 4,800, and 38,400 are now supported when using “REV 1” brain boards or above.
- Two question mark characters (??) may be used instead of the calculated checksum characters in command strings.
- The B1 has 2 commands that can be used to identify which revision of firmware the brain carries. The first is the Checksum command which calculates a checksum on the brain’s ROM and reports it. The second command is the Date command. It will report the date of the firmware revision release.

## Analog Brain Boards (B2)

- The “REV 1” (or greater) Optomux units will notify the host that a serial watchdog has timed out by responding to the first command after a serial watchdog time-out with an N06cr error code. The command will not be executed. This is similar to a power-up operation. This error code will not be sent in response to a “POWER UP CLEAR” command.
- With “REV 1” (or greater) brain boards, averaging is performed using a continuous running average with a sample rate of 100 milliseconds. The number of samples to average is set with the “START INPUT AVERAGING” command as before, however, there is no need to restart the averaging after the number of samples has been reached. The “READ AVERAGE COMPLETE BITS” command indicates that the number of samples to average has been reached. The running average calculation used by Optomux is as follows (where n is the number of samples):

$$\text{New Average} = ((n-1) (\text{Old Average}) + (\text{New Reading}))/n$$

- The baud rate may now be changed without cycling power to the Optomux. Additional baud rates of 600, 2,400, 4,800, and 38,400 are now supported when using “REV 1” brain boards or above.
- Two question mark characters (??) may be used instead of the calculated checksum characters in command strings.



# Command Enhancements

The following commands apply only to “REV 1” (or greater) brain boards. If you try a new command and receive a “N01cr” response, then your brain board does not contain these new features.

## Digital

```
SET ENHANCED DIGITAL WATCHDOG
SET TIMER RESOLUTION
READ MODULE CONFIGURATION
HIGH RESOLUTION SQUARE WAVE
RETRIGGER TIME DELAY
GENERATE n PULSES
START ON PULSE
START OFF PULSE
```

## Analog

```
SET ANALOG WATCHDOG USER-DEFINED VALUE
SET ANALOG WATCHDOG
READ MODULE CONFIGURATION
SET TEMPERATURE PROBE TYPE
READ TEMPERATURE INPUTS
```

The following commands apply only to the surface mount version of the brain boards. If you try a new command and receive a “N01cr” response, then your brain does not contain these new features.

## Digital

```
CHECKSUM ON ROM
DATE OF FIRMWARE
```

# Command Usage Recommendations

The “REV 1” (or greater) Optomux units contain a superset of the commands of original Optomux units. Although some of the new commands offer a better method of doing what a previous command would do, the original commands have been retained for compatibility with existing software. For new installations and new software designs, the following recommendations should be followed.

1. Use the 2-pass protocol for more efficient and faster communications. Therefore, install the B10 jumper and don't use the SET PROTOCOL command.
2. SET TIMER RESOLUTION (command n) is very useful for altering the period of the digital HIGH RESOLUTION SQUARE WAVE command (especially for flashing lights at fast rates).

*CAUTION:* Setting the timer resolution affects all positions on the Optomux unit which use the following commands:

- SET TIME DELAY
  - INITIATE SQUARE WAVE
  - HIGH RESOLUTION SQUARE WAVE
  - RETRIGGER TIME DELAY
  - GENERATE n PULSES
  - START ON PULSE
  - START OFF PULSE
  - READ PULSE COMPLETE BITS
  - READ PULSE DURATION COUNTERS
  - READ AND CLEAR DURATION COUNTERS
3. Don't use the READ AND AVERAGE INPUT (command M). Instead, use the START INPUT AVERAGE (command T), READ AVERAGE COMPLETE BITS (command i), and the READ INPUT AVERAGE DATA (command U). The older READ AND AVERAGE INPUT will wait until the average is complete before returning a response. The communications line is therefore tied up until the average is complete (possibly a long time when using a large number of samples).
  4. Use IMPROVED OUTPUT WAVEFORM (command V) instead of SET OUTPUT WAVEFORM (command R). The IMPROVED OUTPUT WAVEFORM command offers greater flexibility in setting the period of the waveform.

# Setup Commands

---

**POWER-UP CLEAR****COMMAND A**

---

**PURPOSE**

Prevents Optomux from returning a Power-up Clear Expected error message in response to the first instruction following application of power.

**VERSIONS**

Digital, Analog

**FORMAT**

A

**REMARKS**

Only functions if it is the first command sent after power-up; a power-up clear expected error is returned if any other command is sent first, and that command is NOT executed. After a power-up clear expected error is returned, this command does not need to be sent; the next command will be executed normally.

This command has NO effect on Optomux operation or setup — the Power-up Clear Expected error provides an indication to the host that there has been a power failure and that Optomux has been reset to power-up configuration (all positions configured as inputs etc.).

**EXAMPLE**

```
>79AB1cr
```

Perform Power-up Clear at Optomux address 79

**RESET****COMMAND B****PURPOSE**

Resets Optomux to power-up conditions.

**VERSIONS**

Digital, Analog

**FORMAT**

B

**REMARKS**

This command sets all of the operating characteristics of the addressed Optomux to power-up conditions.

**Table 5-1: Power-Up Conditions for Digital and Analog**

<b>DIGITAL</b>	<b>ANALOG</b>
All outputs turned off	0 scale written to all output modules
All positions then configured as inputs	All positions then configured as inputs
Protocol as set by jumper B10	Protocol as set by jumper B10
Watchdog timer disabled	Watchdog timer disabled
Turnaround delay = 0	Turnaround Delay = 0
Counters/duration timers cancelled	All offsets set to 0
Latches cleared	All gain coefficients set to 1
Timer resolution = 10 ms.	All averaging cancelled
	All temperature probe types cancelled

**EXAMPLE**

```
>22BA6cr
```

This example resets Optomux at address 22 to power-up conditions

---

**SET TURNAROUND DELAY****COMMAND C**

---

**PURPOSE**

Tells Optomux to wait for a specified time before responding to commands sent from host.

**VERSIONS**

Digital, Analog

**FORMAT**

C[data]

**REMARKS**

[data] contains a single ASCII digit 0, 1, 2, or 3. Valid delays are:

0	No Delay
1	10 ms
2	100 ms
3	500 ms

If no delay is specified, delay=0 is assumed. On power-up, delay=0.

**EXAMPLE**

```
>92C2E0cr
```

Set Optomux 92 turnaround delay to 100 ms

**SET DIGITAL WATCHDOG****COMMAND D****PURPOSE**

Instructs digital Optomux to monitor activity on the serial communications link and to take a predetermined action if there is no activity within a specified time.

**VERSIONS**

Digital

**FORMAT**

D[data]

**REMARKS**

[data] contains a single ASCII digit 0 through 7 which specifies the time interval and desired action. Valid time and actions are:

**Table 5-2: Values for Watchdog and Their Resulting Action**

[Data]	Time	Action
0		Watchdog disabled
1	10 seconds	Turn all outputs OFF
2	1 minute	Turn all outputs OFF
3	10 minutes	Turn all outputs OFF
4		Watchdog disabled
5	10 seconds	Turn output 0 ON all other outputs OFF
6	1 minute	Turn output 0 ON all other outputs OFF
7	10 minutes	Turn output 0 ON all other outputs OFF

If no data is specified, 0 (watchdog disabled) is assumed. Watchdog is disabled on power-up. The Optomux unit will respond to the first command after a serial watchdog time-out with an error message N06cr, and the command will NOT be executed. This error code is sent as a warning to let the host know a watchdog time-out occurred.

**EXAMPLE**

```
>92D2E1cr
```

Instruct Optomux 92 (hex) to deactivate all outputs if there is no serial line activity for one minute.

## SET ANALOG WATCHDOG

## COMMAND D

**PURPOSE**

Instructs analog Optomux to monitor activity on the serial communications link and to take a predetermined action if there is no activity within a specified time.

**VERSIONS**

Analog

**FORMAT**

D[positions] [data]

**REMARKS**

[positions] contains four ASCII-hex digits indicating the module positions to be affected by the command.

[data] contains 0 to 4 ASCII-hex digits. If the [data] parameter does not contain any data, 0 is assumed.

For [data] values 0 - 7, the times scales and output values are fixed and will override the settings of command m (SET ANALOG WATCHDOG USER-DEFINED VALUE).

[data] values of 8 - 19 will generate a limit error, and the command will not be executed.

For [data] values of 20 - 65,535 (200msec to 10.923 minutes), OPTOMUX will write a user-defined value to the specified positions in the event of a watchdog time-out. The user must use command m (SET ANALOG WATCHDOG USER-DEFINED VALUE) to set the analog output values.

**Table 5-3: Data Values and their Resulting Action**

[Data]	Time	Action
0	Watchdog timer disabled	
1	10 seconds	Write zero-scale
2	1 minute	Write zero-scale
3	10 minutes	Write zero-scale
4	Watchdog timer disabled	
5	10 seconds	Write full-scale
6	1 minute	Write full-scale
7	10 minutes	Write full-scale

*NOTE:* The OPTOMUX unit will respond to the first command after a serial watchdog time-out with an error message N06cr, and the command will NOT be executed. The error message is sent as a warning to let the host know a watchdog time-out occurred.

**EXAMPLE**

```
>1AD000F2BEcr
```

Instructs Optomux at address 1A to output zero-scale at positions 0, 1, 2, and 3 if there is no activity on the serial line for one minute



---

**SET PROTOCOL****COMMAND E**

---

**PURPOSE**

Instructs Optomux to use either 2-pass or 4-pass protocol.

**VERSIONS**

Digital, Analog

**FORMAT**

E[data]

**REMARKS**

[data] contains a single ASCII digit 0 or 1 which selects protocol:

0	Two-Pass Protocol
1	Four-Pass Protocol

If no protocol is specified, protocol=0 is assumed. On power-up, protocol is set according to Jumper B10.

**EXAMPLE**

```
>9BE1F1cr
```

Instruct Optomux 9B to use 4-pass protocol

---

**IDENTIFY OPTOMUX TYPE**

---

**COMMAND F****PURPOSE**

Instructs Optomux to identify itself as either a digital or analog I/O unit.

**VERSIONS**

Digital, Analog

**FORMAT**

F

**REMARKS**

Optomux will respond with a return message containing a 2-digit identification code as follows:

00 = Digital Optomux  
01 = Analog Optomux

**EXAMPLE**

```
>D0FBaCr
```

Command message sent from host requesting Optomux at address D0 to identify itself

```
A0161Cr
```

Message returned by Optomux identifying itself as an analog Optomux I/O unit

---

**SET ENHANCED DIGITAL WATCHDOG****COMMAND m**

---

**PURPOSE**

Instructs digital Optomux to monitor activity on the serial communications link and to take a specified action if there is no activity within a specified time.

**VERSIONS**

Digital

**FORMAT**

m[positions] [data]

**REMARKS**

After this command is issued, if a character is not received within the time specified by [data], Optomux will turn on all outputs corresponding to a 1 and turn off all outputs corresponding to a 0 in the bitmask specified by [positions]. All time delay outputs are cancelled and inputs are not affected.

The delay time may be 0 to 4 digits and is set to [data] \* 10 milliseconds. Delays of less than 200 milliseconds (except 0), will result in a limit error and the command is not executed. A delay time of zero disables the digital watchdog function. If no delay time is sent, 0 is assumed.

The Optomux unit will respond to the first command after a serial watchdog time-out with a N06cr error code message and the command will NOT be executed. This error code is sent as a warning to let the host know a watchdog time-out occurred.

**EXAMPLE**

```
>ECm0A841F47Dcr
```

This command message instructs Optomux at address EC to turn on output modules 11, 9, 7, and 2 and turn off all other output modules if there is no activity on the serial line for 5 seconds. This example assumes that the positions indicated were previously configured as outputs.

---

**SET ANALOG WATCHDOG USER-DEFINED VALUE**

---

**COMMAND m****PURPOSE**

Used in conjunction with command D, Set Analog Watchdog, this command sets the data that an analog Optomux will write to the specified output modules upon a serial watchdog time-out.

**VERSIONS**

Analog

**FORMAT**

m[positions] [data]

**REMARKS**

[positions] contains four ASCII-hex digits indicating the module positions to be affected by the command.

[data] is set only for output modules that correspond to a 1 in the [positions] bitmask. Three characters of data are expected for each output module specified. If enough data is not received (the number of data values must equal the number of bits set to 1 in the bitmask), Optomux will respond with a N05cr data field error message.

The delay time is set by command D, SET ANALOG WATCHDOG.

The Optomux unit will respond to the first command after a serial watchdog time-out with an error message N06cr, and the command will not be executed. This error code is sent as a warning to let the host know a watchdog time-out occurred.

**EXAMPLE**

```
>AAm0081A20555FAcr
```

This message instructs Optomux at address AA to output the value A20 at position 7 and a value of 555 to output module at position 0 when a serial watchdog time-out occurs.

---

**SET TIMER RESOLUTION****COMMAND n**

---

**PURPOSE**

This command sets a global timer value for all timing functions on the Optomux digital brain.

**VERSIONS**

Digital

**FORMAT**

n[data]

**REMARKS**

[data] contains two ASCII hex digits indicating the timer resolution. The timing resolution is equal to [data] \* 10 milliseconds. If the value in [data] is 0, the timer resolution will be 2.56 seconds.

*NOTE:* This command is a global command and affects the time base for the entire Optomux brain board. This command affects the timing resolution for the following commands:

SET TIME DELAY  
INITIATE SQUARE WAVE  
HIGH RESOLUTION SQUARE WAVE  
RETRIGGER TIME DELAY  
GENERATE N PULSES  
START ON PULSE  
START OFF PULSE  
READ PULSE COMPLETE BITS  
READ PULSE DURATION COUNTERS  
READ AND CLEAR DURATION COUNTERS

**EXAMPLE**

```
>FE0A6Acr
```

This example sets the timer resolution of the digital Optomux at address FE to 100 ms.

---

**SET TEMPERATURE PROBE TYPE****COMMAND k**

---

**PURPOSE**

This command sets the probe type for positions using temperature input modules (thermocouples, ICTDs, and RTDs), so that the READ TEMPERATURE INPUTS command can be used to read the temperature directly.

**VERSIONS**

Analog

**FORMAT**

k[positions] [data]

**REMARKS**

[positions] is a field of 4 ASCII-hex digits indicating which positions will be affected by the command.

[data] is a single ASCII-hex character which sets the temperature probe type for all the modules corresponding to a 1 in the bitmask [positions]. The following table indicates the possible values for [data]:

**Table 5-4: Data Values and their Corresponding Temperature Types**

Data	Positions
0	No Temperature Probe
1	ICTD Probe (AD4 Module)
2	10 Ohm RTD Probe (AD14T Module)
3	100 Ohm RTD Probe (AD10T Module)
4	Type J Thermocouple (AD5/AD5T)
5	Type K Thermocouple (AD8/AD8T)
6	Type R Thermocouple (AD17T)
7	Type S Thermocouple (AD17T)
8	Type T Thermocouple (AD18T)
9	Type E Thermocouple (AD19T)
A	Reserved
B	Reserved
C	Reserved
D	Reserved
E	Reserved
F	Reserved

The "SET TEMPERATURE PROBE TYPE" command can be sent multiple times with different values for [positions] and [data] to set a variety of probes on one rack.

**EXAMPLE**

```
>DCk40607F3cr
```

This example shows a command message sent from the host to Optomux at address DC. The message sets the temperature probe type for modules in positions 14, 6, and 5 to a type S thermocouple.





# I/O Configuration Commands

---

**CONFIGURE POSITIONS**
**COMMAND G**


---

**PURPOSE**

Identifies the function of positions as either inputs or outputs.

**VERSIONS**

Digital, Analog

**FORMAT**

G[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit specifies the configuration of four module positions. Positions corresponding to 1 bits are configured as outputs. Positions corresponding to 0 bits are configured as inputs. If the [positions] field is not given, a value of FFFF is assumed by Optomux.

```
0 bit in positions field = Input
1 bit in positions field = Outputs
```

Positions not specified are left unchanged. If the configuration for any relay position is changed, any time-delay, latch, etc. is cleared.

On power-up, all positions are configured as inputs.

**EXAMPLE**

```
>00G11336Fcr
```

Command message sent from host instructing Optomux at address 00 to configure positions 0, 1, 4, 5, 8, and 12 to function as outputs. All other positions will be configured to function as inputs.

```
>45G2E2cr
```

This command message instructs the Optomux I/O unit with address 45 to configure positions 0, 2, and 3 to function as inputs and position 1 to function as an output. All other positions are unchanged.

---

**CONFIGURE AS INPUTS****COMMAND H**

---

**PURPOSE**

Configures module positions to function as inputs.

**VERSIONS**

Digital, Analog

**FORMAT**

H[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the function of four module positions. Module positions corresponding to 1 bits in the position field are configured to function as inputs. All other module positions are left unchanged. If [positions] is not included, Optomux assumes a value of FFFF.

If this command changes the configuration of any position, any previous waveform, latch, etc., is cleared.

**EXAMPLE**

```
>4BH5F3cr
```

Configure positions 0 and 2 on Optomux 4 as inputs; all other positions are left unchanged.

---

**CONFIGURE AS OUTPUTS****COMMAND I**

---

**PURPOSE**

Configures module positions to function as outputs.

**VERSIONS**

Digital, Analog

**FORMAT**

I[positions]

*NOTE:* This command is an uppercase i, NOT a lowercase L.

**REMARKS:**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the function of four module positions. Module positions corresponding to 1 bits in the position field are configured to function as outputs. All other module positions are left unchanged. If [positions] is not given, a value of FFFF is assumed by Optomux.

If the configuration of a point is changed by this command, any time delay, latch, etc. is cleared.

**EXAMPLE:**

```
>4BI5F4cr
```

Configure positions 0 and 2 on Optomux 4B to function as outputs.

---

**READ MODULE CONFIGURATION****COMMAND j**

---

**PURPOSE**

This command returns the current input/output configuration

**VERSIONS**

Digital, Analog

**FORMAT**

j

**REMARKS**

Returns the current configuration for all 16 I/O module positions. Optomux responds by returning a message that contains four ASCII-hex digits that specify which positions are configured as inputs and which are configured as outputs. Each digit corresponds to four module positions in the same manner as the digits contained in a positions field. Module positions corresponding to 1 bits are configured as outputs, and positions corresponding to 0 bits are configured as inputs.

**EXAMPLE**

```
>FFj??cr
```

Command sent from host to Optomux at address FF hex (255 decimal), requesting the current configuration of all the positions. Notice how a ?? is used for the checksum. The ?? is a wildcard character for use when debugging.

```
A0000C0cr
```

Response from Optomux indicating that all positions are configured as inputs.

# Digital Read/Write Commands

---

**WRITE OUTPUTS****COMMAND J**

---

**PURPOSE**

Turns output modules ON and OFF.

**VERSIONS**

Digital

**FORMAT**

J[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the ON/OFF status of four module positions. Module positions corresponding to 1 bits in the position field are turned ON. Module positions corresponding to 0 bits are turned OFF. Time Delays, if set, are implemented upon execution of this command. Module positions that have been configured to function as inputs are not affected by this command. If the [positions] field is left out, Optomux will assume a positions bitmask of FFFF.

**EXAMPLE**

```
>00JFFFFC2cr
```

Turn ON all outputs at Optomux address 00

```
>45J0E3cr
```

Turn OFF modules 0, 1, 2, 3 at Optomux address 45; module positions 4-15 are not affected.

---

**ACTIVATE OUTPUTS****COMMAND K**

---

**PURPOSE**

Turns ON output modules.

**VERSIONS**

Digital

**FORMAT**

K[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. Module positions corresponding to 1 bits in the position field are turned ON. All other module positions are left unchanged. Time delays, if set, are implemented upon execution of this command. Module positions that have been configured to function as inputs are not affected by this command. If the [positions] field is left out, Optomux will assume a positions bitmask of FFFF.

**EXAMPLE**

```
>99K55CCADcr
```

Instructs Optomux at address 99 to activate outputs at positions 2, 3, 6, 7, 8, 10, 12, and 14. All other positions are not affected.

---

**DEACTIVATE OUTPUTS****COMMAND L**

---

**PURPOSE**

Turns OFF output modules.

**VERSIONS**

Digital

**FORMAT**

L[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. Module positions corresponding to 1 bits in the position field are turned OFF. All other module positions are left unchanged. Time delays, if set, are implemented upon execution of this command. Module positions that have been configured to function as inputs are not affected by this command. If the [positions] field is left out, Optomux will assume a positions bitmask of FFFF.

**EXAMPLE**

```
>BCL607cr
```

This command tells the Optomux at address BC to turn OFF outputs at positions 1 and 2.

```
>9ALFFFFDEcr
```

Instruct Optomux at address 9A to turn OFF all outputs (equivalent to WRITE OUTPUTS command with [positions]="0000").

---

**READ ON/OFF STATUS****COMMAND M**

---

**PURPOSE**

Returns the ON/OFF status of all module positions.

**VERSIONS**

Digital

**FORMAT**

M

**REMARKS**

Returns the current state of all 16 module positions. Optomux responds by returning a message that contains four ASCII-hex digits which specify the ON/OFF status of 16 module positions. Each digit corresponds to four module positions in the same manner as the digits contained in the positions field. Module positions corresponding to 1 bits are in the ON state, and positions corresponding to 0 bits are in the OFF state.

**EXAMPLE**

```
>FFMD9cr
```

Command sent from host to Optomux at address FF, requesting the ON/OFF status of all positions.

```
A0000C0cr
```

Response from Optomux indicating that all positions are in the OFF state.

```
A0AC2E6cr
```

Response from Optomux indicating that positions 1, 6, 7, 9, and 11 are ON, and that all others are OFF.



# Digital Latch Commands

---

**SET LATCH EDGES****COMMAND N**

---

**PURPOSE**

Set positions configured as inputs to latch on either ON-to-OFF or OFF-to-ON transitions.

**VERSIONS**

Digital

**FORMAT**

N[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-Hex digits. If [positions] is not given, Optomux assumes a positions bitmask of FFFF. Each digit in the positions field specifies the status of four module positions. Module positions corresponding to 1 bits in the positions field are set to latch on ON-to-OFF transitions. Module positions corresponding to 0 bits in the positions field are set to latch on OFF-to-ON transitions. All other module positions are left unchanged.

Module positions that have been configured to function as outputs are not affected by this command. On power-up, all inputs are set to latch OFF-to-ON.

**EXAMPLE**

```
>55NFFFECECr
```

This command instructs the Optomux at address 55 to set position 0 to latch on OFF-to-ON transitions, all other positions are set to latch on ON-to-OFF transitions.

---

**SET OFF-TO-ON LATCHES****COMMAND 0**

---

**PURPOSE**

Sets positions to latch on OFF-to-ON transitions

**VERSIONS**

Digital

**FORMAT**

0[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. If [positions] is not given, Optomux assumes a positions bitmask of FFFF. Each digit in the positions field specifies the status of four module positions. Module positions corresponding to 1 bits in the positions field are set to latch on OFF-to-ON transitions. All other module positions are left unchanged. Module positions that have been configured to function as outputs are not affected by this command. On power-up, all inputs are set to latch OFF-to-ON.

**EXAMPLE**

```
>FF0C000AEcr
```

This command configures positions 14 and 15 at Optomux address FF to latch on OFF-to-ON transitions; all other positions are not affected.

---

**SET ON-TO-OFF LATCHES****COMMAND P**

---

**PURPOSE**

Sets positions to latch on ON-to-OFF transitions

**VERSIONS**

Digital

**FORMAT**

P[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. If [positions] is not given, Optomux assumes a positions bitmask of FFFF. Each digit in the positions field specifies the status of four module positions. Module positions corresponding to 1 bits in the positions field are set to latch on ON-to-OFF transitions. All other module positions are left unchanged. Module positions that have been configured to function as outputs are not affected by this command. On power-up, all inputs are set to latch OFF-to-ON.

**EXAMPLE**

```
>FFFC000AFcr
```

This command instructs Optomux at address FF to set positions 14 and 15 to latch on ON-to-OFF transitions; all other positions are not affected.

---

**READ LATCHES****COMMAND Q**

---

**PURPOSE**

Returns data indicating which of the inputs have latched.

**VERSIONS**

Digital

**FORMAT**

Q

**REMARKS**

Returns the current state of all latches. Optomux responds by returning a message that contains four ASCII-hex digits that specify which module positions have latched (the correct edge has been detected at that position). Each digit corresponds to four module positions in the same manner as the digits contained in the positions field. Module positions corresponding to 1 bits have been latched. Bits corresponding to positions configured as outputs should be ignored. This command has no effect on the Optomux latches so subsequent READ LATCHES commands will return consistent results.

**EXAMPLE**

```
>77QBfcR
```

Command sent from host computer to read latches on Optomux at address 77.

```
A888E0cR
```

Message returned from Optomux indicating that inputs at positions 15, 11, 7, and 3 have latched.

---

**READ AND CLEAR LATCHES****COMMAND R**

---

**PURPOSE**

Returns data indicating which of the inputs have latched and then resets the latches for specified input positions to the unlatched state.

**VERSIONS**

Digital

**FORMAT**

R[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. Module positions corresponding to 1 bits will have their latches cleared (set to “unlatched”); all other latches are left unchanged. If the positions field is omitted all latches will be cleared.

Optomux responds by returning a message that contains four ASCII-hex digits that specify which module positions have latched (the correct edge has been detected at that position). Each digit corresponds to four module positions in the same manner as the digits contained in the positions field. Module positions corresponding to 1 bits have been latched. Bits corresponding to positions configured as outputs should be ignored.

**EXAMPLE**

```
>77RB032cr
```

Command message to read all latches on Optomux at address 77, and to clear latches for positions 4, 5, and 7.

```
A762AE0cr
```

Response from Optomux indicating that positions 1, 3, 5, 9, 10, 12, 13, and 14 have latched.

---

**CLEAR LATCHES****COMMAND S**

---

**PURPOSE**

Sets latches for specified input positions to unlatched state.

**VERSIONS**

Digital

**FORMAT**

S[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. If [positions] is not given, Optomux assumes a positions bitmask of FFFF. Each digit in the positions field specifies the status of four module positions. Module positions corresponding to 1 bits will have their latches cleared (set to "unlatched"); all other latches are left unchanged. If the positions field is omitted all latches will be cleared.

**EXAMPLE**

```
>77S1cr
```

This command instructs Optomux at address 77 to clear all latches.

```
>77S1F2cr
```

Instructs Optomux at address 77 to clear the latch for position 0.

# Digital Counting Commands

---

**START/STOP COUNTERS****COMMAND T**

---

**PURPOSE**

Starts and stops counting of OFF-to-ON transitions at specified input positions.

**VERSIONS**

Digital

**FORMAT**

T[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. If [positions] is not given, Optomux will assume a positions bitmask of FFFF. Each digit in the positions field specifies the status of four module positions. Optomux will start counting OFF-to-ON transitions at module positions corresponding to 1 bits in the positions field. Optomux will stop counting OFF-to-ON transitions at module positions corresponding to 0 bits in the positions field. All other positions are left unchanged.

This command has no effect on the stored count and counting can be started or resumed at any time.

The maximum count is 65,535; the counter then resets to 0.

Frequencies up to 400 Hertz with a minimum pulse width of 1.25 milliseconds can be counted.

**EXAMPLE**

```
>73T0F34cr
```

This example instructs Optomux at address 73 to start counting at positions 0, 1, 2, and 3, and to stop counting at positions 4, 5, 6, and 7. All other positions are left unchanged.

---

**START COUNTERS****COMMAND U**

---

**PURPOSE**

Starts/resumes counting of OFF-to-ON transitions at specified input positions.

**VERSIONS**

Digital

**FORMAT**

U[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. If [positions] is not given, Optomux will assume a positions bitmask of FFFF. Each digit in the positions field specifies the status of four module positions. Optomux will start counting OFF-to-ON transitions at input positions corresponding to 1 bits in the positions field. All other positions are not affected.

This command has no effect on a preexisting count value, if any.

The maximum count is 65,535; the counter then resets to 0.

Frequencies up to 400 Hertz with a minimum pulse width of 1.25 milliseconds can be counted.

**EXAMPLE**

```
>EEU3042cr
```

This command starts counting at positions 4 and 5 on Optomux address EE.



---

**STOP COUNTERS****COMMAND V**

---

**PURPOSE**

Stops counting of OFF-to-ON transitions at specified input positions.

**VERSIONS**

Digital

**FORMAT**

v[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. If [positions] is not given, Optomux will assume a positions bitmask of FFFF. Each digit in the positions field specifies the status of four module positions. Optomux will stop counting OFF-to-ON transitions at input positions corresponding to 1 bits in the positions field. All other positions are left unchanged.

This command has no effect on the current value of the counters. Counting can be resumed at any time.

**EXAMPLE**

```
>EDV80077cr
```

This command message stops counting at position 11 on Optomux address ED.

---

**READ COUNTERS****COMMAND W**

---

**PURPOSE**

Returns the counter values for the specified input positions.

**VERSIONS**

Digital

**FORMAT**

w[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. Optomux will return the counter values for the input positions corresponding to 1 bits in the positions field. Values are returned as four ASCII-hex digits which represent a 16-bit value (0 to 65,535 decimal). Counter values are returned in sequence from the highest to lowest module position. Attempts to read an output will result in a string of four question marks "???" being returned for that position. This command has no effect on the current value of the counters. If the positions field is omitted from the command message a position field of FFFF is assumed (all counters are read).

**EXAMPLE**

```
>23W6F2cr
```

Command message sent from host to Optomux at address 23 to read counters for module positions 1 and 2.

```
AB00000089Acr
```

Message returned by Optomux. Counter value is B000 for position 2 (45,056 decimal) and 0008 (8 decimal) for position 1.

```
>23W555Bcr
```

Command message sent from host computer to Optomux. Read counters 0, 2, 4, 6, 8, and 10 at Address 23.

```
A123405671111????ABCD000127cr
```

The return message can be interpreted as follows:

Position	Hex Data	Decimal Value
0	0001	1
2	ABCD	43,981
4	????	position 4 is an output
6	1111	4,369
8	0567	1,383
10	1234	4,660

---

**READ AND CLEAR COUNTERS****COMMAND X**

---

**PURPOSE**

Returns the counter values for the specified positions and then sets the counters for those positions to 0.

**VERSIONS**

Digital

**FORMAT**

X[positions]

**REMARKS**

This command performs the READ COUNTERS function and then clears the counters for the positions that were read. [positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. Optomux will return the counter values for the module positions corresponding to 1 bits in the positions field and then clear the counters for those positions. Values are returned as four ASCII-hex digits which represent a 16-bit value (0 to 65,535 decimal). Counter values are returned in sequence from the highest to lowest module position. Attempts to read an output will result in a string of four question marks (????) being returned for that position. If the positions field is omitted from the command message a position field of FFFF is assumed (all counters are read and cleared).

**EXAMPLE**

```
>A8X80069cr
```

This message instructs Optomux at address A8 to return a counter value for position 11, then clear the counter to 0.

```
>A0F00D6cr
```

This response from Optomux shows the counter value of position 11 to be F00 hex or 3,840 decimal.

---

**CLEAR COUNTERS****COMMAND Y**

---

**PURPOSE**

Resets counters for specified input positions.

**VERSIONS**

Digital

**FORMAT**

Y[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. Optomux will clear the counter values for the input positions corresponding to 1 bits in the positions field. All other positions are left unchanged. If the positions field is omitted from the command message a position field of FFFF is assumed (all counters are cleared).

**EXAMPLE**

```
>A8Y8006Acr
```

This command clears the counters for position 11 at Optomux address A8.



# Digital Time Delay/Pulse Commands

---

**SET TIME DELAY**
**COMMAND Z**


---

**PURPOSE**

Sets specified output positions to be used in a pulsed or delayed mode.

**VERSIONS**

Digital

**FORMAT**

Z[positions] [modifier] [data]

**REMARKS**

[positions] four ASCII-hex digits which specify the module positions to be affected. Module positions corresponding to 1 bits in the positions field will be used in the selected delayed or pulsed mode.

[data] contains from 1 to 4 ASCII-Hex digits representing the length of the delay. The actual delay time is equal to

$$\begin{aligned} \text{desired time} &= [\text{data}] * \text{TRS} * 10 \text{ mSec.} \\ &\text{or} \\ [\text{data}] &= \text{desired time} / (\text{TRS} * 10 \text{ mSec.}) \end{aligned}$$

Where TRS is the Timer Resolution Setting as set by the "Set Timer Resolution Command."  
The default for TRS is 1.

*NOTE:* A value of 0 for [data] is equivalent to a value of FFFF (65,535).

[modifier] contains a single ASCII character which specifies the type of delay to be used. Valid delay types are:

**Table 10-1: Values for “Modifier” Parameter**

[modifier]	Description of Delay Type
G	Normal Operation - turn off existing delay (when using this modifier [data] may be omitted from the command message).
H	When instructed to go from OFF to ON, turn ON for desired time, then turn OFF
I	When instructed to go from OFF to ON, stay OFF for desired time, then turn ON
J	When instructed to go from ON to OFF, turn OFF for desired time, then turn ON
K	When instructed to go from ON to OFF, stay ON for desired time, then turn OFF
L	Initiate a continuous square wave with "ON" and "OFF" times computed as follows: "ON" time = 2.56 seconds * TRS * the first two digits in the data field (leftmost characters) "OFF" time = 2.56 x * TRS * the last two digits in the data field
M	Initiate a continuous high resolution square wave with "ON" and "OFF" times computed as follows: "ON" time = TRS * the first two digits in the data field (leftmost characters) "OFF" time = TRS * the last two digits in the data field

The timer resolution is set by the “SET TIMER RESOLUTION” command.

With type L and M modifiers, the square wave will continue to be output until the delay type for the output is changed or turned off. The “WRITE OUTPUTS,” “ACTIVATE OUTPUTS,” and “DEACTIVATE OUTPUTS” commands will have NO EFFECT when using the type L and M modifiers.

### EXAMPLE

All examples assume a TRS value of 1 (10 mSec).

```
>89ZCI66C3cr
```

Set module positions 2 and 3 at Optomux address 89 for delay type I with a pulse length of 1.02 seconds.

```
>24Z1000GC8cr
```

Turn off existing delay at Optomux address 24, module position 12.

```
>10Z1111K3E87Acr
```

At Optomux address 10, set positions 12, 8, 4, and 0 for a type K delay of 10 seconds.

```
>11Z66L015842cr
```

At address 11, set positions 6, 5, 2, and 1 for a square wave — ON 2.56 seconds, OFF 3.75 minutes.

```
>1EZ42M041F5Ecr
```

At address 1E, set positions 1 and 6 for a high resolution square wave; ON for 40 mSec, and OFF for 310 mSec.



---

**RETRIGGER TIME DELAY****COMMAND h**

---

**PURPOSE**

Restarts or triggers an existing time delay.

**VERSIONS**

Digital

**FORMAT**

h[positions]

**REMARKS**

This command overrides an existing time delayed output by setting the time delay counter to the original time delay value, then retriggering the time delay if it was currently active. The original time delay was the value set by the "Set Time Delay" command. This command can be used to extend the time-out period of an output (as in a watchdog circuit).

Since the "SET TIME DELAY" command cancels a time delay that is active, the "RETRIGGER TIME DELAY" command can be used after a "SET TIME DELAY" command to dynamically change the time-out period.

This command does not affect square wave generation. If no [positions] bitmask is sent, an FFFF (all positions) is assumed.

**EXAMPLE**

```
>2Eh2041cr
```

This command instructs Optomux at address 2E to retrigger a time delay on module position 5.

**GENERATE N PULSES****COMMAND i****PURPOSE**

This command instructs Optomux to output a specified number of pulses.

**VERSIONS**

Digital

**FORMAT**

i [positions] [modifier] [data]

**REMARKS**

The bitmask in [positions] must consist of four ASCII-hex digits and represents the output modules to be affected. The quantity in [modifier] must be two digits (00 to FF) and represents one-half of the period for the pulses in increments of the timer resolution setting. All pulses have a 50 percent duty cycle. The timer resolution is set using the "SET TIMER RESOLUTION" command. A time of 0 will cancel any time delay or pulse stream currently being generated by the specified output. Sending this command when a pulse train is already active, will cause the pulse train to be retriggered with the latest values.

The quantity specified in the [data] field can be from 0 to 4 ASCII-hex digits (0 to FFFF) and represents the number of pulses to be output. A 0 in this field will generate 65,536 pulses.

**EXAMPLE**

```
>CCi0040320064E2cr
```

This example instructs Optomux at address CC to output 100 pulses at position 6. Assuming the Timer Resolution Setting (TRS) is 1 (10 mSec.), the pulses will have a period of 1 second (500 mSec. OFF, 500 mSec. ON).

**START ON PULSE****COMMAND k****PURPOSE**

Turns on all specified outputs for a specific length of time.

**VERSIONS**

Digital

**FORMAT**

k[positions] [data]

**REMARKS**

[positions] must contain four ASCII-hex digits representing the bitmask for the outputs to be pulsed. Positions corresponding to 1 bits will activate a pulse for the duration specified by the value in the [data] field. The quantity in the [data] field represents the period of the "ON" pulse in increments of the timer resolution. The timer resolution is set using the "SET TIMER RESOLUTION" command. [data] may be from 0 to 4 ASCII-hex characters. If a zero is specified, the command does nothing. If no [data] value is given, a zero is assumed. This function is retriggerable, therefore, it can be used as a watchdog circuit by continuously sending this command at a rate faster than the pulse length. To cancel the pulse, set [data] to a 1, and the pulse will turn off and the outputs will be deactivated within one timer increment.

**EXAMPLE**

```
>BBk000514??cr
```

This command configures positions 2 and 0 of the Optomux at address BB to generate an "ON" pulse for a duration of 20 times the timer resolution setting (TRS). All other positions are not affected.

---

**START OFF PULSE****COMMAND I**

---

**PURPOSE**

Turns off all specified outputs for a specific length of time.

**VERSIONS**

Digital

**FORMAT**

`l[positions] [data]`

*NOTE:* This command letter is a lowercase L, NOT an uppercase i.

**REMARKS**

[positions] must contain four ASCII-hex digits representing the bitmask for the outputs to be pulsed. Positions corresponding to 1 bits will deactivate a pulse for the duration specified by the value in the [data] field. The quantity in the [data] field represents the period of the "OFF" pulse in increments of the timer resolution. The timer resolution is set using the "SET TIMER RESOLUTION" command. [data] may be from 0 to 4 ASCII-hex characters. If a zero is specified, the command does nothing. If no [data] value is given, a zero is assumed. This function is retriggerable, therefore, it can be used as a watchdog circuit by continuously sending this command at a rate faster than the pulse length. To cancel the pulse, set [data] to a 1, and the pulse will turn off and the outputs will be activated within one timer increment.

**EXAMPLE**

```
>441000106??cr
```

This command configures position 0 of the Optomux at address 44 to generate an "OFF" pulse for a duration of six times the timer resolution setting (TRS). All other positions are not affected.

# Digital Pulse Duration Measurement Commands

---

**SET PULSE TRIGGER POLARITY**
**COMMAND a**


---

**PURPOSE**

Instructs Optomux to measure ON and OFF pulses.

**VERSIONS**

Digital

**FORMAT**

a[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the desired pulse trigger polarity of four module positions. If [positions] is omitted, Optomux assumes a value of FFFF.

Positions corresponding to 1 bits are set to measure ON duration, while positions corresponding to 0 bits are set to measure OFF duration.

Optomux will measure the duration of the first pulse of the appropriate level and store the result to be recalled by the host. As soon as Optomux has measured a complete pulse for any position, a "pulse complete bit" is set to indicate that an entire pulse has been measured. The host computer can read the "pulse complete bits" to determine whether Optomux has finished the measurement process, and can then read the values of the pulse duration counters, if any.

Maximum pulse duration is "FFFF" (65,535) \* TRS. where TRS is the Timer Resolution Setting set by the "SET TIMER RESOLUTION" command.

This command does not clear preexisting duration counter values or pulse complete bits. If a positions "pulse complete bit" has been previously set, no measurements are made until that positions' "pulse complete bit" and duration counter are cleared by either the CLEAR DURATION COUNTERS command or the READ AND CLEAR DURATION COUNTER command.

**EXAMPLE**

```
>88a506cr
```

Instruct Optomux at address 88 to configure positions 2 and 0 to measure "ON" pulses; positions 1 and 3 to measure "OFF" pulses; and leave all other positions unchanged.

**TRIGGER ON POSITIVE PULSE****COMMAND b****PURPOSE**

Sets the specified positions to measure the duration of positive (ON) pulses.

**VERSIONS**

Digital

**FORMAT**

b[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. If [positions] is omitted, Optomux assumes a value of FFFF. Positions corresponding to 1 bits are set to measure ON duration. All other positions are left unchanged.

Optomux will measure the duration of the first positive pulse and store the result to be recalled by the host. As soon as Optomux has measured a complete pulse for a given position, a "pulse complete bit" is set to indicate that an entire pulse has been measured. The host computer can read the "pulse complete bits" to determine whether Optomux has finished the measurement process and can then read the values of the pulse duration counters, if any.

Maximum pulse duration is "FFFF" (65,535) \* TRS. where TRS is the Timer Resolution Setting set by the "SET TIMER RESOLUTION" command.

This command does not clear preexisting duration counter values or pulse complete bits. If a positions "pulse complete bit" has been previously set, no measurements are made until that positions' "pulse complete bit" and duration counter are cleared by either the CLEAR DURATION COUNTERS command or the READ AND CLEAR DURATION COUNTER command.

**EXAMPLE**

```
>BBb51Bcr
```

This command configures positions 2 and 0 to measure "ON" pulses at Optomux address BB. All other positions are not affected.

**TRIGGER ON NEGATIVE PULSE****COMMAND c****PURPOSE**

Sets the specified positions to measure the duration of negative (OFF) pulses.

**VERSIONS**

Digital

**FORMAT**

c[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Each digit in the positions field specifies the status of four module positions. If [positions] is omitted, Optomux assumes a value of FFFF. Positions corresponding to 1 bits are set to measure OFF duration. All other positions are unchanged.

Optomux will measure the duration of the first negative pulse, and store the result to be recalled by the host. As soon as Optomux has measured a complete pulse for a given position, a "pulse complete bit" is set to indicate that an entire pulse has been measured. The host computer can read the "pulse complete bits" to determine whether Optomux has finished the measurement process and can then read the values of the pulse duration counters, if any.

Maximum pulse duration is "FFFF" (65,535) \* TRS. where TRS is the Timer Resolution Setting set by the "SET TIMER RESOLUTION" command.

This command does not clear preexisting duration counter values, or pulse complete bits. If a positions' pulse complete bit has been previously set, no measurements are made until that "pulse complete bit" is cleared by either the CLEAR DURATION COUNTERS or READ AND CLEAR DURATION COUNTERS command.

**EXAMPLE**

```
>BBc51Ccr
```

This command configures positions 2 and 0 to measure "OFF" pulses at Optomux address BB. All other positions are not affected.

---

**READ PULSE COMPLETE BITS****COMMAND d**

---

**PURPOSE**

Allows the host computer to determine which positions have completed the pulse duration measurement period.

**VERSIONS**

Digital

**FORMAT**

d

**REMARKS**

Optomux responds to this command with a return message containing four ASCII-hex digits. Each of these digits represents the status of four positions. Positions corresponding to 1 bits have completed a pulse of the correct level. No further measurements can be made at these positions until the pulse complete bits are reset using either the READ AND CLEAR DURATION COUNTERS or the CLEAR DURATION COUNTERS command.

**EXAMPLE**

Assume that positions 0–7 have been configured to function as outputs.

```
>76dD1cr
```

Message sent from host requesting the pulse complete bits from Optomux address 76.

```
A1100C2cr
```

Return message from Optomux indicating the proper level pulse has been measured for positions 12 and 8.



**READ PULSE DURATION COUNTERS****COMMAND e****PURPOSE**

Returns the pulse duration counter values for the specified positions.

**VERSIONS**

Digital

**FORMAT**

e[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. The pulse duration counter values will be returned for the module positions corresponding to 1 bits in the [positions] field. If [positions] is omitted, Optomux assumes a value of FFFF.

Pulse duration values are returned as 4 ASCII-Hex digits representing a single 16-bit number (0 to 65,535 decimal). Values are returned in sequence from the highest to lowest module position. Attempts to read an output will result in a string of four question marks "?????" being returned for that position.

If the "READ PULSE DURATION" command is used before the pulse has finished, the current duration will be returned. To verify that a pulse has completed, see the "READ PULSE COMPLETE BITS" command.

The duration of the measured pulse can be calculated as follows:

$$\text{duration} = [\text{returned data}] * \text{TRS} * 10 \text{ mSec.}$$

where TRS is the timer resolution setting as configured by the "Set Timer Resolution" command.

**EXAMPLE**

```
>FFe20083cr
```

Command sent from host requesting the counter value for position 9 on Optomux address FF.

```
A6090CFcr
```

Message returned from Optomux indicating that the duration of the pulse measured was 8.24 minutes. This example assumes a TRS value of 2.

$$\begin{aligned} 6,090 \text{ hex} &= 24,720 \text{ decimal} \\ 24,720 * \text{TRS} * 10 \text{ mSec.} &= 8.24 \text{ minutes} \end{aligned}$$

**READ AND CLEAR DURATION COUNTERS****COMMAND f****PURPOSE**

Returns pulse duration counters for the specified positions and then clears them to enable measurement of the next pulse.

**VERSIONS**

Digital

**FORMAT**

f[positions]

**REMARKS**

Performs READ PULSE DURATION COUNTERS function, then clears the counter and pulse-complete bits for the counters that were read. See READ PULSE DURATION COUNTERS.

Pulse duration values are returned as four ASCII-hex digits representing a single 16-bit number (0 to 65,535 decimal). Values are returned in sequence from the highest to lowest module position. Attempts to read an output will result in a string of four question marks "????" being returned for that position.

If the "READ AND CLEAR DURATION COUNTERS" command is used before the pulse has finished, the current duration will be returned, the counting will stop, and the pulse complete bit will not be set. To verify that a pulse has completed, see the "READ PULSE COMPLETE BITS" command.

The duration of the measured pulse can be calculated as follows:

$$\text{duration} = [\text{returned data}] * \text{TRS} * 10 \text{ mSec.}$$

Where TRS is the timer resolution setting as configured by the "Set Timer Resolution" command.

**EXAMPLE**

```
>01f1F8cr
```

Command sent from host requesting the counter value for position 0 of Optomux at address 1. The counter is then cleared after reading.

```
A6090CFcr
```

Message returned from Optomux indicating that the duration of the pulse measured was 8.24 minutes. This example assumes a TRS value of 2.

```
6,090 hex = 24,720 decimal
24,720 * TRS * 10 mSec. = 8.24 minutes
```

---

**CLEAR DURATION COUNTERS****COMMAND g**

---

**PURPOSE**

Clears the duration counters and pulse complete bits for the specified positions to enable measurement of the next pulse.

**VERSIONS**

Digital

**FORMAT**

g[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits that specify which positions are to be affected. Positions corresponding to 1 bits will have their duration counters and pulse-complete bits cleared. If the positions field is omitted all duration counters are cleared.

**EXAMPLE**

```
>22g82065cr
```

Clear duration counters for positions 5 and 11 on the Optomux I/O unit with address 22.



# Analog Read/Write Commands

---

**WRITE ANALOG OUTPUTS**
**COMMAND J**


---

**PURPOSE**

Instructs Optomux to output the 12-bit value specified to all positions represented by a 1 in the positions field.

**FORMAT**

J[positions] [data]

**REMARKS**

[positions] is four ASCII-hex characters. All four characters must be present, even if the B2 brain board is connected to a 4-point analog rack (P4AH).

[data] contains a 12-bit ASCII-hex value to be written to each position identified by a 1-bit in the [positions] field. Values range from 000h (0 decimal) to FFFh (4,095 decimal).

**EXAMPLE**

Assume Optomux at address FF has DA4 (0-5 V) output modules in positions 0-3. We want to set positions 1 and 3 to 1.25 volts.

$$(1.25 \text{ V} - 0 \text{ V}) / 5 \text{ V} * 4095 = 1024 \text{ decimal} = 400 \text{ Hex}$$

The command becomes:

```
>FFJ000A4003Bcr
```

---

**UPDATE ANALOG OUTPUTS****COMMAND S**

---

**PURPOSE**

Used to write values to one or more analog outputs.

**FORMAT**

S[positions] [data]

**REMARKS**

[positions] contains four ASCII-hex digits indicating which output positions are to be affected by the command. Output module positions corresponding to 1 bits in the positions field will have a value from the data field written to them.

[data] contains three ASCII-hex digits for each module position specified in the positions field. The 12-bit values are placed in the data field in order starting with the highest numbered module position and ending with the lowest numbered module position. Optomux will respond with a data field error if the command does not include three digits of data for each bit specified in the positions field.

Attempts to write to module positions that have been configured to function as inputs will be ignored.

**EXAMPLE**

```
>D0S0224FFF0C01F0ABcr
```

Instruct Optomux at address D0 to write the value 1F0 to position 2, 0C0 to position 5, and FFF to position 9.

**READ ANALOG OUTPUTS****COMMAND K****PURPOSE**

Returns a 12-bit value for each output identified by a 1-bit in the positions field.

**FORMAT**

K[positions]

**REMARKS**

[positions] contains 1-4 ASCII-Hex digits indicating which output positions are to be affected by the command. A zero value is assumed for leading positions that are omitted. Output module positions corresponding to 1 bits in the positions field will be read.

Returns three ASCII-hex characters for each position read. Returns "???" if the requested position is configured as an input. Since only three characters are returned for each position, do not subtract the offset of 1,000 hex from the value.

**EXAMPLE**

Positions 0–7 configured as inputs; Address 86.

```
>86K1004Acr
```

This command instructs Optomux to return the current value of position 8.

```
AB2EB9cr
```

This is the response if position 8 had a hex value of B2E.

*NOTE:* [positions] only contains three characters; a 0 value is assumed for leading positions which are omitted within the [positions] field. >86K01007Acr is a functionally identical command.

The following command requests the values for positions 7, 8, and 9:

```
>86K38054cr
```

A response of:

```
A000BE2???06cr
```

indicates that position 9 is an output with a value of 000 Hex, position 8 is an output with a value of BE2 hex, and position 7 is configured as an input.

**READ ANALOG INPUTS****COMMAND L****PURPOSE**

Returns the value of each input identified by a 1-bit in the positions field.

**FORMAT**

L[positions]

**REMARKS**

Returns four ASCII-hex characters for each position read. Returns "???" if the requested position is configured as an output.

The normal range of values for inputs is 1,000h to 1FFFh. If the most significant character is any character other than 1, it is an indication of over/under range. 0 = under range; >1 = over range

**EXAMPLE**

```
>90L5EAcr
```

This message reads the values of positions 0 and 2 from an Optomux at address 90 hex.

```
A1000188889cr
```

This response from Optomux indicates a value of 1,000 hex for position 2 and a value of 1,888 hex for position 0. Both values are valid readings in the range of 1,000 to 1FFF hex.

The next example shows what happens when a position read is configured as an output.

If the values for positions 10, 8, and 0 are requested with the following command message:

```
>90L5014Bcr
```

And positions 8 and 10 were configured as outputs, Optomux would return a response like the following:

```
A???????1FEEF9cr
```

Where position 10 is an output, position 8 is an output, and position 0 = 1FEE (a valid value).

If an output module is placed in a position that is configured as an input, and that value is requested, Optomux will return 0000. A value of 0000 will result in a decimal value of -4,096 after the 1,000 hex offset is subtracted.



**READ AND AVERAGE INPUTS****COMMAND M****PURPOSE**

Instructs Optomux to average the value of a single position over a specified number of samples and then to return the result.

**FORMAT**

M[position] [data]

**REMARKS**

[position] is a single ASCII-hex character representing the module position affected (0 to F).

[data] is two ASCII-hex digits representing the desired number of samples 01-FF hex (1-255 decimal).

Averaging is done using a continuous running average with a sample rate of 100 milliseconds. After the number of samples has been reached, the value is returned to the host. The following equation shows how the average is calculated:

$$\text{AVERAGE} = ((N-1) (\text{OLD AVERAGE}) + (\text{NEW READING})) / N$$

*NOTE:* This command only returns a response when it is done averaging. Therefore, if [data] is very large, there may be a long delay before you receive an acknowledgment message. Since this delay essentially ties up the whole bus for a period of time, it is recommended that you use the START INPUT AVERAGING, READ AVERAGE COMPLETE BITS, and READ INPUT AVERAGE DATA commands instead.

**EXAMPLE**

```
>03M70A58cr
```

Return the average of 10 samples of position 7 on Optomux at address 03.

**START INPUT AVERAGING****COMMAND T****PURPOSE**

Instructs Optomux to start averaging the value of input positions over a specified number of samples.

**FORMAT**

T[positions] [data]

**REMARKS**

[positions] contains four ASCII-hex digits indicating which input positions are to be affected by the command. Optomux will begin averaging input module positions corresponding to 1 bits in the positions field. All other positions will be unaffected.

[data] contains from 0 to 4 ASCII-hex digits specifying the number of consecutive readings to be averaged. If no data field is present, 0 samples will be averaged. The sample rate is 100 milliseconds. This command is used in conjunction with the "READ AVERAGE COMPLETE BITS," and the "READ INPUT AVERAGE DATA" commands.

Averaging is done using a continuous running average with a sample rate of 100 milliseconds. The number of samples to average is set with the START INPUT AVERAGE command. The READ AVERAGE COMPLETE BITS command indicates when the number of samples has been reached. The following equation shows how the average is calculated:

$$\text{AVERAGE} = ((N-1) (\text{OLD AVERAGE}) + (\text{NEW READING})) / N$$

This equation still applies, even after N has reached its limit, but keep in mind that N will retain the value of the limit as new readings are calculated.

**EXAMPLE**

```
>1AT04181F0Acr
```

Instruct Optomux at address 1A to start calculating the average of 1F samples for positions 3, 4, and 10.

---

**READ AVERAGE COMPLETE BITS****COMMAND i**

---

**PURPOSE**

Allows the host to determine which positions have completed averaging.

**FORMAT**

i

**REMARKS**

Optomux responds by returning a message that contains four ASCII-hex digits which specify the complete/incomplete status of the averaging process for all 16 module positions. Optomux has completed averaging the desired number of samples for module positions corresponding to 1 bits. Averaging has not been completed for positions corresponding to zero bits. Bits corresponding to positions that have been configured as outputs or to positions that Optomux has not been instructed to average values for should be ignored.

**EXAMPLE**

Assume that Start Input Averaging commands have been sent instructing Optomux start averaging at position 9, 10, and 11.

```
>A1iDBcr
```

Instruct Optomux at address A1 to return average complete bits.

```
A0A00D1cr
```

Response from Optomux indicating that averaging has been completed for positions 9 and 11. Optomux has not completed averaging position 10.

**READ INPUT AVERAGE DATA****COMMAND U****PURPOSE**

Instructs Optomux to return the results of averaging at specified input positions initiated by the Start input averaging command.

**FORMAT**

U[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits specifying the input positions that are to return averaged data. Optomux will return values for module positions corresponding to 1 bits in the positions field. If [positions] is omitted, Optomux will assume a value of FFFF. If Optomux has not completed averaging, the current value of the average will be returned. The READ AVERAGE COMPLETE BITS command provides a means of determining when averaging is complete.

Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position. Attempts to read the average of an output will result in four question marks ??? being returned.

Averaging is done using a continuous running average with a sample rate of 100 milliseconds. The number of samples to average is set with the START INPUT AVERAGE command. The READ AVERAGE COMPLETE BITS command indicates when the number of samples has been reached. The following equation shows how the average is calculated:

$$\text{AVERAGE} = ((N-1) (\text{OLD AVERAGE}) + (\text{NEW READING})) / N$$

This equation still applies, even after N has reached its limit, but keep in mind that N will retain the value of the limit as new readings are calculated.

**EXAMPLE**

```
>FFU2447cr
```

This message instructs Optomux at address FF to return the averages for positions 2 and 5.

```
A1A0D1F0CD0cr
```

Response from Optomux indicating the average for position 2 is 1FOC hex and the average for position 5 is 1A0D hex.

**READ TEMPERATURE INPUTS****COMMAND I****PURPOSE**

This command instructs Optomux to report the current temperature readings in degrees Celsius for the specified modules.

**VERSIONS**

Analog

**FORMAT**

`l[positions]`

*NOTE:* This command letter is a lowercase L, NOT an uppercase i.

**REMARKS:**

[positions] is a bitmask from 0 to 4 ASCII-hex digits representing the bitmask for the module positions. If [positions] is omitted, Optomux assumes a bitmask value of FFFF.

Optomux returns four ASCII-hex characters representing a signed 16-bit number for each channel corresponding to a 1 in the [positions] bitmask. The first three characters represent whole degrees Celsius while the last character represents the fractional part (sixteenths – remember it's hex) of degrees Celsius. In other words, the data returned is 16 times the temperature in degrees Celsius.

Optomux returns ??? for channels which are outputs or have not had a temperature probe type set.

Channels which read below the scale of the probe type which is set, will return a -273°C.

Channels which read above the scale for the probe type which is set, will return a 2,047°C.

**EXAMPLE**

```
>FF10A00C9cr
```

This command instructs Optomux to return the temperature values for inputs at positions 9 and 11.

This example assumes that the “SET TEMPERATURE PROBE TYPE” command was used previously for positions 9 and 11.

If the response from Optomux is as follows:

```
A08A8046BBDcr
```

This indicates that position 11 has a value of 08A8 hex and position 9 has a value of 046B hex. Converting the hex values to decimal, we get position 11 equal to 2,216, and position 9 equal to 1,131. Remember, these returned values are actually 16 times greater than the actual temperature, therefore we must divide these values by 16.

```
Position 11 Temperature = 2,216/16 = 138.5° C  
Position 9 Temperature = 1,131/16 = 70.68° C
```

In the next example, the following command is sent:

```
>FF1800C0cr
```

to read the temperature value from position 15. If the response is:

```
AF956EAcr
```

then the value for position 15 is F956. You will notice that the sign bit is set, therefore this value is a negative number. Converting the signed value of F956 to decimal results in a value of -1,705. Dividing -1,705 by 16 results in a temperature of -106.56°C.

**READ AVERAGE TEMPERATURE INPUTS****COMMAND o****PURPOSE**

This command instructs Optomux to report the average temperature readings in degrees Celsius for the specified modules.

**VERSIONS**

Analog

**FORMAT**

`o[positions]`

**REMARKS**

[positions] is a bitmask from 0 to 4 ASCII-hex digits representing the bitmask for the module positions. If [positions] is omitted, Optomux assumes a bitmask value of FFFF.

Optomux returns four ASCII-hex characters representing a signed 16-bit number for each channel corresponding to a 1 in the [positions] bitmask. The first three characters represent whole degrees Celsius while the last character represents the fractional part (sixteenths – remember it's hex) of degrees Celsius. In other words, the data returned is 16 times the temperature in degrees Celsius.

Optomux returns ??? for channels that are outputs or have not had a temperature probe type set.

Averaging must have been previously initiated with the Start Input Averaging command T.

Channels that read below the scale of the probe type which is set will return a -273°C.

Channels that read above the scale for the probe type which is set will return a 2047°C.

**EXAMPLE**

```
>FFo0A00CCcr
```

This command instructs Optomux to return the temperature values for inputs at positions 9 and 11.

This example assumes that the “SET TEMPERATURE PROBE TYPE” and “START INPUT AVERAGING” commands were previously used for positions 9 and 11.

If the response from Optomux is as follows:

```
A08A8046BBDcr
```

This indicates that position 11 has a value of 08A8 hex and position 9 has a value of 046B hex. Converting the hex values to decimal, we get position 11 equal to 2,216, and position 9 equal to 1,131. Remember, these returned values are actually 16 times greater than the actual temperature, therefore we must divide these values by 16.

```
Position 11 Temperature = 2,216/16 = 138.5° C  
Position 9 Temperature = 1,131/16 = 70.68° C
```

In the next example, the following command is sent:

```
>FFo8000C3cr
```

to read the average temperature value from position 15. If the response is:

```
AF956EAcr
```

then the value for position 15 is F956. You will notice that the sign bit is set, therefore this value is a negative number. Converting the signed value of F956 to decimal results in a value of -1,705. Dividing -1,705 by 16 results in a temperature of -106.56°C.



# Analog Input Range Commands

---

**SET INPUT RANGE**

---

**COMMAND N****PURPOSE**

Defines an input range (high and low limits) for the specified input positions.

**FORMAT**

N[positions] [data]

**REMARKS**

All four positions characters must be supplied.

[data] must be six ASCII-hex characters; three representing the high limit (12-bits) followed by three representing the low limit. Latches are set when either the high or low limits are exceeded. See Read Out-Of-Range Latches.

**EXAMPLE**

```
>09N0003760710AFcr
```

For positions 0 and 1 on Optomux at address 09 hex, set the upper limit at 760 hex and the lower limit at 710 hex. The defined range of values as read back by the Read Analog Inputs command will be from 1,710 to 1,760.

---

**READ OUT-OF-RANGE LATCHES**

---

**COMMAND 0****PURPOSE**

Reads the high and low Out-Of-Range latches.

**FORMAT**

- (Hex 4F)

**REMARKS**

Response is eight ASCII-hex characters.

The first four characters show the status of the high-limit latches; the remaining four show the low-limit latches. Positions corresponding to 1 bits in the first 16-bit value have exceeded their high limits. Positions corresponding to 1 bits in the second 16-bit value have fallen below their low limits.

**EXAMPLE**

```
>710B7cr
```

Read Out-Of-Range latches at Optomux address 71 Hex.

```
A0001001062cr
```

Position 0 has exceeded the high limit since the latches were last cleared, and position 4 has exceeded the low limit.

---

**READ AND CLEAR OUT-OF-RANGE LATCHES****COMMAND P**

---

**PURPOSE**

This command performs the Read Out-Of-Range Latches command (all positions), then resets the latches specified by the [positions] field.

**FORMAT**

P[positions]

**REMARKS**

Although all latches are read, only those corresponding to 1 bits in the [positions] field are reset. If no [positions] field is present, all latches are cleared.

The data portion of the response is eight ASCII-hex characters in the format HHHHLLLL.

The first four characters show the status of the high-limit latches, the remaining four show the low-limit latches. Positions corresponding to 1 bits in the first 16-bit value have exceeded their high limits. Positions corresponding to 1 bits in the second 16-bit value have fallen below their low limits.

**EXAMPLE**

```
>70P4EBcr
```

Read Out-Of-Range Latches and reset latch at position 2 at Optomux address 70.

```
A0001000465cr
```

Module 0 has exceeded the upper limit, module 2 has exceeded the lower limit; only position 2 will have the latch reset.

---

**CLEAR OUT-OF-RANGE LATCHES**

---

**COMMAND Q****PURPOSE**

Clears Out-Of-Range Latches for specified positions.

**FORMAT**

Q[positions]

**REMARKS**

Optomux will clear the Out-Of-Range latches for input positions corresponding to 1 bits in the positions field. If no positions field is supplied, all latches are reset.

**EXAMPLES**

```
>00Q5E6cr
```

Clears latches at positions 0 and 2 at Optomux address 00 hex.

```
>99QC3cr
```

Clears all latches at Optomux address 99 hex.

**READ LOWEST VALUES****COMMAND a****PURPOSE**

Instructs Optomux to return the lowest readings for specified input positions.

**FORMAT**

```
a[positions]
```

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits that specify which positions are to be affected by the command. If [positions] is omitted, Optomux will assume a bitmask of FFFF. Optomux will return the lowest reading for each position corresponding to 1 bits in the positions field. The reading will be the lowest Optomux has encountered since last receiving a Read And Clear Lowest Values or Clear Lowest Values command. Optomux sets all low values to 3,000 hex (over range) upon power-up.

Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position.

**EXAMPLE**

```
>2Aa4A17Acr
```

Instructs Optomux at address 2A to return the low values for positions 0, 5, 7, and 10.

```
A130110010FFA100D59cr
```

Response from Optomux indicating:

```
Lowest Value for position 0 = 100D
Lowest Value for position 5 = 0FFA under range
Lowest Value for position 7 = 1,001
Lowest Value for position 10 = 1,301
```

---

**CLEAR LOWEST VALUES**

---

**COMMAND b****PURPOSE**

Instructs Optomux to clear the lowest readings for specified input positions.

**FORMAT**

b[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits that specify which positions are to be affected by the command. If [positions] is omitted, Optomux will assume a bitmask of FFFF. Optomux will clear the lowest reading for each input position corresponding to 1 bits in the positions field by setting each positions lowest value to 3,000 hex (extreme over-range). This will allow Optomux to store the lowest value encountered in subsequent readings.

**EXAMPLE**

```
>2Ab4A17Bcr
```

Instructs Optomux at address 2A to clear the lowest values for positions 0, 5, 7, and 10.

**READ AND CLEAR LOWEST VALUES****COMMAND c****PURPOSE**

Instructs Optomux to read and clear the lowest readings for specified input positions.

**FORMAT**

c [positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits that specify which positions are to be affected by the command. If [positions] is omitted, Optomux will assume a bitmask of FFFF. Optomux will return the lowest reading for each position corresponding to 1 bits in the positions field. The values returned will be the lowest Optomux has encountered since last receiving a Read And Clear Lowest Values or Clear Lowest Values command. Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position.

Optomux will clear the lowest value for each of the specified positions by setting each positions lowest value to 3,000 hex (extreme over-range). This will enable Optomux to store the lowest value encountered in subsequent readings.

**EXAMPLE**

```
>2Ac4A17Ccr
```

Instructs Optomux at address 2A to return the low values for positions 0, 5, 7, and 10 and then to clear them.

```
A130110010FFA100D59cr
```

Response from Optomux indicating:

```
Lowest Value for position 0 = 100D
Lowest Value for position 5 = 0FFA (under-range)
Lowest Value for position 7 = 1,001
Lowest Value for position 10 = 1,301
```

**READ PEAK VALUES****COMMAND d****PURPOSE**

Instructs Optomux to return the peak readings for specified input positions.

**FORMAT**

```
d[positions]
```

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits that specify which positions are to be affected by the command. If [positions] is omitted, Optomux will assume a bitmask of FFFF. Optomux will return the peak reading for each position corresponding to 1 bits in the positions field. The value will be the highest Optomux has encountered since last receiving a Read And Clear Peak Values or Clear Peak Values command. Optomux sets all peak values to 0000 Hex (under range) upon power-up.

Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position.

**EXAMPLE:**

```
>2AdAA089cr
```

Instructs Optomux at address 2A to return the peak values for positions 5, 7, 9, and 11.

```
A1DFF10C11FAC1FB0BAcr
```

Response from Optomux indicating:

```
peak Value for position 5 = 1FB0
peak Value for position 7 = 1FAC
peak Value for position 9 = 10C1
peak Value for position 11 = 1DFF
```



---

**CLEAR PEAK VALUES****COMMAND e**

---

**PURPOSE**

Instructs Optomux to clear the peak readings for specified input positions.

**FORMAT**

e [positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits that specify which positions are to be affected by the command. If [positions] is omitted, Optomux will assume a bitmask of FFFF. Optomux will clear the peak values for each input position corresponding to 1 bits in the positions field by setting the peak value for each position to 0000 hex (under range). This will allow Optomux to store the highest value encountered in subsequent readings.

**EXAMPLE**

```
>2Ae3C07Ecr
```

Instructs Optomux at address 2A to clear the peak values for positions 6, 7, 8, and 9.

---

**READ AND CLEAR PEAK VALUES****COMMAND f**

---

**PURPOSE**

Instructs Optomux to read and clear the peak values for specified input positions.

**FORMAT**

f[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits that specify which positions are to be affected by the command. If [positions] is omitted, Optomux will assume a bitmask of FFFF. Optomux will return the peak value for each position corresponding to 1 bits in the positions field. The values returned will be the highest Optomux has encountered since last receiving a Read and Clear Peak Values or Clear Peak Values command. Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position.

Optomux will clear the peak value for each of the specified positions by setting each positions' peak value to 0000 hex (under-range). This will enable Optomux to store the highest value encountered in subsequent readings.

**EXAMPLE**

```
>2AfC000ACcr
```

Instructs Optomux at address 2A to return and then clear the peak values for positions 14 and 15.

```
A1DFA1C1CE4
```

Response from Optomux indicating:

```
Peak Value for position 14 = 1C1C  
Peak Value for position 15 = 1DFA
```

# Analog Gain/Offset Commands

---

**CALCULATE OFFSETS**
**COMMAND g**


---

**PURPOSE**

Instructs Optomux to calculate and return offsets for specified input positions.

**FORMAT**

```
g [positions]
```

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Optomux will return offset values for input positions corresponding to 1 bits in the positions field. If [positions] is omitted, Optomux will assume a bitmask of FFFF. The offset values are calculated using the current values of the inputs. Therefore, this command is only meaningful when the specified module positions are receiving the value you wish to consider zero scale.

This command is usually used during system installation and calibration when known inputs (zero scale) can be applied to the modules. The values obtained can be used during Optomux initialization. Optomux can be instructed to use these offset values by including them in the Set Offsets command.

When utilizing the gain coefficients feature, it is necessary to set offsets for module positions before setting the gain coefficients.

Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position in the same form required by the Set Offsets command. Negative offsets are returned as the complement of the number 2.

```
0001 = +1 Offset
FFFF = -1 Offset
```

Attempts to calculate the Offset for an output will result in four question marks ??? being returned.

**EXAMPLE**

```
>DFgF37cr
```

Instructs Optomux at address DF to calculate and return offset values for positions 0, 1, 2 and 3.

Response from Optomux -

```
A000AFFFE000100036Ccr
```

Offset for position 0 = 0003

Offset for position 1 = 0001

Offset for position 2 = FFFE

Offset for position 3 = 000A

```
>AFg5555C2cr
```

Instructs Optomux at address AF to calculate and return offset values for positions 0, 2, 4, 6, 8, 10, 12, and 14.

Response from Optomux -

```
A0020FFE9000EFFFD????????001A000D55cr
```

Offset for position 0 = 000D

Offset for position 2 = 001A

Offset for position 4 = ????

Offset for position 6 = ????

Offset for position 8 = FFFD

Offset for position 10 = 000E

Offset for position 12 = FFE9

Offset for position 14 = 0020

---

**SET OFFSETS****COMMAND W**

---

**PURPOSE**

Instructs Optomux to set offsets for specified input positions.

**FORMAT**

W[positions] [data]

**REMARKS**

[positions] contains four ASCII-hex digits indicating which input positions are to be affected by the command. Values from the data field will be used as Offsets for input module positions corresponding to 1 bits in the positions field.

[data] contains four ASCII-hex digits for each module position specified in the positions field. The values are placed in the data field in order starting with the highest numbered module position and ending with the lowest numbered module position. Optomux will respond with a data field error if the command does not include four digits of data for each bit specified in the positions field.

Offsets are placed in the [data] field in the same form as they are returned by the Calculate Input Offsets command. Negative offsets are represented as the complement of the number 2.

```
0001 = +1 offset  
FFFF = -1 offset
```

Attempts to set Offsets for positions that have been configured to function as outputs will be ignored.

When utilizing the gain coefficients feature, it is necessary to set offsets for module positions before setting the gain coefficients.

**EXAMPLE**

```
>DFW000F000AFFFE0001000323cr
```

Instructs Optomux at address DF to set the following offsets:

Offset for position 0 = 0003

Offset for position 1 = 0001

Offset for position 2 = FFFE

Offset for position 3 = 000A

```
>DFW55550020FFE9000EFFFD00000000001A000D92cr
```

Instructs Optomux at address DF to set the following offset values for positions 0, 2, 4, 6, 8, 10, 12, and 14.

Offset for position 0 = 000D

Offset for position 2 = 001A

Offset for position 4 = 0000

Offset for position 6 = 0000

Offset for position 8 = FFFD

Offset for position 10 = 000E

Offset for position 12 = FFE9

Offset for position 14 = 0020

**CALCULATE AND SET OFFSETS****COMMAND h****PURPOSE**

Instructs Optomux to calculate and set offsets for specified input positions. The calculated offsets are returned to the host.

**FORMAT**

```
h[positions]
```

**REMARKS**

[positions] contains from 0 to four ASCII-hex digits. Optomux will set and return offset values for input positions corresponding to 1 bits in the positions field. If [positions] is omitted, Optomux will assume a bitmask of FFFF. The offset values are calculated using the current values of the inputs. Therefore, this command is only meaningful when the specified module positions are receiving the value you wish to consider zero scale.

This command is usually used during system installation and calibration when known inputs (zero scale) can be applied to the modules.

Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position in the same form required by the Set Offsets command. Negative offsets are returned as the complement of the number 2.

```
0001 = +1 offset
FFFF = -1 offset
```

Attempts to calculate and set the offset for an output will result in four question marks ??? being returned for that position.

When utilizing the gain coefficients feature, it is necessary to set offsets for module positions before setting the gain coefficients.

**EXAMPLE**

```
>00h302592cr
```

Instructs Optomux at address 00 to calculate and set offsets for positions 0, 2, 5, 12, and 13.

```
A000B0010FFFE000100032Ecr
```

Response from Optomux indicating that the following offsets have been set:

```
Offset for position 0 = 0003  
Offset for position 2 = 0001  
Offset for position 5 = FFFE  
Offset for position 12 = 0010  
Offset for position 13 = 000B
```

```
>DFh5555C6cr
```

Instructs Optomux at address DF to calculate and set offset values for positions 0, 2, 4, 6, 8, 10, 12, and 14.

```
A0020FFE9000EFFFDF???????001A000D55cr
```

Response from Optomux indicating that the following offset values have been set:

```
Offset for position 0 = 000D  
Offset for position 2 = 001A  
Offset for position 4 = ??? Is an Output  
Offset for position 6 = ??? Is an Output  
Offset for position 8 = FFFD  
Offset for position 10 = 000E  
Offset for position 12 = FFE9  
Offset for position 14 = 0020
```



**CALCULATE GAIN COEFFICIENTS****COMMAND X****PURPOSE**

Instructs Optomux to calculate and return gain coefficients for specified input positions.

**FORMAT**

X[positions]

**REMARKS**

[positions] contains from 0 to four ASCII-hex digits. Optomux will calculate and return gain coefficients for input positions corresponding to 1 bits in the positions field. If [positions] is omitted, Optomux assumes a bitmask of FFFF. The gain coefficients are calculated using the current values of the inputs. Therefore, this command is only meaningful when the specified module positions are receiving the value you wish to consider full-scale.

This command is usually used during system installation and calibration when known input values (full-scale) can be applied to the modules. The values obtained can be used during Optomux initialization. Optomux can be instructed to use these gain coefficients by including them in the Set Input Gain Coefficients command.

Offsets should be set before setting gain coefficients.

Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position in the same form required by the Set Input Gain Coefficients command. A decimal point is assumed to be between the first digit and the last three. The first ASCII-hex digit represents the whole part of the coefficient and the last three digits represent the fractional part. On power-up, all gain coefficients are set to 1,000 hex.

Attempts to calculate the gain coefficient for an output will result in four question marks ??? being returned.

**EXAMPLE**

```
>DFXF28cr
```

Instructs Optomux at address DF to calculate and return gain coefficients for positions 0, 1, 2, and 3.

Response from Optomux -

```
A1400101E1019114835cr
```

	<b>hex</b>	<b>decimal</b>
Coefficient for position	0 = 1148	1.0800
Coefficient for position	1 = 1019	1.0060
Coefficient for position	2 = 101E	1.0073
Coefficient for position	3 = 1400	1.2500

```
>AFX5001A5cr
```

Instructs Optomux at address AF to calculate and return gain coefficients for positions 0, 12, and 14.

Response from Optomux -

```
A130A????126EAFcr
```

	<b>hex</b>	<b>decimal</b>
Coefficient for position	0 = 126E	1.1520
Coefficient for position	12 = ????	
Coefficient for position	14 = 130A?	1.1900

## SET GAIN COEFFICIENTS

## COMMAND Y

**PURPOSE**

Instructs Optomux to set gain coefficients for specified module positions.

**FORMAT**

Y[positions] [data]

**REMARKS**

[positions] contains four ASCII-hex digits indicating which input positions are to be affected by the command. Values from the data field will be used as gain coefficients for input module positions corresponding to 1 bits in the positions field.

[data] contains four ASCII-hex digits for each module position specified in the positions field. The values are placed in the data field in order starting with the highest numbered module position and ending with the lowest numbered module position. Optomux will respond with a data field error if the command does not include four digits of data for each bit specified in the positions field.

Gain coefficients are in the same form as those returned by the Calculate Gain Coefficients command. The first digit represents the whole part of the coefficient and the last three digits represent the fractional part.

Attempts to set coefficients for positions that have been configured to function as outputs will be ignored.

It is necessary to set offsets for module positions before setting the gain coefficients.

**EXAMPLE**

```
>DFY000F1400101E10191148EEcr
```

Instructs Optomux at address DF to set the following gain coefficients.

	<b>hex</b>	<b>decimal</b>
Coefficient for position	0 = 1148	1.0803
Coefficient for position	1 = 1019	1.0063
Coefficient for position	2 = 101E	1.0075
Coefficient for position	3 = 1400	1.2500

```
>AFY500113081333126E1Acr
```

Instructs Optomux at address AF to set the following gain coefficients:

	<b>hex</b>	<b>decimal</b>
Coefficient for position	0 = 126E	1.1520
Coefficient for position	12 = 1333	1.2000
Coefficient for position	14 = 1308	1.1900

---

**CALCULATE AND SET GAIN COEFFICIENTS****COMMAND Z**

---

**PURPOSE**

Instructs Optomux to calculate and set gain coefficients for specified input positions. The calculated coefficients are returned to the host.

**FORMAT**

Z[positions]

**REMARKS**

[positions] contains from 0 to 4 ASCII-hex digits. Optomux will set and return gain coefficients for input positions corresponding to 1 bits in the positions field. If [positions] is omitted, Optomux will assume a bitmask of FFFF. The gain coefficients are calculated using the current values of the inputs. Therefore, this command is only meaningful when the specified module positions are receiving the value you wish to consider full-scale.

This command is usually used during system installation and calibration when known inputs (full-scale) can be applied to the modules.

Four ASCII-hex digits are returned for each position specified in the positions field. Values are returned in sequence from the highest to lowest module position in the same form required by the Set Gain Coefficients command. The first digit represents the whole part of the coefficient and the last three digits represent the fractional part.

Attempts to calculate and set the gain coefficients for an output will result in four question marks ??? being returned for that position.

When utilizing the gain coefficients feature, it is necessary to set offsets for module positions before setting the gain coefficients.

**EXAMPLE**

```
>DFZF2Acr
```

Instructs Optomux at address DF to calculate and set gain coefficients for positions 0, 1, 2, and 3.

```
A103111C2127E13684Dcr
```

Response from Optomux indicating that the following gain coefficients have been set:

	<b>hex</b>	<b>decimal</b>
Coefficient for position	0 = 1368	1.213
Coefficient for position	1 = 127E	1.156
Coefficient for position	2 = 11C2	1.110
Coefficient for position	3 = 1031	1.012

```
>AFZ5001A7cr
```

Instructs Optomux at address AF to calculate and set gain coefficients for positions 0, 12, and 14.

```
A13081333126E74cr
```

Response from Optomux indicating that the following gain coefficients have been set.

	<b>hex</b>	<b>decimal</b>
Coefficient for position	0 = 126E	1.152
Coefficient for position	12 = 1,333	1.200
Coefficient for position	14 = 1,308	1.190



# Analog Waveform Commands

---

**SET OUTPUT WAVEFORM****COMMAND R**

---

**PURPOSE**

Initiates a constant waveform at any of the output positions.

**FORMAT**

R[positions] [modifiers] [data]

**REMARKS**

All four [positions] characters must be present.

[modifiers] contains two ASCII-hex characters. The first character specifies the rate of the waveform. Possible rates are given in Table 1 below. The second modifier character specifies the type of waveform (see table 2).

*NOTE:* A "0" rate terminates the current waveform; no other data is required.

[data] contains four ASCII-hex digits. The first two ASCII-hex characters represent the upper 8- bits of the 12-bit high limit of the waveform. The lower 4 bits are 0000. The second two ASCII-hex characters represent the upper 8-bits of the 12-bit low limit of the waveform. The lower 4- bits are 0000.

Period of entire ramp — from 000 to FFF. To calculate actual period, determine the percentage of a full 0 to FFF transition your limits represent. Multiply this fraction by the full-scale period. This result is the actual period of 1/2 of a triangle or square wave. It is the entire period of a ramp.

**Table 15-1: Possible Waveform Rates**

0	Disable waveform	8	1.09 minutes
1	2.18 minutes	9	32.8 seconds
2	3.28 minutes	A	21.8 seconds
3	4.37 minutes	B	16.4 seconds
4	5.46 minutes	C	13.1 seconds
5	6.56 minutes	D	10.9 seconds
6	7.65 minutes	E	9.4 seconds
7	8.74 minutes	F	8.2 seconds

**Table 15-2: Possible Waveform Types**

Character	Waveform Type
0	Square Wave with 50% duty cycle
1	Triangle wave with a positive initial slope
2	Ramp Up - waveform terminates upon reaching the upper limit
3	Continuous Ramp Up
4	Square wave (50% duty cycle)
5	Triangle wave; initial slope = DOWN
6	Ramp Down, waveform terminates at lower limit
7	Continuous Ramp Down

**EXAMPLE**

```
>00R000183A036B8cr
```

At Optomux 00, position 0, initiate a continuous Ramp Up with:

Upper Limit = A00

Lower Limit = 360

Period = 1.09 minutes

```
>AAR007070AA00E4cr
```

At Optomux AA, positions 4, 5, and 6, initiate a square wave with:

Upper Limit = AA0

Lower Limit = 000

Period = (AA / FF) \* 8.74 min) = (0.66 \* 8.74 min) = 11.6 min



## IMPROVED OUTPUT WAVEFORMS

## COMMAND V

**PURPOSE**

Initiates a constant waveform at any of the output positions.

**FORMAT**

V[positions] [modifier] [data]

**REMARKS**

[positions] contains four ASCII-hex digits indicating which output positions are to be affected. Module positions corresponding to 1 bits are affected by the command.

[modifier] is a single ASCII-hex digit that specifies which type of waveform is to be generated. Possible waveform types are given in the table below.

*NOTE:* A type "0" wave form terminates the current waveform.

**Table 15-3: Possible Waveform Types**

<b>Modifier</b>	<b>Waveform Type</b>
0	Waveform Off
1	Triangle wave with a positive initial slope
2	Ramp Up, waveform terminates upon reaching the upper limit
3	Sawtooth, continuous ramp up
4	Square Wave, 50% duty cycle
5	Triangle wave with negative initial slope
6	Ramp Down, waveform terminates at lower limit
7	Sawtooth, continuous ramp down

[data] contains a 3-digit value representing the high limit of the waveform, a 3-digit value representing the low limit, and a 4-digit value representing the period of the waveform. The order is important.

high limit - three ASCII-hex digits representing the high limit of the wave form.  
000 = zero scale, FFF = full scale

low limit - three ASCII-hex digits representing the low limit of the wave form.  
000 = zero scale, FFF = full scale

period - four ASCII-hex digits indicating the period of the wave form. The period can range from 0001 to 7FFF, each count representing 100 milliseconds (0.1 second to 54.61 minutes). The value represents the period of 1/2 of a triangle or square wave. It is the entire period of sawtooths and ramps.

*NOTE:* The waveform is updated by Optomux every 50 milliseconds, therefore, waveforms with short periods may not provide a smooth output.

### EXAMPLES

```
>FEV00081FF00C0007808cr
```

Instruct Optomux at address FE to output a triangle wave at positions 3 with a high limit of FF0, a low limit of 00C, and a period of 12 seconds.

```
>01V00243F0A00C0078D9cr
```

Instruct Optomux at address 01 to output a continuous ramp up (sawtooth) at positions 2 and 5 with a high limit of F0A, a low limit of 00C, and a period of 12 seconds.

# Revision Identification Commands

(for digital surface mount brains only)

---

CHECKSUM ON ROM

COMMAND \_

---

## PURPOSE

Initiates brain firmware revision by calculating the unique checksum of the ROM.

## FORMAT

—

## REMARKS

\_ is the underscore character. The underscore is an ASCII 5F hex.

## EXAMPLE

```
>00_BFcr
```

---

**DATE OF FIRMWARE****COMMAND `**

---

**PURPOSE**

Identifies brain firmware revision by date of release.

**FORMAT**

`

**REMARKS**

` is the single quote under the tilde sign on the computer keyboard and is an ASCII 60 hex.

**EXAMPLE**

```
>00`C0cr
```

# Appendix A

## HEX - ASCII Character Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0 (1) ^@ NUL	16 ▶ ^P DLE	32 (1) space	48 0	64 @	80 P	96 ,	112 p	128 Ç	144 É	160 á	176 █	192 L	208 ll	224 α	240 ≡
1	1 ☺ ^A SOH	17 ▼ ^Q DC1	33 !	49 1	65 A	81 Q	97 a	113 q	129 ù	145 æ	161 í	177 █	193 ⊥	209 ⌚	225 β	241 ±
2	2 ☹ ^B STX	18 ↑ ^R DC2	34 “	50 2	66 B	82 R	98 b	114 r	130 é	146 Æ	162 ó	178 █	194 T	210 ⌚	226 Γ	242 ≥
3	3 ♥ ^C ETX	19 !!	35 #	51 3	67 C	83 S	99 c	115 s	131 â	147 ô	163 ú	179 	195 	211 ll	227 π	243 ≤
4	4 ♦ ^D EOT	20 π	36 \$	52 4	68 D	84 T	100 d	116 t	132 ä	148 ö	164 ñ	180 	196 -	212 ll	228 Σ	244 
5	5 ♣ ^E ENQ	21 § ^U NAK	37 %	53 5	69 E	85 U	101 e	117 u	133 à	149 ò	165 Ñ	181 	197 +	213 F	229 σ	245 J
6	6 ♠ ^F ACK	22 ■ ^V SYN	38 &	54 6	70 F	86 V	102 f	118 v	134 â	150 û	166 a	182 	198 	214 ll	230 μ	246 ÷
7	7 • ^G BEL	23 ↑ ^W ETB	39 ,	55 7	71 G	87 W	103 g	119 w	135 ç	151 ù	167 °	183 	199 	215 	231 τ	247 ≈
8	8 █ ^H BS	24 ↑ ^X CAN	40 (	56 8	72 H	88 X	104 h	120 x	136 ê	152 ÿ	168 ¿	184 	200 ll	216 	232 Φ	248 °
9	9 ○ ^I HT	25 ↓ ^Y EM	41 )	57 9	73 I	89 Y	105 i	121 y	137 ë	153 ÿ	169 r	185 	201 ll	217 J	233 ⊖	249 •
A	10 ○ ^J LF	26 →	42 *	58 :	74 J	90 Z	106 j	122 z	138 è	154 Û	170 ¬	186 	202 ll	218 r	234 Ω	250 ·
B	11 ♂ ^K VT	27 ←	43 +	59 ;	75 K	91 [	107 k	123 {	139 ï	155 ç	171 ½	187 	203 ll	219 █	235 ø	251 √
C	12 ♀ ^L FF	28 L	44 ,	60 <	76 L	92 \ l	108 l	124 ;	140 î	156 £	172 ¼	188 ll	204 ll	220 █	236 ∞	252 n
D	13 ¶ ^M CR	29 ↔	45 -	61 =	77 M	93 ]	109 m	125 }	141 ì	157 ¥	173 ;	189 ll	205 =	221 █	237 ∅	253 ²
E	14 ^N SO	30 ▲	46 .	62 >	78 N	94 ^	110 n	126 ~	142 Ä	158 B	174 «	190 	206 ll	222 █	238 ε	254 █
F	15 ☒ ^O SI	31 ▼	47 /	63 ?	79 O	95 —	111 o	127 △	143 Å	159 f	175 »	191 	207 ll	223 █	239 ∩	255 (1)

(1) = No symbol defined

## HEX To Binary Conversion Table

Hex	Binary
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

# Appendix B

## IMPLEMENTING DRIVERS FOR OPTOMUX

Opto 22 has drivers available for Optomux I/O on DOS, Win16, and Win32. However, to implement a driver on other platforms, a “generic” Optomux driver is available. The “generic” Optomux driver builds packets to be sent to Optomux hardware and parses the response received. It handles all but the communications with the serial device since the serial device is usually platform-dependant. This “generic” Optomux driver is available in the OptoDriver Toolkit.

When implementing drivers, it's sometimes helpful to use a computer running a terminal program as a “sniffer” to monitor Optomux communications. Normal communications, from a utility such as OptoScan, can be compared to communications from the driver being developed.







# Appendix C

## Surge Protection For RS-422/485 Communication Links

For more information or for catalog requests, contact:

Phoenix Contact Inc.  
P.O Box 4100  
Harrisburg, PA 17111-0100

E-mail: [info@phoenixcon.com](mailto:info@phoenixcon.com)  
Phone: (717) 944-1300  
Fax: (717) 944-1625  
Fax-on-demand: (800) 944-9901



# Appendix D

## Troubleshooting Questions and Answers

- Q.** I send a command to turn on output 5 of an Optomux at address 255 and output 4 goes on.
- A.** Module positions are numbered from 0 to 15, therefore the fifth bit in the bitmask is for position labeled 4. Following are the bitmask values for each position:

Position	Bitmask
0	0001
1	0002
2	0004
3	0008
4	0010
5	0020
6	0040
7	0080
8	0100
9	0200
10	0400
11	0800
12	1000
13	2000
14	4000
15	8000

- Q.** I send a command to activate several outputs, the Optomux responds with no error, but none of my outputs come on.
- A.** If an output does not turn on, check the following:
1. Make sure you configured the outputs of the Optomux correctly. Optomux defaults to all positions configured as inputs on power-up.
  2. If power was lost at any time and returned, Optomux will have lost its configuration. Optomux will have reported a N00 error to the first command sent after the power-up to indicate that configuration was lost.
  3. If using a digital Optomux, make sure you are using output modules with a 5 VDC logic voltage (OAC5, ODC5, etc.). Sometimes modules with a 15 VDC or 24 VDC logic voltage (ODC15, OAC24, etc.) are inserted by mistake. The LED may turn on or be dim with the 15 VDC and 24 VDC modules but there will be no output on the field side.
  4. If the output LED is on, but the load does not turn on, check the field voltage and wiring. Also, the load may not draw enough current to meet the minimum current requirements of OAC5 modules (20 mA).
- Q.** I send a command message to Optomux at address 0, and I get no response. However, the Optomux's REC light flashes.
- A.** The receive lights on all the Optomux boards wired in a multidrop mode should light whenever a command is sent. Only the Optomux at the address that matches the command message address should respond. When an Optomux unit responds, the XMT (transmit) light will flash. At high baud rates, and for messages with only a short response ("A" only), the flash will be so brief it may be missed visually.
- If the host receives no response, and no flash of the XMT light is visible at the slowest baud rate (300) then check the following:
1. Make sure the address in the command message matches the Optomux address. Check the jumpers. A common mistake is a reversal of jumpers. All address jumpers (B0 through B7) installed are address 0, and all address jumpers (B0 through B7) removed corresponds to address 255.
  2. Make sure the baud rate is correct between the host and the Optomux. All Optomux units on the same link should be configured to the same baud rate.
  3. Make sure you have a solid +5 volts powering the Optomux unit. If the voltage is too low, the RCV light may flash, but Optomux will not be able to respond. Measure the voltage at the Optomux power terminals — not at the power supply.
  4. Check to make sure the communications link is wired with the correct polarity and that the A group of jumpers is configured correctly.

- Q.** When I read position 3 of an Optomux analog board, I receive a 0000, which converts to a decimal -4,096 value when I subtract the 1,000 hex offset.
- A.** A -4,096 decimal reading may indicate one of the following:
1. You are reading a module position that has no input module installed.
  2. You are reading a thermocouple module that has no thermocouple installed or the thermocouple probe is open.
  3. An ICTD module which has the ICTD wired in reverse.
  4. A 4–20 mA module is wired with reverse polarity.
  5. The field connections are made to the wrong terminals. Field connections vary with each module. Refer to the Optomux Family Data Book (Form #524) for information on wiring each specific module. In general, field connections made to the terminals on the rack are made to the terminals labeled UPPER (closest to module), and if the module has terminals on the top of the module, connections would be made there. Analog racks have the terminal channels labeled from 1 to 4 (or 16). These channels correspond to module positions 0 to 3 (or 15) respectively.
  6. If all inputs on that board have the same -4,096 reading, then check to make sure the board has +15 and -15 VDC at the corresponding terminals (with reference to the terminal that is marked COMMON).
- Q.** I receive a large number of checksum errors (N02) when I send commands to Optomux.
- A.** Make sure that you are using twisted pair cable. The RS-422/485 network is only reliable when the communications cable is twisted (+ and - lines of EACH pair twisted together), with at least 1/2 twist per inch. Sometimes, the wire is twisted but one of the connections of a pair is actually used as the mate to the opposite pair (- of one pair used as - of opposite pair). This cross-twist condition is usually due to jacketed twisted pair cables which have a wire of each pair with the same color code. In this case, strip the jacket back far enough to properly identify the individual pairs.

You should also check the Group A jumpers to make sure the termination and bias jumpers are correctly installed.



# Product Support

If you have any questions about this product, contact Opto 22 Product Support Monday through Friday, 8 a.m. to 5 p.m. Pacific Time.

<b>Phone:</b>	800/TEK-OPTO (835-6786) 909/695-3080
<b>Fax:</b>	909/695-3017
<b>E-mail:</b>	support@opto22.com
<b>Bulletin Board System (BBS):</b>	909/695-1367 <a href="http://bbs.opto22.com">http://bbs.opto22.com</a> (24 hours a day, 7 days a week)
<b>OptoWorld Wide Web site:</b>	<a href="http://www.opto22.com">http://www.opto22.com</a>

When accessing the BBS, use the following modem settings:

- No parity, 8 data bits, 1 stop bit
- Baud rates up to 28,800
- Z-modem protocol for uploads and downloads (optional but recommended)

When calling for technical support, be prepared to provide the following information about your system to the Product Support engineer:

- Software and version being used
- Controller firmware version
- PC configuration
- A complete description of your hardware and operating systems, including:
  - jumper configuration
  - accessories installed (such as expansion daughter cards)
  - type of power supply
  - types of I/O units installed
  - third-party devices installed (e.g., barcode readers)
- Specific error messages seen

