# OPTOWARE USER'S GUIDE

Form 92-000107 — January, 2000

OPTO 22

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

**Product Support Services:**
800/TEK-OPTO (835-6786) or 951-695-3080
Fax: 951-695-3017
E-mail: support@opto22.com
web: support.opto22.com

**OptoWare User's Guide**
**Form 92-000107 — January, 2000**



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

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

---

# TABLE OF CONTENTS

# 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 4,095 I/O points can be accessed over a distance of several thousand feet at moderate speeds with high reliability.

The OptoWare drivers provide access to Optomux I/O. There are drivers provided for DOS, 16-bit Windows, and 32 bit Windows. "C" code for a "generic" Optomux driver is also provided to aid in implementing a driver on other platforms. Example programs and projects are provided in several languages on several platforms.

This manual covers all Optomux drivers. The appenices describe any platform specific information since driver function names and function parameters may differ slightly on different platforms.

With this manual, the drivers and the examples provided have all the necessary information to access Optomux I/O from your application.

## ABOUT THIS MANUAL

This manual is organized as follows:

- **Chapter 1: Introduction** — OptoWare introduction and overview.
- **Chapter 2: Quick Reference** — Index of commands.
- **Chapter 3: Command Description** — Description of each command.
- **Appendix A: Using the OptoWare Driver With Interpretive Basic** — DOS Example explaining the need for the BLOAD statement.
- **Appendix B: Sample Application Program** — An interpretive Basic program using OptoWare to do process control.
- **Appendix C: Using Utility Programs** — Describes utilities provided and their purpose.
- **Appendix D: Using the Optomux DLL in 16-bit Windows** — Describes how to use the Optomux DLL in Win16.
- **Appendix E: Using Optomux in 32-bit Windows** — Describes how to access Opto I/O in Win32.

# 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 plus signs 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

OptoWare is a software driver package designed to communicate to Optomux I/O units.

The OptoWare driver provides an interface between an Optomux network and application programs written in high-level languages. OptoWare allows communication to the Optomux network by simply calling a subroutine. The OptoWare driver performs the following functions:

- Builds and transmits Optomux command messages
- Carries out all the necessary handshaking
- Converts the data returned by Optomux to a form that is easily manipulated in high level languages
- Does extensive error checking and returns diagnostic error codes.

Using OptoWare saves much of the time and effort that would be spent becoming familiar with the intricacies of the Optomux serial protocol and message structure.

To use the OptoWare driver in an application program, one must know the following:

- How to call a subroutine from the application program
- How to tell the OptoWare driver what Optomux command to send by assigning values to parameters
- How to interpret the data passed back by the OptoWare driver.

# OPTOMUX

**Optomux is a communication protocol** used to communicate with a family of intelligent analog and digital I/O (input/output) devices. These distributed I/O units communicate with a host computer over an RS-485 serial data link. Each Optomux I/O unit has a unique address that allows the host computer to communicate with up to 256 different Optomux I/O units on a single serial communication link (address range: 0 to 255).

An Optomux I/O unit is composed of two parts, a brain board, and an I/O mounting rack. Each Optomux brain board contains a microprocessor that supports communication with the host computer and the I/O modules. The I/O mounting racks come in four-point, eight-point, and 16-point configurations. Input and output modules can be mixed in any combination on a given Optomux I/O unit.

The microprocessor provides each Optomux I/O unit with local intelligence that relieves the host computer from handling many time-consuming control functions such as latching, time delays, event counting, analog input averaging, etc.

The Optomux I/O units act as slave devices to the host computer. The host computer instructs an Optomux I/O unit to perform tasks or return requested data by issuing commands to that Optomux I/O unit.

# OPTOMUX SERIAL PROTOCOL

Optomux is capable of carrying out two types of protocol with the host computer: Two-pass protocol and four-pass protocol. These two protocol types are briefly described below. The OptoWare driver is capable of using either protocol. A general understanding of the two-pass and four-pass protocol is required to determine which type to use based on speed requirement, level of security, etc.

## Two–Pass Protocol

The two-pass protocol requires two distinct messages between the host computer and the specified Optomux I/O unit. The host computer initiates the transaction by sending a command to one of the 256 possible Optomux I/O units in the network. The addressed Optomux verifies that the command is valid and then executes the command. Upon command completion, Optomux returns an acknowledgment message along with any requested data. If an error is detected at any time, Optomux will not execute the command and will return a message to the host indicating the nature of the error.

Summary of two-pass protocol transaction:

- Command from host to Optomux
- Acknowledge command (and data, if applicable) from Optomux to host

  or

- Error message from Optomux to host.

For most applications, the two-pass protocol is recommended because it is more efficient. Message integrity is maintained due to checksum verification of command messages.

## Four-Pass Protocol

The four-pass protocol requires four distinct messages between the host computer and the specified Optomux I/O unit. The host computer initiates the transaction by sending a command to one of the 256 possible Optomux I/O units in a network. If the command is received correctly, the addressed Optomux I/O unit echoes the entire message back to the host computer. The host computer then transmits the execute command (E) to Optomux, which executes the command and returns an acknowledgment along with any necessary data upon command completion. If an addressed Optomux I/O unit detects an error, an error message is returned to the host and the protocol is terminated.

Summary of four-pass protocol transaction:

1. Command from host to Optomux
2. Command echoed back to host from Optomux
3. Execute command sent from host to Optomux
4. Acknowledge command (and data, if applicable) from Optomux to host.

# GETTING STARTED

Before proceeding with this section, please make backup copies of the OptoWare driver diskettes for your safety.

For users of LC2 or LC4 Local Controllers, the OptoWare driver software is built into the Opto 22 BASIC and FORTH languages.

The OptoWare driver subroutine allows high-level languages to communicate with Optomux by calling a single subroutine. The task of communicating to the analog and digital I/O is reduced to understanding how to call the OptoWare driver.

Most high-level languages provide a mechanism for calling assembly language subroutines. Usually this is done by including a CALL statement in the program. When the program is executed, the CALL statement causes the processor to begin execution of the subroutine referenced in the CALL statement. When the subroutine is finished, the processor continues execution of the program at the statement following the CALL statement.

For subroutine use to be more than a means of merely avoiding duplicate statements in a program, there must also be some mechanism to pass information between the program and the subroutine. For example, an assembly language subroutine named FOO calculates 2 + 9 * 5 and displays the results. If this operation is required at many different spots in the program, simply include the statement CALL FOO everywhere it is needed instead of repeatedly including the statements that would calculate and display the results.

Now assume that a value of A + B * C needs to be calculated and the result displayed in different parts of the program. If the values A, B, and C could be passed to the subroutine, then this would be very useful. Data is passed from the program to the subroutine by including a list of parameters in the CALL statement. The parameter list is simply a list of the variables from the program that contain the values used by the subroutine. This technique also provides the mechanism for retrieving data from the assembly language subroutine. The subroutine changes the value of one or more of the variables included in the parameter list. This is how the OptoWare driver interfaces with the Optomux network.

# OPTOWARE DRIVER PARAMETERS

The OptoWare driver subroutine requires the six parameters listed below to be passed every time it is called. These six parameters are used to provide the OptoWare driver with all the data needed to specify what command is sent to which Optomux I/O unit. Data values and error codes are also returned to the application program in the parameters.

The following is an example of a BASIC CALL statement to the Optomux network program written in Microsoft BASIC or Opto 22 BASIC.

CALL OptoWare [ERRORS%,ADDR%,CMD%,POSITIONS%(0),MODIFIERS%(0),INFO%(0)]

## Parameter List

The parameter list contains the names of the six variables to be passed to the OptoWare driver. All parameters passed to the driver must be 16-bit integers. The variable names in the parameter list above end with a percentage sign (%). The percentage sign that appears after a variable name specifies that the variable is a 16-bit integer in Microsoft BASIC and Opto 22 BASIC.

The parameters from left to right are described as follows:

### First Parameter: ERRORS

The ERRORS parameter is an integer variable used to return error codes. A value of zero in this parameter indicates that no errors were encountered and the command has been successfully executed. A value less than zero indicates that the driver has detected an error and the command has not been executed. The value indicates what type of error was encountered. For example, -29 indicates a turnaround time-out error, which means the driver sent a command to Optomux, but Optomux did not answer.

### Second Parameter: ADDRESS

The ADDRESS parameter is an integer variable that contains the Optomux I/O unit address of the intended Optomux (0-255 decimal).

### Third Parameter: COMMAND

The COMMAND parameter is an integer variable that contains the number of the desired command.

### Fourth Parameter: POSITIONS ARRAY

The POSITIONS array parameter is a 16-element integer array used to specify which module positions are to be affected by the command. Each analog or digital Optomux I/O unit can hold either four, eight, or 16 I/O modules. These modules are placed in locations on the Optomux board that are numbered from 0 to 3 (four-point Optomux), 0 to 7 (eight-point Optomux), or 0 to 15 (16-point Optomux). Position numbers are entered in consecutive array elements starting with the first element. The array element following the last position number must be set to a negative one (-1) to inform the driver there are no more module positions contained in the array. If all 16 positions are entered in the array, the negative one (-1) is not required since there is no array element following the last position number.

***Note:*** *There is no relationship between the array element number and the module position.*

## Example

To specify module positions 0, 2, 5, 6, and 15 to be affected by a particular command, the POSITIONS array parameter will contain the following values:

POSITIONS%(0) = 0
POSITIONS%(1) = 2
POSITIONS%(2) = 5
POSITIONS%(3) = 6
POSITIONS%(4) = 15
POSITIONS%(5) = -1

### Fifth Parameter: MODIFIERS ARRAY

The MODIFIERS array parameter is a two-element integer array used to hold the command modifiers. Some commands have several different modes of execution. For example, the Set Time Delay command allows the user to select different types of time delay. This information is contained in the MODIFIERS array.

### Sixth Parameter: INFO ARRAY

The INFO array parameter is a 16-element integer array that is used to hold data values necessary to execute the command. It is also used to hold all returned data. There are two types of data returned by the driver:

1. Data values that are not associated with a specific module position. For example, the Identify Optomux Type command returns a single value that is not associated with a module position. This type of data is returned in the first element of the INFO array parameter.

2. Data values that are associated with specific module positions. For example, the Read On/Off Status command returns 16 values indicating the status of the 16-module positions on the Optomux. These values are returned in the INFO array elements that correspond to the particular module position. The value for module position 0 is returned in the first element of the data array, the value for module position 4 is returned in the fifth element of the array, and so on.

***Note:*** *The module positions are numbered from 0 to 15.*

## Example

A command was sent to read the counters for module positions 0, 3, and 6 on a particular Optomux I/O unit. The values of the counts will be returned in array elements: INFO%(0), INFO%(3), and INFO%(6), respectively.

# ERROR CODES

There are two types of errors returned by the OptoWare driver: 1) Optomux detected errors and 2) OptoWare driver detected errors. The following describes the errors returned by Optomux and the OptoWare driver.

## Optomux Detected Errors

| ERROR NUMBER | ERROR DESCRIPTION |
|---|---|
| -1 | **Power Up Clear Expected**<br>This error indicates that the Optomux power has been cycled, or for some reason the Optomux processor has been reset. |

***Note:*** *If this message is received, it means that Optomux has gone through its power up sequence and has reset all values of that Optomux I/O unit to the default values. It will be necessary to reinitialize the Optomux I/O unit. The error flag on the I/O unit is automatically cleared when this error is returned.*

| | |
|---|---|
| -2 | **Undefined Command**<br>This error indicates that an invalid command has been sent to the Optomux. Possibly an analog command was sent to a digital board. |
| -3 | **Checksum Error**<br>This error indicates that the checksum received by the Optomux did not match the checksum calculated by the Optomux. |
| -4 | **Input Buffer Overrun**<br>The message received by the Optomux was too large for the Optomux to handle. It was probably caused by sending an analog command to a digital Optomux. |
| -5 | **Non-printable ASCII Character Received**<br>Only characters from 21 Hex to 7F Hex are valid command characters. |
| -6 | **Data Field Error**<br>Not enough data characters were received by Optomux for the specified bitmask. |
| -7 | **Serial Watchdog Time-out**<br>This error indicates that there was no serial communication for the time specified by the Watchdog commands. |
| -8 | **Invalid Limits Set**<br>The data field(s) passed to the Optomux contained values too large for the specified command. |

## OptoWare Driver Detected Errors

| ERROR NUMBER | ERROR DESCRIPTION |
|---|---|
| -20 | **Invalid Command Number**<br>The passed command number is not a valid Optomux or OptoWare driver command. |
| -21 | **Invalid Module Position**<br>The module number specified was larger than 15. |
| -22 | **Data Range Error**<br>The specified data was out-of-range for the command. |
| -23 | **Invalid First Modifier**<br>The first modifier number was out of range for the specified command. |
| -24 | **Invalid Second Modifier**<br>The second modifier number was out of range for the specified command. |
| -25 | **Invalid Address**<br>The passed address was not between 0 and 255. |
| -27 | **Not Enough Return Data**<br>There was not enough data received from the Optomux. It was probably caused by sending an analog command to a digital board or a digital command to an analog board. |
| -28 | **Invalid Return Data**<br>The data received from the Optomux I/O does not match what is required by the specified command. It was probably caused by sending an analog command to a digital board or a digital command to an analog board. This error also occurs if the driver is set for two-pass protocol but Optomux is in four-pass protocol. |
| -29 | **Turn Around Time Out**<br>The Optomux I/O did not respond to the command. Either the Optomux I/O unit is not online or a wrong address was specified. |
| -30 | **Input Buffer Overrun**<br>The message received by the OptoWare driver was longer than the driver's input buffer. |
| -31 | **Checksum Error**<br>The checksum of the message received by the OptoWare driver does not match the checksum calculated by the driver. |
| -33 | **Send Error**<br>This error indicates that the message cannot be sent by the OptoWare driver; either the turnaround time is too short for the processor used, or the CTS line on the serial port is not enabled when using a driver that uses flow control. |
| -34 | **Incorrect Command Echo In Four-Pass**<br>The command echo does not match the command. The addressed Optomux is probably in the two-pass mode. |

# QUICK REFERENCE

## ANALOG COMMANDS

| SYSTEM COMMANDS | COMMAND NUMBERS |
|---|---|
| Power Up Clear | 0 |
| Reset | 1 |
| Set Turnaround Delay | 2 |
| Set Analog Watchdog Delay | 45 |
| Set Analog Watchdog Time-out | 78 |
| Set Optomux Protocol | 4 |
| Identify Optomux Type | 5 |
| **CONFIGURE COMMANDS** | |
| Configure Positions | 6 |
| Configure As Inputs | 7 |
| Configure As Outputs | 8 |
| Set Temperature Probe Type | 76 |
| **READ AND WRITE COMMANDS** | |
| Write Analog Outputs | 35 |
| Read Analog Outputs | 36 |
| Read Analog Inputs | 37 |
| Average And Read Input | 38 |
| Update Analog Outputs | 46 |
| Start Averaging Inputs | 47 |
| Read Average Complete Bits | 48 |
| Read Averaged Inputs | 49 |
| Read Configuration | 70 |
| Read Temperature Inputs | 77 |
| Read Average Temperature Inputs | 79 |
| **INPUT RANGE COMMANDS** | |
| Set Input Range | 39 |
| Read Out-Of-Range Latches | 40 |
| Read And Clear Out-Of-Range Latches | 41 |
| Clear Out-Of-Range Latches | 42 |
| Read Lowest Values | 58 |

| INPUT RANGE COMMANDS | COMMAND NUMBERS |
|---|---|
| Read And Clear Lowest Values | 60 |
| Clear Lowest Values | 59 |
| Read Peak Values | 61 |
| Read And Clear Peak Values | 63 |
| Clear Peak Values | 62 |

| GAIN AND OFFSET COMMANDS | |
|---|---|
| Calculate Input Offsets | 52 |
| Set Input Offsets | 53 |
| Calculate And Set Input Offsets | 54 |
| Calculate Gain Coefficients | 55 |
| Set Gain Coefficients | 56 |
| Calculate And Set Gain Coefficients | 57 |

| WAVEFORM COMMANDS | |
|---|---|
| Set Output Waveform | 43 |
| Turn Off Existing Waveforms | 44 |
| Enhanced Output Waveform | 50 |
| Cancel Enhanced Waveforms | 51 |

# DIGITAL COMMANDS

| SYSTEM COMMANDS | COMMAND NUMBERS |
|---|---|
| Power Up Clear | 0 |
| Reset | 1 |
| Set Turnaround Delay | 2 |
| Set Digital Watchdog Delay | 3 |
| Set Enhanced Digital Watchdog Time-out | 71 |
| Set Optomux Protocol | 4 |
| Identify Optomux Type | 5 |
| Set Timer Resolution | 75 |

| CONFIGURE COMMANDS | |
|---|---|
| Configure Positions | 6 |
| Configure As Inputs | 7 |
| Configure As Outputs | 8 |

| READ AND WRITE COMMANDS | COMMAND NUMBERS |
|---|---|
| Write Digital Outputs | 9 |
| Write Binary Outputs | 65 |
| Activate Digital Outputs | 10 |
| Deactivate Digital Outputs | 11 |
| Read On/Off Status | 12 |
| Read Binary On/Off Status | 64 |
| Read Configuration | 70 |

| LATCH COMMANDS | |
|---|---|
| Set Latch Edges | 13 |
| Set Off-To-On Latches | 14 |
| Set On-To-Off Latches | 15 |
| Read Latches | 16 |
| Read And Clear Latches | 17 |
| Clear Latches | 18 |
| Read Binary Latches | 66 |
| Read And Clear Binary Latches | 67 |

| COUNTING COMMANDS | |
|---|---|
| Start And Stop Counters | 19 |
| Start Counters | 20 |
| Stop Counters | 21 |
| Read Counters | 22 |
| Read And Clear Counters | 23 |
| Clear Counters | 24 |

| TIME DELAY AND PULSE COMMANDS | |
|---|---|
| Set Time Delay | 25 |
| Initiate Square Wave | 26 |
| Turn Off Time Delay/Square Wave | 27 |
| High Resolution Square Wave | 68 |
| Retrigger Time Delay | 69 |
| Generate N Pulses | 72 |
| Start On Pulse | 73 |
| Start Off Pulse | 74 |

| DURATION MEASUREMENT COMMANDS | |
|---|---|
| Set Pulse Trigger Polarity | 28 |
| Trigger On Positive Pulse | 29 |
| Trigger On Negative Pulse | 30 |
| Read Pulse Complete Bits | 31 |
| Read Pulse Duration Counters | 32 |
| Read And Clear Duration Counters | 33 |
| Clear Duration Counters | 34 |

# DRIVER COMMANDS

| DRIVER COMMANDS | COMMAND NUMBERS |
|---|---|
| Set Driver Protocol | 100 |
| Set Turnaround Delay | 101 |
| Set Serial Port Number | 102 |
| Set Number Of Retries | 103 |
| Configure Serial Port | 104 |

# ERROR CODES

### OPTOMUX DETECTED ERRORS:

-1 Power Up Clear Expected
-2 Undefined Command
-3 Checksum Error
-4 Input Buffer Overrun
-5 Non-printable ASCII Character Received
-6 Data Field Error
-7 Serial Watchdog Time-out
-8 Invalid Limit Set

### OPTOWARE DRIVER DETECTED ERRORS:

-20 Invalid Command Number
-21 Invalid Module Position
-22 Data Range Error
-23 Invalid First Modifier
-24 Invalid Second Modifier
-25 Invalid Address
-27 Not Enough Return Data
-28 Invalid Return Data
-29 Turnaround Time Out (Optomux did not respond within the specified time interval)
-30 Input Buffer Overrun
-31 Checksum Error
-33 Send Error (Message cannot be sent out; probable serial port problem)
-34 Incorrect Command Echo In Four-Pass

# COMMAND DESCRIPTIONS

Descriptions of all the Analog and Digital Optomux commands are included in this section. The previous chapter contains a list of all the commands. The following list may be useful as a quick reference. The rest of this chapter describes each command in detail.

The format of the command descriptions is as follows:

## SAMPLE COMMAND #

### PURPOSE

Briefly describes what the command does.

### VERSIONS

Indicates which type of Optomux brain boards allow the command. For example, if you look at any particular command in this chapter, you will see either "Analog," "Digital," or "Analog and Digital."

### PARAMETERS

Indicates which of the six parameters passed to the driver are relevant to the command. All six parameters must be passed to the driver every time it is called; but, depending upon the specific command, the values of only some of the parameters may be used to fully specify the command and the returning data.

### REMARKS

Describes in detail what the command does and how it should be used.

### EXAMPLE

Shows actual BASIC program segments that demonstrate the use of the command. This manual uses Microsoft BASIC as implemented on the IBM PC for its examples.

## POWER UP CLEAR 0

### PURPOSE

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

### VERSIONS

Analog and Digital.

### PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 0.

### 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

Sends a Power Up Clear command to the Optomux addressed as 107.

```
200     ADDR% = 107              'Optomux Address Is 107
210     CMD% = 0                 'Power Up Clear Command
220     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSI%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

# RESET 1

### PURPOSE

Resets Optomux to power up conditions.

### VERSIONS

Analog and Digital.

### PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*POSITIONS ARRAY:* Contains the value 1.

### REMARKS

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

| ANALOG | DIGITAL |
|---|---|
| Set all output modules to zero scale | Turn off all outputs modules |
| Configure all positions as inputs | Configure all positions as inputs |
| Set protocol based on jumper B10 | Set protocol based on jumper B10 |
| Disable watchdog timer | Disable watchdog timer |
| Set turnaround delay to zero | Set turnaround delay to zero |
| Set all offset to zero | Cancel counters and duration timers |
| Set all gain coefficients to one | Clear latches |
| Cancel all averaging | Set timer resolution to 10 ms |
| Cancel all temperature probe types | |

### EXAMPLE

This example sends RESET commands to Optomux units at addresses 0 through 25.

```
300     CMD% = 1                    'Reset Command
310     FOR ADDR% = 0 TO 25         'Set Up Loop To Increment Optomux
        Address
320     GOSUB 1000                  'Call The Driver
330     NEXT

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

## SET TURNAROUND DELAY 2

### PURPOSE

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

### VERSIONS

Analog and Digital.

### PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 2.

*INFO ARRAY:* The first element of the INFO array parameter contains a value specifying the desired delay.

*LENGTH:*

| VALUE | DELAY |
|---|---|
| 0 | No Delay |
| 1 | 10 msec |
| 2 | 100 msec |
| 3 | 500 msec |

### EXAMPLE

This example sets the Optomux I/O units at addresses 0 through 240 to wait for 100 ms before responding to the host.

```
100     CMD% = 2                  'Set Turnaround Delay Command
110     INFO%(0) = 2              'Specify 100 ms Delay
120     FOR ADDR% = 0 TO 240      'Set Up Loop To Increment Address
130     GOSUB 1000                'Call The Driver
140     NEXT                      'Go To Next Address

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET DIGITAL WATCHDOG 3

## 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.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 3.

*INFO ARRAY:* The first element of the INFO array parameter contains a value from 0 through 7, which specifies desired action and time interval. Valid actions and times are:

| Value | Action | Time |
|---|---|---|
| 0 | Watchdog disabled | Not applicable |
| 1 | Turn all outputs OFF | 10 seconds |
| 2 | Turn all outputs OFF | 1 minute |
| 3 | Turn all outputs OFF | 10 minutes |
| 4 | Watchdog disabled | Not applicable |
| 5 | Turn output 0 ON, all others OFF | 10 seconds |
| 6 | Turn output 0 ON, all others OFF | 1 minute |
| 7 | Turn output 0 ON, all others OFF | 10 minutes |

## EXAMPLE

This example instructs the Optomux at address 92 (hex) to turn off all outputs if no activity is detected on the Optomux network for one minute.

```
100     ADDR% = 92                'Optomux Address Is 92
110     CMD% = 3                  'Digital Watchdog Delay Command
130     INFO%(0) = 2              'Specify One Minute Delay All Outputs Off
140     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET OPTOMUX PROTOCOL 4

## PURPOSE

Instructs Optomux to use either two-pass or four-pass protocol.

## VERSIONS

Analog and Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 4.

*INFO ARRAY:* The first element of the INFO array parameter contains a value of 0 or 1 and selects the protocol:

| VALUE | TYPE |
|---|---|
| 0 | Two-pass Protocol |
| 1 | Four-pass Protocol |

## REMARKS

This command is used to change the protocol that Optomux is using, it is necessary to use the Set Driver Protocol command (command 100) to set the driver to the same protocol. The driver default is two-pass protocol.

## EXAMPLE

This example instructs the Optomux at address 0 to use four-pass protocol and then sets the driver to four-pass protocol.

```
100    ADDR% = 0                'Optomux Address Is 0
110    CMD% = 4                 'Set Optomux Protocol Command
120    INFO%(0) = 1             'Select Four-pass Protocol
130    GOSUB 1000               'Call The Driver
140    CMD% = 100               'Set The Driver To Four-pass
150    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# IDENTIFY OPTOMUX TYPE 5

## PURPOSE

Instructs Optomux to identify itself as either an Analog or Digital I/O unit.

## VERSIONS

Analog and Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 5.

*INFO ARRAY:* A value will be returned in the first element of the INFO array parameter indicating the type of Optomux which is at the specified address*:*

| VALUE | TYPE |
|---|---|
| 0 | Digital Optomux |
| 1 | Analog Optomux |

## EXAMPLE

This example sends Identify Optomux commands to addresses 0 through 220 and displays the type of Optomux found at each address.

```
100     CMD% = 5                  'Identify Optomux Command
110     FOR ADDR% = 0 TO 220      'Set Up Loop To Increment The Address
120     GOSUB 1000                'Call The Driver
130     '
140     '                         Display Message Indicating What
                                  Type Of Optomux
150     '
160     IF INFO%(0) = 0 THEN PRINT ADDR%;" is a digital rack"
170     IF INFO%(0) = 1 THEN PRINT ADDR%;" is a analog rack"
180     NEXT

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < THEN GOSUB 200 'Test For Errors
1020    RETURN
```

# CONFIGURE POSITIONS 6

## PURPOSE

Configures the positions to function as either inputs or outputs.

## VERSIONS

Analog and Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 6.

*POSITIONS ARRAY:* Contains the positions that are to function as outputs. Module positions not specified in the POSITIONS array will be configured to function as inputs.

## REMARKS

If the configuration for any position is changed by this command, then any time delay, latch, etc. is cleared. On power up, all positions are configured as inputs.

## EXAMPLE

This example configures module positions 2, 3, and 0 to function as outputs on the Optomux at address 100, all other positions will be configured as inputs.

```
100    MACHINE1% = 100          'Set Up Variable For Address
110    FAN% = 2                 'Fan Is Position 2
120    COMPRESSOR% = 3          'Compressor Is Position 3
130    ALARM% = 0               'Alarm Is Position 0

500    ADDR% = MACHINE1%        'Optomux Address Is 100
510    CMD% = 6                 'Select Configure Command
520    POSITIONS%(0) = FAN%     'Fan Is An Output (Position 2)
530    POSITIONS%(1) = COMPRESSOR%'Compressor Is An Output (Position 3)
540    POSITIONS%(2) = ALARM%   'Alarm Is An Output (Position 0)
550    POSITIONS%(3) = -1       'End Of List
560    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# CONFIGURE AS INPUTS 7

## PURPOSE

Configures the positions to function as inputs.

## VERSIONS

Analog and Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 7.

*POSITIONS ARRAY:* Contains the positions that are to function as inputs. All other positions not specified in the POSITIONS array will be left unchanged.

## REMARKS

If the configuration for any position is changed by this command, then any time delay, latch, etc. is cleared. On power up, all positions are configured as inputs.

## EXAMPLE

This example configures module positions 4, 11, and 15 to function as inputs on the Optomux at address 100.

```
100    MACHINE1% = 100          'Optomux Address Of Machine 1 Is 100
110    SWITCH1% = 15            'Position 15 Is Switch 1
120    SWITCH2% = 11            'Position 11 Is Switch 2
130    COUNT1% = 4              'Position 4 Is Count Input

500    ADDR% = MACHINE1%        'Optomux Address Is 100
510    CMD% = 7                 'Configure As Inputs Command
520    POSITIONS%(0) = SWITCH1% 'Specify Position 15 Is An Input
530    POSITIONS%(1) = SWITCH2% 'Specify Position 11 Is An Input
540    POSITIONS%(2) = COUNT1%  'Specify Position 4 Is An Input
550    POSITIONS%(3) = -1       'End Of List
560    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# CONFIGURE AS OUTPUTS 8

## PURPOSE

Configures the positions to function as outputs.

## VERSIONS

Analog and Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 8.

*POSITIONS ARRAY:* Contains the positions that are to function as outputs. All other positions not specified in the POSITIONS array will be left unchanged.

## REMARKS

If the configuration for any position is changed by this command, then any time delay, latch, etc. is cleared. On power up, all positions are configured to function as inputs.

## EXAMPLE

This example configures module positions 0, 2, 3, and 4 to function as outputs on the Optomux at address 160.

```
100    CONVEYOR%=160            'Optomux Address Of Conveyor Is 160
110    MOTOR1% = 2              'Motor #1 Is Position 2
120    MOTOR2% = 3              'Motor #2 Is Position 3
130    MOTOR3% = 4              'Motor #3 Is Position 4
140    ALARM% = 0               'Alarm Is Position 0

500    ADDR% = CONVEYOR%        'Optomux Address Is Equal To CONVEYOR
510    CMD% = 8                 'Configure As Outputs Command
520    POSITIONS%(0) = MOTOR1%
530    POSITIONS%(1) = MOTOR2%  'Put Positions To Be Outputs
540    POSITIONS%(2) = MOTOR3%  'In The POSITIONS Array
550    POSITIONS%(3) = ALARM%
560    POSITIONS%(4) = -1       'End Of List
570    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% <0 THEN GOSUB 15000                'Test For Errors
1020   RETURN
```

# WRITE DIGITAL OUTPUTS 9

## PURPOSE

Turns output modules ON and OFF.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 9.

*POSITIONS ARRAY:* Contains the output positions that are to be turned ON. All other outputs will be turned OFF.

## REMARKS

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.

## EXAMPLE

This example turns ON the output modules in positions 2 and 3 on the Optomux at address 100. All other output positions will be turned OFF.

```
500     ADDR% = 100              'Optomux Address Is 100
510     CMD% = 9                 'Write Outputs Command
520     POSITIONS%(0) = 2        'Specify Position 2
530     POSITIONS%(1) = 3        'Specify Position 3
540     POSITIONS%(2) = -1       'End Of List
550     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020    RETURN
```

# ACTIVATE DIGITAL OUTPUTS 10

## PURPOSE

Turns ON the specified outputs.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 10. |
| *POSITIONS ARRAY:* | Contains a list of the output positions that are to be turned ON. All other outputs will remain unchanged. |

## REMARKS

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.

## EXAMPLE

This example turns ON the output modules in positions 2 and 3 on the Optomux at address 160. All other outputs will remain unchanged.

```
100     CONVEYOR% = 160
110     MOTOR1% = 2
120     MOTOR2% = 3

500     ADDR% = CONVEYOR%          'Set Optomux Address To CONVEYOR%
510     CMD% = 10                  'Activate Outputs Command
520     POSITIONS%(0) = MOTOR1%    'Specify Modules To Activate
530     POSITIONS%(1) = MOTOR2%
540     POSITIONS%(2) = -1         'End Of List
550     GOSUB 1000                 'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020    RETURN
```

# DEACTIVATE DIGITAL OUTPUTS 11

## PURPOSE

Turns OFF output modules.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 11.

*POSITIONS ARRAY:* Contains the output positions that are to be turned OFF. All other positions will be left unchanged.

## REMARKS

Module positions that have been configured as inputs are not affected by this command.

## EXAMPLE

This example turns OFF the output modules in positions 2 and 3 on the Optomux at address 160. All other output positions will be left unchanged.

```
500     ADDR% = 160              'Optomux Address Is 160
510     CMD% = 11                'Deactivate Relays Command
520     POSITIONS%(0) = 2        'Specify Position 2
530     POSITIONS%(1) = 3        'Specify Position 3
540     POSITIONS%(2) = -1       'End Of List
550     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020    RETURN
```

# READ ON/OFF STATUS 12

## PURPOSE

Returns the ON/OFF status of all module positions.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 12.

*INFO ARRAY:* The return data will be passed back in the INFO array parameter. Each element of the INFO array will contain a 1 or a 0 indicating the ON/OFF (1 = ON, 0 = OFF) status of one module position. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain a 1 or a 0, indicating the status of module position 2. Remember, module positions are numbered 0 through 15.

## EXAMPLE

This example sends a Read ON/OFF Status command to the Optomux at address 0 and displays a message indicating the status of each of the 16 module positions.

```
110     ADDR% = 0                 'Optomux Address Is 0
120     CMD% = 12                 'Read ON/OFF Status Command
130     GOSUB 1000                'Call The Driver
140     FOR I% = 0 TO 15          'Display Status Of All 15 Positions
150     IF INFO% (I%) = 0 THEN GO TO 180'1 = ON, 0 = OFF
160     PRINT "Module position ";I%;" is ON"
170     GO TO 190
180     PRINT "Module position ";I%;" is OFF"
190     NEXT

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET LATCH EDGES 13

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 13.

*POSITIONS ARRAY:* The positions array contains the module positions that are to latch on ON-to-OFF transitions. All other positions that have been configured as inputs will be set to latch on OFF-to-ON transitions.

## REMARKS

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

## EXAMPLE

This example instructs the Optomux at address 125 to set positions 2 and 14 to latch on ON-to-OFF transitions, all other positions are set to latch on OFF-to-ON transitions.

```
100    ADDR% = 125              'Optomux Address Is 125
110    CMD% = 13                'Set Latch Edges Command
120    POSITIONS%(0) = 2        'Specify Position 2
130    POSITIONS%(1) = 14       'Specify Position 14
140    POSITIONS%(2) = -1       'End Of List
150    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# SET OFF-TO-ON LATCHES 14

## PURPOSE

Sets positions to latch on OFF-to-ON transitions.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 14. |
| *POSITIONS ARRAY:* | Contains the positions that are to latch on OFF-to-ON transitions. All other positions are left unchanged. |

## REMARKS

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

## EXAMPLE

This example instructs the Optomux at address 125 to set positions 0, 1, 2, 3, 4, 5, 6, and 7 to latch on OFF-to-ON transitions, all other positions are left unchanged.

```
500     ADDR% = 125              'Optomux Address Is 125
510     CMD% = 14                'Set OFF-to-ON Latches Command
520     FOR I% = 0 TO 7          'Specify Module Positions 0 Through 7
530     POSITIONS% (I%) = I%     'In The POSITIONS Array
540     NEXT
550     POSITIONS% (I%) = -1     'End Of List
560     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

# SET ON-TO-OFF LATCHES 15

## PURPOSE

Sets positions to latch on ON-to-OFF transitions.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 15.

*POSITIONS ARRAY:* Contains the positions that are to latch on ON-to-OFF transitions. All other positions are left unchanged.

## REMARKS

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

## EXAMPLE

This example instructs the Optomux at address 125 to set positions 2 and 14 to latch on ON-to-OFF transitions, all other positions are left unchanged.

```
500     ADDR% = 125              'Optomux Address Is 125
510     CMD% = 15                'Set ON-to-OFF Latches Command
520     POSITIONS%(0) = 2        'Specify Position 2
530     POSITIONS%(1) = 14       'Specify Position 14
540     POSITIONS%(2) = -1       'End Of List
550     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# READ LATCHES 16

## PURPOSE

Returns data indicating which of the inputs have latched.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 16.

*INFO ARRAY:* Return data will be passed back in the INFO array parameter. Each element of the INFO array will contain a 1 or a 0 indicating whether or not one of the 16 module positions has latched (1 = latched, 0 = unlatched). The index into the INFO array corresponds to the module position. For example, the third element of the INFO array will contain a 1 or a 0 indicating the latched/unlatched status of module position 2. Remember, module positions are numbered 0 through 15.

## REMARKS

This command has no effect on the Optomux latches. Subsequent Read Latches commands will return consistent results.

## EXAMPLE

This example sends a Read Latches command to the Optomux at address 240 and displays a message indicating the latched/unlatched status of each of the 16 module positions.

```
100    ADDR% = 240              'Optomux Address Is 240
110    CMD% = 16                'Read Latches Command
120    GOSUB 1000               'Call The Driver
130    FOR I% = 0 TO 15         'Display Status Of All Latches
140    IF INFO% (I%) = 0 THEN GO TO 170'0 = Unlatched, 1 = Latched
150    PRINT "Position ";I%;" has latched"
160    GO TO 180
170    PRINT "Position ";I%;" has not latched"
180    NEXT

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# READ AND CLEAR LATCHES 17

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 17.

*POSITIONS ARRAY:* Contains module positions that are to have their latches cleared. All others will be unchanged.

*INFO ARRAY:* Return data will be passed back in the INFO array parameter. Each element of the INFO array will contain a 1 or a 0 indicating whether or not one of the 16 module positions has latched (1 = latched, 0 = unlatched). The index into the INFO array corresponds to the module position. For example, the third element of the INFO array will contain a 1 or a 0 indicating the latched/unlatched status of module position 2.

## EXAMPLE

This example sends a Read And Clear Latches command to the Optomux at address 77 for positions 4 and 5. Messages are displayed indicating the latched/unlatched status of each of the 16 module positions.

```
100    ADDR% = 77               'Optomux Address Is 77
110    CMD% = 17                'Read And Clear Latches Command
120    POSITIONS%(0) = 4        'Specify Position 4
130    POSITIONS%(1) = 5        'Specify Position 5
140    POSITIONS%(2) = -1       'End Of List
150    GOSUB 1000               'Call The Driver
160    FOR I% = 0 TO 15         'Display Status Of All Latches
170    IF INFO% (I%) = 0 THEN GO TO 200'0 = Unlatched, 1 = Latched
180    PRINT "Position ";I%;" has latched"
190    GO TO 210
200    PRINT "Position ";I%;" has not latched"
210    NEXT

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020   RETURN
```

# CLEAR LATCHES 18

## PURPOSE

Sets latches for specified positions to the unlatched state.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 18.

*POSITIONS ARRAY:* Contains the positions that are to have their latches cleared (set to unlatched). All other module positions will be left unchanged.

## EXAMPLE

This example clears the latches for modules in positions 2 and 3 on the Optomux at address 160. All other latches will be unchanged.

```
100    MACHINE1% = 160          'Optomux Address Is 160
110    SWITCH1% = 2             'Switch 1 Is At Position 2
120    SWITCH 2% = 3            'Switch 2 Is At Position 3

500    ADDR% = MACHINE1%
510    CMD% = 18                'Clear Latches Command
520    POSITIONS%(0) = SWITCH1% 'Specify Position SWITCH1%
530    POSITIONS%(1) = SWITCH2% 'Specify Position SWITCH2%
540    POSITIONS%(2) = -1       'End Of List
550    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020   RETURN
```

# START AND STOP COUNTERS 19

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 19. |
| *POSITIONS ARRAY:* | Contains the positions at which Optomux will start counting ON-to-OFF transitions. Optomux will stop counting ON-to-OFF transitions at all other positions. |

## REMARKS

This command has no effect on the stored count. Counting can start or resume at any time. The maximum count is 65,535. The counter then resets to 0. Frequencies up to 400 Hz (50% duty cycle) can be counted.

***Note:*** *Using the Generate N Pulses command (command 72) will degrade the maximum counting frequency to about 350 Hz.*

## EXAMPLE

This example instructs the Optomux at address 233 to start counting on positions 7 and 8. Counting will be stopped at all other positions.

```
500     ADDR% = 233              'Optomux Address Is 233
510     CMD% = 19                'Start And Stop Counters Command
520     POSITIONS%(0) = 7        'Specify Position 7
530     POSITIONS%(1) = 8        'Specify Position 8
540     POSITIONS%(2) = -1       'End Of List
550     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# START COUNTERS 20

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 20. |
| *POSITIONS ARRAY:* | Contains the positions at which Optomux will start/resume counting ON-to-OFF transitions. All other positions will not be affected. |

## REMARKS

This command has no effect on the stored count. Counting can start or resume at any time. The maximum count is 65,535. The counter then resets to 0. Frequencies up to 400 Hz (50% duty cycle) can be counted.

***Note:*** *Using the Generate N Pulses command (command 72) will degrade the maximum counting frequency to about 350 Hz.*

## EXAMPLE

This example instructs the Optomux at address 233 to start counting at positions 7 and 8.

```
100    MACHINE1% = 233
110    BADPARTS% = 7
120    GOODPARTS% = 8

500    ADDR% = MACHINE1%           'Optomux Address Is 233
510    CMD% = 20                   'Start Counting Command
520    POSITIONS%(0) = BADPARTS%   'Specify Position 7
530    POSITIONS%(1) = GOODPARTS%  'Specify Position 8
540    POSITIONS%(2) = -1          'End Of List
550    GOSUB 1000                  'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020   RETURN
```

# STOP COUNTERS 21

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 21.

*POSITIONS ARRAY:* Contains the positions at which Optomux will stop counting ON-to-OFF transitions. All other positions will not be affected.

## REMARKS

This command has no effect on the stored count, counting can resume at any time.

## EXAMPLE

This example instructs the Optomux at address 233 to stop counting at positions 7 and 8.

```
100     MACHINE1% = 233
110     BADPARTS% = 7
120     GOODPARTS% = 8

600     ADDR% = MACHINE1%           'Optomux Address Is 233
610     CMD% = 21                   'Stop Counting Command
620     POSITIONS%(0) = BADPARTS%   'Specify Position 0
630     POSITIONS%(1) = GOODPARTS%  'Specify Position 1
640     POSITIONS%(2) = -1          'End Of List
650     GOSUB 1000                  'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020    RETURN
```

# READ COUNTERS 22

## PURPOSE

Returns the counter values for the specified positions.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 22. |
| *POSITIONS ARRAY:* | Contains the module positions for which Optomux will return the counter values. |
| *INFO ARRAY:* | Counter values will be passed back in the INFO array parameter. These values are returned in the INFO array elements that correspond to the particular module positions; i.e., the counter value for module position 0 is returned in the first element of the INFO array, the value for module position 4 is returned in the fifth array element, etc. |

## REMARKS

This command has no effect on the stored count. Counting can start or resume at any time. The maximum count is 65,535. The counter then resets to 0. Frequencies up to 400 Hz (50% duty cycle) can be counted.

***Note:*** *Using the Generate N Pulses command (command 72) will degrade the maximum counting frequency to about 350 Hz.*

## EXAMPLE

This example sends a Read Counters command to the Optomux at address 75, requesting then displaying the counter values for positions 2 and 5.

```
100    ADDR% = 75                'Optomux Address is 75
110    CMD% = 22                 'Read Counters Command
120    POSITIONS%(0) = 2         'Specify Position 2
130    POSITIONS%(1) = 5         'Specify Position 5
140    POSITIONS%(2) = -1        'End Of List
150    GOSUB 1000                'Call The Driver
160    PRINT "Count at position #2 = ";INFO% (2)     'Display Count
170    PRINT "Count at position #5 = ";INFO% (5)     'Values

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000                  'Test For Errors
1020   RETURN
```

# READ AND CLEAR COUNTERS 23

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 23.

*POSITIONS ARRAY:* Contains the module positions for which Optomux will return and clear the counter values. All other positions will not be affected.

*INFO ARRAY:* Counter values will be passed back in the INFO array parameter. The values are returned in the INFO array elements that correspond to the particular module positions. The counter value for module position 0 is returned in the first element of the INFO array, the value for module position 1 is returned in the second array element, etc.

## EXAMPLE

This example reads and clears the counter values for positions 2 and 5 on the Optomux at address 75. The returned count values are displayed.

```
100     ADDR% = 75                  'Optomux Address Is 75
110     CMD% = 23                   'Read And Clear Counters Command
120     POSITIONS%(0) = 2           'Specify Position 2
130     POSITIONS%(1) = 5           'Specify Position 5
140     POSITIONS%(2) = -1          'End Of List
150     GOSUB 1000                  'Call The Driver
160     PRINT "Count at pos.
#2 = ";INFO% (2)                    'Display Counters Values
170     PRINT "Count at pos.
#5 = ";INFO%(5)

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020    RETURN
```

# CLEAR COUNTERS 24

## PURPOSE

Resets counters for specified module positions to zero.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 24. |
| *POSITIONS ARRAY:* | Contains the module positions where Optomux will clear the counters. All other positions will be unchanged. |

## EXAMPLE

This example instructs the Optomux at address 125 to clear the counters at positions 0 through 7. All other positions are left unchanged.

```
500     ADDR% = 125              'Optomux Address Is 125
510     CMD% = 24                'Clear Counters Command
520     FOR I% = 0 TO 7
530     POSITIONS% (I%) = I%     'Specify Positions 0 Through 7
540     NEXT
550     POSITIONS% (I%) = -1     'End Of List
560     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET TIME DELAY 25

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 25.

*POSITIONS ARRAY:* Contains the module positions that will be used in the selected delay or pulsed mode.

*MODIFIERS ARRAY:* The first element of the MODIFIERS array parameter contains a single value which selects the type of delay to be used. Valid delay types are:

| MODIFIERS%(0) | DELAY DESCRIPTION |
|---|---|
| 0 | On Pulse - When instructed to go from OFF to ON, turn ON for current timer resolution x [first element of INFO array] then turn OFF. |
| 1 | On Delay - When instructed to go from OFF to ON, stay OFF for current timer resolution x [first element of INFO array] then turn ON. |
| 2 | Off Pulse - When instructed to go from ON to OFF, turn OFF for current timer resolution x [first element of INFO array] then turn ON. |
| 3 | Off Delay - When instructed to go from ON to OFF, stay ON or current timer resolution x [first element of INFO array] then turn OFF. |

*INFO ARRAY:* The delay length is specified in the first element of the INFO array. The range of possible delay lengths is 0 to 65,535 (in units of the current timer resolution). A 0 value is equivalent to a delay length of 65,536. See Set Timer Resolution command (command 75).

### EXAMPLE

This example sets positions 2 and 3 on the Optomux at address 137 to turn on for 1.2 seconds and then turn off when Optomux is instructed to turn them on (assumed default timer resolution of 10 ms).

```
500     ADDR% = 137               'Optomux Address Is 137
510     CMD% = 25                 'Set Time Delay Command
520     POSITIONS%(0) = 2         'Specify Position 2
530     POSITIONS%(1) = 3         'Specify Position 3
540     POSITIONS%(2) = -1        'End Of List
550     MODI%(0) = 0              'Delay Type Is Zero "Pulse On"
560     INFO%(0) = 120            'Set Delay Length To 1.2 Seconds
570     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# INITIATE SQUARE WAVE 26

## PURPOSE

Initiates a continuous square wave at specified output positions.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 26.

*POSITIONS ARRAY:* Contains the positions that are to output continuous square waves. All other positions will not be affected.

*INFO ARRAY:* The first two elements of the INFO array contain values that define the on and off times of the square wave. The values can be between 0 and 255 (0 is equivalent to a delay value of 256). The on time is defined as the current timer resolution x 256 x [the first element]. The off time is defined as the current timer resolution x 256 x [the second element]. The maximum for on and off times is 2.56 seconds x 256 x 256 (2796.20 minutes or 46.6 hours).

## REMARKS

The square wave will continue until it is turned off using command number 27 or a time delay is set for that position. The Write Digital Outputs, Activate Digital Outputs, and Deactivate Digital Outputs commands will have no effect.

## EXAMPLE

This example instructs the Optomux at address 92 to initiate a continuous square wave at position 12 with an on time of 2.56 seconds and an off time of 5.12 seconds (assumed timer resolution of 10 ms).

```
100    ADDR% = 92               'Optomux Address Is 92
110    CMD% = 26                'Initiate Square Wave Command
120    POSITIONS%(0) = 12       'Specify Position 12
130    POSITIONS%(1) = -1       'End Of List
140    INFO%(0) = 1             'Set On Time Is 2.56 Seconds
150    INFO%(1) = 2             'Set Off Time Is 5.12 Seconds
160    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# TURN OFF TIME DELAY/SQUARE WAVE 27

## PURPOSE

Turns off existing Time Delay or Square Wave at specified positions.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 27.

*POSITIONS ARRAY:* Contains the positions at which Optomux will turn off Time Delays and Square Waves. All other positions will remain unchanged.

## EXAMPLE

This example instructs the Optomux at address 233 to turn off any existing time delays or square waves at positions 7 and 8.

```
600     ADDR% = 233              'Optomux Address Is 233
610     CMD% = 27                'Turn Off Time Delay/Waveform Command
620     POSITIONS%(0) = 7        'Specify Position 7
630     POSITIONS%(1) = 8        'Specify Position 8
640     POSITIONS%(2) = -1       'End Of List
650     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

## SET PULSE TRIGGER POLARITY 28

### PURPOSE

Instructs Optomux to measure ON and OFF pulses.

### VERSIONS

Digital.

### PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 28. |
| *POSITIONS ARRAY:* | Contains the input positions for which Optomux is to measure ON duration. All other input positions will be set to measure OFF duration. |

### REMARKS

Optomux will measure the duration of the first pulse of the appropriate level and store the result to be recalled. 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 "Pulse Complete Bits" can be read to determine whether Optomux has finished the measurement process.

The resolution for the duration counters is dependent upon the current timer resolution. The default value gives you a resolution of 10 ms which will allow you to measure a pulse of up to 10.92 minutes. If you have selected the lowest resolution, (2.56 seconds as opposed to 0.01 seconds) you could measure a pulse of up to 2,796.16 minutes (46.6 hours).

This command does not clear pre-existing duration counter values or Pulse Complete Bits. If a position's Pulse Complete Bit has been previously set, no measurements are made until that position's Pulse Complete Bit and duration counter are cleared by either the Clear Duration Counters or Read And Clear Duration Counters command.

### EXAMPLE

This example instructs the Optomux at address 100 to measure ON duration at input positions 2, 3, and 0, and to measure OFF duration at all other input positions.

```
100    MACHINE1% = 100          'Optomux Address For Machine #1 is 100
110    BUTTON1% = 2             'Button #1 Is At Module Position 2
120    BUTTON2% = 3             'Button #2 Is At Module Position 3
130    BUTTON3% = 0             'Button #3 Is At Module Position 0

500    ADDR% = MACHINE 1%       'Optomux Address Is 100
510    CMD% = 28                'Set Pulse Level Command
520    POSITIONS%(0) = BUTTON1% 'Specify Position 2
530    POSITIONS%(1) = BUTTON2% 'Specify Position 3
540    POSITIONS%(2) = BUTTON3% 'Specify Position 0
550    POSITIONS%(3) = -1       'End Of List
560    GOSUB 1000               'Call The Driver

1000   CALL OptoWare(ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# TRIGGER ON POSITIVE PULSE 29

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 29. |
| *POSITIONS ARRAY:* | Contains a list of the input positions for which Optomux is to measure ON duration. All other positions will be unchanged. |

## REMARKS

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 "Pulse Complete Bits" can be read to determine whether Optomux has finished the measurement process.

The resolution for the duration counters is dependent upon the current timer resolution. The default value gives you a 10 ms resolution which will allow you to measure a pulse of up to 10.92 minutes. If you have selected the lowest resolution, (2.56 seconds as opposed to 0.01 seconds) you could measure a pulse of up to 2,796.16 minutes (46.6 hours).

This command does not clear pre-existing duration counter values or Pulse Complete Bits. If a position's Pulse Complete Bit has been previously set, no measurements are made until that position's Pulse Complete Bit and duration counter are cleared by either the Clear Duration Counters or Read And Clear Duration Counters command.

## EXAMPLE

This example instructs the Optomux at address 100 to measure ON duration at input positions 5, 6, and 7. All other input positions will not be affected.

```
500    ADDR% = 100              'Optomux Address Is 100
510    CMD% = 29                'Measure ON Pulses Command
520    POSITIONS%(0) = 5        'Specify Position 5
530    POSITIONS%(1) = 6        'Specify Position 7
540    POSITIONS%(2) = 7        'Specify Position 8
550    POSITIONS%(3) = -1       'End Of List
560    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# TRIGGER ON NEGATIVE PULSE 30

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 30. |
| *POSITIONS ARRAY:* | Contains a list of the input positions for which Optomux is to measure OFF duration. All other input positions will be unchanged. |

## REMARKS

Optomux will measure the duration of the first pulse of the appropriate level and store the result to be recalled. 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 "Pulse Complete Bit" can be read to determine whether Optomux has finished the measurement process.

The resolution for the duration counters is dependent on the current timer resolution. The default value gives you a 10 ms resolution, which will allow you to measure a pulse of up to 10.92 minutes. If you have selected the lowest resolution, (2.56 seconds as opposed to 0.01 seconds) you could measure a pulse of up to 2,796.16 minutes (46.6 hours).

This command does not clear pre-existing duration counter values, or Pulse Complete Bits. If a position's Pulse Complete Bit has been previously set, no measurements are made until that position's Pulse Complete Bit and duration counter are cleared by either the Clear Duration Counters or Read And Clear Duration Counters command.

## EXAMPLE

This example instructs the Optomux at address 100 to measure OFF duration at input positions 5, 6, and 7. All other input positions will not be affected.

```
500     ADDR% = 100              'Optomux Address Is 100
510     CMD% = 30                'Measure OFF Pulses Command
520     POSITIONS%(0) = 5        'Specify Position 5
530     POSITIONS%(1) = 6        'Specify Position 6
540     POSITIONS%(2) = 7        'Specify Position 7
550     POSITIONS%(3) = -1       'End Of List
560     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

# READ PULSE COMPLETE BITS 31

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 31.

*INFO ARRAY:* The return information will be passed back in the INFO array parameter. Each element of the INFO array will contain a 1 or a 0 indicating whether or not Optomux has completed measuring a pulse at one of the 16 module positions. Module positions corresponding to array elements set to 1 have completed a pulse of the correct level. No further measurements can be made at these positions until the Pulse Complete Bits are reset by using either the Clear Duration Counters or Read And Clear Duration Counters command.

## EXAMPLE

This example sends a Read Pulse Complete bits command to the Optomux at address 2, requesting Optomux to return the status of all pulse duration counters. A message indicating whether or not each position has completed measuring a pulse is displayed. Assume that positions 0 through 7 have been configured to measure pulse durations.

```
100    ADDR% = 2              'Optomux Address Is 2
110    CMD% = 31              'Read Pulse Complete Bits Command
120    GOSUB 1000             'Call The Driver
130    FOR I% = 0 TO 7        'Print Status For Positions 0 Through 7
140    IF INFO%(I%) = 0 THEN GO TO 170
150    PRINT                  "Pulse measurement finished at position ";I%
160    GO TO 180
170    PRINT                  "Pulse measurement not done at position ";I%
180    NEXT

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# READ PULSE DURATION COUNTERS 32

## PURPOSE

Returns the pulse duration counter values for the specified positions.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 32. |
| *POSITIONS ARRAY:* | Contains the module positions for which Optomux will return the duration counter values. |
| *INFO ARRAY:* | The duration counter values will be passed back in the INFO array parameter. These values are returned in the INFO array elements that correspond to the particular module positions; i.e., the counter value for module position 0 is returned in the first element of the INFO array, the value for module position 4 is returned in the fifth array element, etc. |

## REMARKS

Attempts to read an output will result in a value of zero being returned for that position.

Values are returned in the current timer resolution (a value of 2 equals a pulse length of 2 x current time base), i.e., pulses up to 10.92 minutes can be timed with a resolution of 10 ms.

## EXAMPLE

This example sends a Read Pulse Duration Counters command to the Optomux at address 75, requesting Optomux to return the pulse duration counter values for positions 2 and 5 and then displays the returned values (assumed timer resolution of 10 ms).

```
300     ADDR% = 75                'Optomux Address Is 75
310     CMD% = 32                 'Read Duration Counters Command
320     POSITIONS%(0) = 2         'Specify Position 2
330     POSITIONS%(1) = 5         'Specify Position 5
340     POSITIONS%(2) = -1        'End Of List
350     GOSUB 1000                'Call The Driver
360     PRINT "Pulse at position #2 = ";INFO%(2)/100;"seconds"
370     PRINT "Pulse at position #5 = ";INFO%(5)/100;"seconds"

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# READ AND CLEAR DURATION COUNTERS 33

## PURPOSE

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

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 33.

*POSITIONS ARRAY:* Contains module positions for which Optomux will return and clear duration counter values.

*INFO ARRAY:* The duration counter values will be passed in the INFO array parameter. These values are returned in the INFO array elements that correspond to the particular module positions; i.e., the counter value for module position 0 is returned in the first element of the INFO array, the value for module position 4 is returned in the fifth array element, etc.

## REMARKS

Performs Read Pulse Duration Counters command then clears the counter and Pulse Complete Bits for the counters that were read.

Values are returned in the current timer resolution (a value of 2 equals a pulse length of 2 x current time base), i.e., pulses up to 10.92 minutes can be timed with a resolution of 10 ms.

## EXAMPLE

This example sends a Read And Clear Pulse Duration Counters command to the Optomux at address 75, requesting Optomux to return and clear the pulse duration counter values for positions 2 and 5. The returned values will be displayed (assumed timer resolution of 10 ms).

```
300    ADDR% = 75               'Optomux Address Is 75
310    CMD% = 33                'Read And Clear Duration Counters
320    POSITIONS%(0) = 2        'Specify Position 2
330    POSITIONS%(1) = 5        'Specify Position 5
340    POSITIONS%(2) = -1       'End Of List
350    GOSUB 1000               'Call The Driver
360    PRINT "Pulse at position #2 = ";INFO%(2)/100;"seconds"
370    PRINT "Pulse at position #5 = ";INFO%(5)/100;"seconds"

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# CLEAR DURATION COUNTERS 34

## PURPOSE

Clears the duration counters and Pulse Complete Bits for the specified positions to enable measurement of the next pulse.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 34. |
| *POSITIONS ARRAY:* | Contains the module positions for which Optomux will clear the duration counters and Pulse Complete Bits. |

## EXAMPLE

This example sends a Clear Duration Counters command to the Optomux at address 75, clearing the Pulse Complete Bits and duration counters for positions 2 and 5.

```
500    ADDR% = 75               'Optomux Address Is 75
510    CMD% = 34                'Clear Duration Counters Command
520    POSITIONS%(0) = 2        'Specify Position 2
530    POSITIONS%(1) = 5        'Specify Position 5
540    POSITIONS%(2) = -1       'End Of List
550    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# WRITE ANALOG OUTPUTS 35

## PURPOSE

Instructs Optomux to write a specified value to one or more analog outputs.

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 35. |
| *POSITIONS ARRAY:* | Contains the output positions that are to be written to. |
| *INFO ARRAY:* | The first element of the INFO array contains the value that will be written to each position specified in the POSITIONS array. Value range from 0 to 4,095. |

## EXAMPLE

This example writes 30% of full-scale to positions 3 and 4 on the Optomux at address 109.

```
800     ADDR% = 109               'Optomux Address Is 109
810     CMD% = 35                 'Write Analog Outputs Command
820     POSITIONS%(0) = 3         'Specify Position 3
830     POSITIONS%(1) = 4         'Specify Position 4
840     POSITIONS%(2) = -1        'End Of List
850     INFO%(0) = 0.30 * 4095    'Write 30% Of Full Scale
860     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# READ ANALOG OUTPUTS 36

## PURPOSE

Returns the value of each output specified in the POSITIONS array.

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 36. |
| *POSITIONS ARRAY:* | Contains the module positions for which Optomux will return the output data. |
| *INFO ARRAY:* | The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the value for module position 2, etc. |

## EXAMPLE

This example instructs the Optomux at address 86 to return the current values of the analog outputs at positions 7 and 8 and displays the results.

```
500    ADDR% = 86                          'Optomux Address Is 86
510    CMD% = 36                           'Read Analog Outputs Command
520    POSITIONS%(0) = 7                   'Specify Position 7
530    POSITIONS%(1) = 8                   'Specify Position 8
540    POSITIONS%(2) = -1                  'End Of List
560    GOSUB 1000                          'Call The Driver
570    PRINT "Position 7 is ";INFO%(7)     'Display Value Of Position 7
580    PRINT "Position 8 is ";INFO%(8)     'Display Value Of Position 8

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000              'Test For Errors
1020   RETURN
```

# READ ANALOG INPUTS 37

## PURPOSE

Returns the current value of each input position specified in the POSITIONS array.

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 37.

*POSITIONS ARRAY:* Contains the module positions for which Optomux will return values.

*INFO ARRAY:* The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the value for module position 2, etc.

## REMARKS

Reading an output position with the Read Analog Inputs command will return a -4,096 for the output position.

## EXAMPLE

This example instructs the Optomux at address 86 to return the current values of the analog inputs at positions 7 and 10. Assume that positions 7 and 10 contain 0 to 5-volt input modules.

```
500    ADDR% = 86               'Optomux Address Is 86
510    CMD% = 37                'Read Analog Inputs Command
520    POSITIONS%(0) = 7        'Specify Position 7
530    POSITIONS%(1) = 10       'Specify Position 10
540    POSITIONS%(2) = -1       'End of List
560    GOSUB 1000               'Call The Driver
570    PRINT "Voltage at position 7 is ";INFO%(7)/4095 * 5
580    PRINT "Voltage at position 10 is ";INFO%(10)/4095 * 5

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# AVERAGE AND READ INPUT 38

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 38.

*POSITIONS ARRAY:* The first element of the POSITIONS array contains the input position that Optomux is to average.

*INFO ARRAY:* The first element of the INFO array contains the number of samples that Optomux is to average (maximum 255). Return data will be passed back in the INFO array element corresponding to the module position being averaged. For example, an average for position 2 would be returned in the third element of the INFO array, etc.

## EXAMPLE

This example instructs the Optomux at address 211 to average position 4 over 8 samples and display the result.

```
100    ADDR% = 211                          'Optomux Address Is 211
110    CMD% = 38                            'Average And Read Command
120    POSITIONS%(0) = 4                    'Specify Position 4
130    INFO%(0) = 8                         'Average For 8 Samples
140    GOSUB 1000                           'Call The Driver
150    PRINT "The average = ";INFO%(4)      'Display The Results

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020   RETURN
```

# SET INPUT RANGE 39

## PURPOSE

Defines the high and low limits for the specified input positions.

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 39. |
| *POSITIONS ARRAY:* | Contains the positions for which Optomux will set high and low limits. |
| *INFO ARRAY:* | The high limit is passed in the first element of the INFO array and the low limit is passed in the second element. |

## REMARKS

This command defines a range for the specified inputs. If the input is found to be out of the specified range, one of two latches is set (high or low limit exceeded). The latches can be read using the Read Out-of-Range Latches command (command 40).

## EXAMPLE

This example sets the input range of positions 0 and 3 at 25% to 50% of full-scale on the Optomux at address 9.

```
500     ADDR% = 9                 'Optomux Address Is 9
510     CMD% = 39                 'Set Input Range Command
520     POSITIONS%(0) = 0         'Specify Position 0
530     POSITIONS%(1) = 3         'Specify Position 3
540     POSITIONS%(2) = -1        'End Of List
550     INFO%(0) = 4095 * 0.50    'High Limit Is 50% Of Full Scale
555     INFO%(1) = 4095 * 0.25    'Low Limit Is 25% Of Full Scale
560     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

## READ OUT-OF-RANGE LATCHES 40

### PURPOSE

Instructs Optomux to return the Out-of-Range Latches for all positions.

### VERSIONS

Analog.

### PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 40.

*INFO ARRAY:* The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. A value of 0 indicates that the position has remained within the limits or has been configured to function as an output. A value of 1 indicates that the low limit latch has been set. A value of 2 indicates that the high limit has been exceeded, and a value of 3 indicates that both high and low limit latches have been set.

### EXAMPLE

This example displays the current state of the Out-of-Range Latches from the Optomux at address 70.

```
100    ADDR% = 70                'Optomux Address Is 70
110    CMD% = 40                 'Read Out-of-Range Latches Command
120    GOSUB 1000                'Call The Driver
130    FOR I% = 0 TO 15          'Display The Results
140    PRINT "Module ";I%;
150    IF INFO%(I%) = 0 THEN PRINT " is within range"
160    IF INFO%(I%) = 1 THEN PRINT " has dropped below low limit"
170    IF INFO%(I%) = 2 THEN PRINT " has exceeded high limit"
180    IF INFO%(I%) = 3 THEN PRINT " has exceeded both limits"
190    NEXT

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# READ AND CLEAR OUT-OF-RANGE LATCHES 41

## PURPOSE

Performs the Read Out-of-Range Latches command (all positions), then resets the latches specified in the POSITIONS array.

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 41.

*POSITIONS ARRAY:* Contains the positions for which the Out-of-Range Latches are to be cleared.

*INFO ARRAY:* The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. A value of 0 indicates that the position has remained within the limits or has been configured to function as an output. A value of 1 indicates that the low limit latch has been set. A value of 2 indicates that the high limit has been exceeded, and a value of 3 indicates that both high and low limit latches have been set.

## EXAMPLE

This example instructs the Optomux at address 143 to return the Out-of-Range Latches and to clear the Out-of-Range Latches for positions 2 and 5.

```
100    ADDR% = 143              'Optomux Address Is 143
110    CMD% = 41                'Read And Clear Out-of-Range Latches Cmd
120    POSITIONS%(0) = 2        'Specify Position 2
130    POSITIONS%(1) = 5        'Specify Position 5
140    POSITIONS%(2) = -1       'End Of List
150    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

## CLEAR OUT-OF-RANGE LATCHES 42

### PURPOSE

Clear Out-of-Range Latches for specified positions.

### VERSIONS

Analog.

### PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 42.

*POSITIONS ARRAY:* Contains the positions that are to have their Out-of-Range latches cleared.

### EXAMPLE

This example instructs the Optomux at address 70 to clear the Out-of-Range latches for position 6.

```
600     ADDR% = 70                'Optomux Address Is 70
610     CMD% = 42                 'Clear Out-of-Range Latches Command
620     POSITIONS%(0) = 6         'Specify Position 6
630     POSITIONS%(1) = -1        'End Of List
640     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET OUTPUT WAVEFORM 43

## PURPOSE

Initiates a constant waveform at any of the output positions (see Enhanced Output Waveform command, 50).

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 43.

POSITIONS ARRAY: Contains the output positions where Optomux is to initiate waveforms.

MODIFIERS ARRAY: The first element of the MODIFIERS array contains a value from 0 to 14 which specifies the period of the ramp waveforms or one-half the period of the triangle and square waves. The times shown below indicate the resulting periods assuming a full-scale change:

| MODIFIERS%(0) | TIME |
|---|---|
| 0 | 2.18 minutes |
| 1 | 3.28 minutes |
| 2 | 4.37 minutes |
| 3 | 5.46 minutes |
| 4 | 6.56 minutes |
| 5 | 7.65 minutes |
| 6 | 8.74 minutes |
| 7 | 1.09 minutes |
| 8 | 32.8 seconds |
| 9 | 21.8 seconds |
| 10 | 16.4 seconds |
| 11 | 13.1 seconds |
| 12 | 10.9 seconds |
| 13 | 9.4 seconds |
| 14 | 8.2 seconds |

The second element of the MODIFIERS array contains a value from 0 to 7 which specifies what type of waveform is to be initiated:

| MODIFIERS%(1) | WAVEFORM DESCRIPTION |
|---|---|
| 0 | Stop waveform output |
| 1 | Triangle wave with positive initial slope |
| 2 | Ramp up, waveform terminates at upper limit |
| 3 | Continuous ramp up |
| 4 | Square wave (50% duty cycle) |
| 5 | Triangle wave negative initial slope |
| 6 | Ramp down, waveform terminates at lower limit |
| 7 | Continuous ramp down |

INFO ARRAY: The high limit is passed in the first element of the INFO array and the low limit is passed in the second element. If a lower value is passed in the first element of the INFO array, a -8 error (specified limits invalid) will be returned. The range limit is 0 to 255 (0 = zero scale, 255 = full-scale).

### EXAMPLE

This example instructs the Optomux at address 177 to initiate a continuous square wave at output modules 4, 13, and 14. The high limit is 170 (66.4% of full-scale at 8.74 minutes for full-scale change x 2).

```
500     ADDR% = 177               'Optomux Address Is 177
510     CMD% = 43                 'Set Waveform Command
520     POSITIONS%(0) = 4         'Specify Position 4
530     POSITIONS%(1) = 13        'Specify Position 13
540     POSITIONS%(2) = 14        'Specify Position 14
550     POSITIONS%(3) = -1        'End Of List
560     MODI%(0) = 6              'Period Is 8.74 Minutes
570     MODI%(1) = 4              'Select Square Wave
580     INFO%(0) = 170            'High Limit 66.4% Of Full Scale
590     INFO%(1) = 0              'Low Limit Is Zero Scale
600     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# TURN OFF EXISTING WAVEFORM 44

## PURPOSE

Turns off existing waveforms that were initiated using command 43.

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 44. |
| *POSITIONS ARRAY:* | Contains the positions where Optomux is to cancel waveforms. |

## REMARKS

This command will only cancel waveforms that were initiated using command 43. If the Enhanced Output Waveform command (command 50) was used to initiate the waveform then the Cancel Enhanced Waveforms command (command 51) must be used to cancel the Enhanced Output Waveform command.

## EXAMPLE

This example cancels the output waveform at position 2 on the Optomux at address 24.

```
500     ADDR% = 24              'Optomux Address Is 24
510     CMD% = 44               'Cancel Waveforms Command
520     POSITIONS%(0) = 2       'Specify Position 2
530     POSITIONS%(1) = -1      'End Of List
540     GOSUB 1000              'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

# SET ANALOG WATCHDOG 45

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 45.

*POSITIONS ARRAY:* Contains the module positions that will be affected by the command. A value of -1 indicates the end of the list.

*INFO ARRAY:* The first element of the INFO array: Info%(0) contains a value from 0 to 8, or 20 to 65,535. If Info%(0) parameter does not contain any data, a value of 0 is assumed.

For values of 0 - 8, the time scales and output values are fixed and will override any settings established by command 78 (SET ANALOG WATCH DOG USER-DEFINED VALUE).

Values of 9 - 19 will generate a limit error (error code -8), and the command will not be executed.

For values of 20 - 65,535, the value will be multiplied by 10 msec (200 msec to 10.923 minutes). This sets the time between no communications activity detected and the action to be taken. After the timer expires, OPTOMUX will write a user-defined value to the specified positions. The user must use command 78 (SET ANALOG WATCHDOG USER-DEFINED VALUE) to set the analog output values.

| Value | Action | Time |
|---|---|---|
| 0 | Watchdog timer disabled | Not applicable |
| 1 | Write zero scale | 10 seconds |
| 2 | Write zero scale | 1 minute |
| 3 | Write zero scale | 10 minutes |
| 4 | Watchdog timer disabled | Not applicable |
| 5 | Write full-scale | 10 seconds |
| 6 | Write full-scale | 1 minute |
| 7 | Write full-scale | 10 minutes |
| 8 | Watchdog timer disabled | Not applicable |
| 9 to 19 | Limit error | Not applicable |
| 20 to 65,535 | See command 78 | Value times 10 mSec. |

***Note:*** *The OPTOMUX unit will respond to the first command after a serial watchdog time-out with an error message (error code -7), 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.*

## REMARKS

When using a value from 20 to 65,535 for the first element of the INFO array, command 78 (Set Analog Watchdog Time-out) must also be used. This will allow for a variable time span and a variable output of the analog signal. Please see command 78 for detailed information.

## EXAMPLE

This example instructs the Optomux at address 202 to write full-scale to position 15 if there is no activity on the serial link for 10 minutes.

```
100     ADDR% = 202                 'Optomux Address Is 202
110     CMD% = 45                   'Set Analog Watchdog Delay Command
120     POSIT%(0) = 15              'Specify Position 15
130     POSIT%(1) = -1              'End Of List
140     INFO%(0) = 7                'Select 10 Minutes Write Full Scale
150     GOSUB 1000                  'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSIT%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# UPDATE ANALOG OUTPUTS 46

## PURPOSE

Writes values to one or more analog outputs.

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 46.

*POSITIONS ARRAY:* Contains the output positions that are to have values written to them. All other output positions will not be affected.

*INFO ARRAY:* Contains the values that will be written to the output positions included in the POSITIONS array. Values are passed in the INFO array elements that correspond to the particular module position; i.e., the value for module position 0 is contained in the first element of the INFO array, the value for module position 4 is included in the fifth array element, and so on.

## REMARKS

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

## EXAMPLE

This example sends an Update Analog Outputs command to the Optomux at address 75, writing 61% of full-scale to position 3 and 84% of full-scale to position 6.

```
300     ADDR% = 75                'Optomux Address Is 75
310     CMD% = 46                 'Read Duration Counters Command
320     POSITIONS%(0) = 3         'Specify Position 3
330     POSITIONS%(1) = 6         'Specify Position 6
340     POSITIONS%(2) = -1        'End Of List
350     INFO%(3) = 4095 * 0.61    'Write 61% Of Full Scale To Position 3
350     INFO%(6) = 4095 * 0.84    'Write 84% Of Full Scale To Position 6
360     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# START AVERAGING INPUTS 47

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 47. |
| *POSITIONS ARRAY:* | Contains the positions at which Optomux will start averaging. All other positions will not be affected. |
| *INFO ARRAY:* | Contains the number of samples that Optomux is to average. |

## REMARKS

The average values can be read back using the Read Averaged Inputs command (command 49). The Read Average Complete Bits command (command 48) can be used to determine whether Optomux has completed averaging the specified number of samples.

## EXAMPLE

This example instructs the Optomux at address 233 to start averaging the inputs at positions 7 and 8 for 15 samples.

```
500    ADDR% = 233              'Optomux Address Is 233
510    CMD% = 47                'Start Averaging Command
520    POSITIONS%(0) = 7        'Specify Position 7
530    POSITIONS%(1) = 8        'Specify Position 8
540    POSITIONS%(2) = -1       'End Of List
550    INFO%(0) = 15            'Number Of Samples To Average Is 15
560    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# READ AVERAGE COMPLETE BITS 48

## PURPOSE

Allows the host to determine which positions have completed averaging.

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 48.

*INFO ARRAY:* The return INFO will be passed back in the INFO array parameter. Each element of the INFO array will contain a -1 or a 0 indicating whether or not Optomux has completed averaging at one of the 16 module positions. Module positions corresponding to array elements set to -1 have completed averaging the desired number of samples. Optomux has not completed averaging at positions corresponding to array elements set to 0.

## EXAMPLE

This example sends a Read Average Complete Bits command to the Optomux at address 2. A message indicating whether or not each position has completed averaging is displayed. Assume that Optomux has been instructed to average inputs at positions 0 through 7.

```
100    ADDR% = 2                'Optomux Address Is 2
110    CMD% = 48                'Read Average Complete Bits Command
120    GOSUB 1000               'Call The Driver
130    FOR I% = 0 TO 7          'Print Status For Positions 0 Through 7
140    IF INFO%(I%) = 0 THEN GO TO 170
150    PRINT "Averaging is complete at position ";I%
160    GO TO 180
170    PRINT "Averaging is still in process at position ";I%
180    NEXT

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# READ AVERAGED INPUTS 49

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 49. |
| *POSITIONS ARRAY:* | Contains the input positions whose averages are to be returned. |
| *INFO ARRAY:* | The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the average value for the module position 2, etc. |

## REMARKS

The Read Average Complete Bits command can be used to determine when averaging is complete. If Optomux has not completed averaging, the current value of the average will be returned.

## EXAMPLE

This example instructs the Optomux at address 86 to return the averages for the analog inputs at positions 7 and 10. Assume that positions 7 and 10 contain 0 to 5 volt input modules and that Optomux has been previously instructed to average these inputs.

```
500     ADDR% = 86                'Optomux Address Is 86
510     CMD% = 49                 'Read Averaged Inputs Command
520     POSITIONS%(0) =7          'Specify Position 7
530     POSITIONS%(1) = 10        'Specify Position 10
540     POSITIONS%(2) = -1        'End Of List
560     GOSUB 1000                'Call The Driver
570     PRINT "Average voltage at position 7 is ";INFO%(7)/4095 * 5
580     PRINT "Average voltage at position 10 is ";INFO%(10)/4095 * 5

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# ENHANCED OUTPUT WAVEFORM 50

## PURPOSE

Initiates a constant waveform at any of the output positions.

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 50.

*POSITIONS ARRAY:* Contains the output positions where Optomux is to generate waveforms.

*MODIFIERS ARRAY:* The first element of the MODIFIERS array contains a value from 0 to6, which specifies what type of waveform is to be generated.

| MODIFIERS%(0) | WAVEFORM DESCRIPTION |
|---|---|
| 0 | Triangle wave with positive initial slope |
| 1 | Ramp up, waveform terminates at upper limit |
| 2 | Sawtooth, continuous ramp up |
| 3 | Square wave with 50% duty cycle |
| 4 | Triangle wave with negative initial slope |
| 5 | Ramp down, waveform terminates at lower limit |
| 6 | Sawtooth, continuous ramp down |

*INFO ARRAY:* The first element of the INFO array contains the high limit of the waveform (0 - 4,095). The second element of the INFO array contains the low limit of the waveform (0 - 4,095). The third element contains the period of the waveform (1 - 32,767 in 100 ms units). The value represents the period of half of the triangle or square waveforms. It is the entire period of the sawtooths and ramps waveforms.

### EXAMPLE

This example instructs the Optomux at address 92 to initiate a ramp up at position 12 with a low limit of 10% of full-scale and a high limit of 65% of full-scale for a period of five seconds.

```
500     ADDR% = 92                'Optomux Address Is 92
510     CMD% = 50                 'Improved Waveforms Command
520     POSITIONS%(0) =12         'Specify Position 12
530     POSITIONS%(1) = -1        'End Of List
540     MODI%(0) = 1              'Specify Ramp Up
550     INFO%(0) = 4095 * 0.65    'High Limit Is 65% Of Full Scale
560     INFO%(1) = 4095 * 0.10    'Low Limit Is 10% Of Full Scale
570     INFO%(2) = 50             'Set Period To 5 Seconds
580     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# CANCEL ENHANCED WAVEFORMS 51

## PURPOSE

Instructs Optomux to cancel the Enhanced Output Waveforms that were initiated using the Enhanced Output Waveform command (command 50).

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 51. |
| *POSITIONS ARRAY:* | Contains the positions at which waveforms are to be turned off. |

## REMARKS

This command can only be used to cancel waveforms that were initiated using the Enhanced Output Waveform command (command 50).

## EXAMPLE

This example instructs the Optomux at address 29 to stop generating output waveforms at positions 7 and 8.

```
600     ADDR% = 29                'Optomux Address Is 29
610     CMD% = 51                 'Cancel Enhanced Waveforms Command
620     POSITIONS%(0) =7          'Specify Position 7
630     POSITIONS%(1) = 8         'End Of List
650     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# CALCULATE INPUT OFFSETS 52

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 52. |
| *POSITIONS ARRAY:* | Contains the positions where Optomux is to return offsets. |
| *INFO ARRAY:* | The offset values will be returned in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the offset value for module position 2, etc. |

## REMARKS

The offset values are calculated using the current values of the inputs. Therefore, this command is effective only 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 command, it is necessary to set offsets for module positions before setting the gain coefficients.

## EXAMPLE

This example instructs the Optomux at address 26 to return the offset values for positions 4, 5, and 13.

```
500     ADDR% = 26              'Optomux Address Is 26
510     CMD% = 52               'Calculate Input Offsets Command
520     POSITIONS%(0) = 4       'Specify Position 4
530     POSITIONS%(1) = 5       'Specify Position 5
540     POSITIONS%(2) = 13      'Specify Position 13
550     POSITIONS%(3) = -1      'End Of List
560     GOSUB 1000              'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

# SET INPUT OFFSETS 53

## PURPOSE

Instructs Optomux to set offsets for specified input positions.

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 53. |
| *POSITIONS ARRAY:* | Contains the positions where Optomux is to set offsets. |
| *INFO ARRAY:* | Contains offset values for each module position specified in the POSITIONS array. The values are placed in the array element corresponding to the module position (offset for position 0 in first element of array). Offsets are placed in the INFO array in the same form as they are returned by the Calculate Input Offsets command. |

## REMARKS

Any 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

This example sets offsets for all 16 positions at the Optomux address 209. The offset values are assumed to be in an array named OFFSETS%( ), which was previously filled with the appropriate values.

```
500    ADDR% = 209              'Optomux Address Is 209
510    CMD% = 53                'Set Offsets Command
520    FOR I% = 0 TO 15
530    INFO%(I%) = OFFSETS%(I%) 'Assign Offsets To INFO Array
540    POSITIONS%(I%) = I%      'Specify Positions 0 Through 15
550    NEXT
560    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# CALCULATE AND SET INPUT OFFSETS 54

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 54. |
| *POSITIONS ARRAY:* | Contains the positions where Optomux is to set and return offsets. |
| *INFO ARRAY:* | The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the offset value for position 2, etc. |

## REMARKS

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 offset value returned by this command can be saved on a host computer's disk and used during system initialization via the SET OFFSETS command.

Attempts to calculate and set the offset for an output will have no effect.

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

## EXAMPLE

This example calculates and sets offsets for positions 2 and 3 on the Optomux at address 211. The offset values are saved to a corresponding array.

```
600    ADDR% = 211              'Optomux Address Is 211
610    CMD% = 54                'Calculate And Set Offsets Command
620    POSITIONS%(0) =2         'Specify Position 2
630    POSITIONS%(1) = 3        'Specify Position 3
640    POSITIONS%(2) = -1       'End Of List
650    GOSUB 1000               'Call The Driver
670    OFFSET%(2) = INFO%(2)    'Save Offset For Position 2
680    OFFSET%(3) = INFO%(3)    'Save Offset For Position 3

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# CALCULATE GAIN COEFFICIENTS 55

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 55. |
| *POSITIONS ARRAY:* | Contains the positions that Optomux is to return gain coefficients. |
| *INFO ARRAY:* | The gain coefficients will be returned in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the gain coefficient for module position 2, etc.<br><br>Values returned are 10,000 times the actual gain coefficients. For example, a value of 14,000 represents a gain coefficient of 1.40. This is the same format used by the Calculate And Set Gain Coefficients and the Set Gain Coefficients commands. |

## REMARKS

The gain coefficient 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 full-scale.

This command is usually used during system installation and calibration when known inputs (full-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 Gain Coefficients command.

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

## EXAMPLE

This example instructs the Optomux at address 260 to calculate and return gain coefficients for positions 4 and 5. The gain coefficients are then saved in a separate array for future use.

```
500     ADDR% = 260              'Optomux Address Is 260
510     CMD% = 55                'Calculate Gain Coefficients Command
520     POSITIONS%(0) = 4        'Specify Position 4
530     POSITIONS%(1) = 5        'Specify Position 5
550     POSITIONS%(3) = -1       'End Of List
560     GOSUB 1000               'Call The Driver
580     GAINCOEF%(4) = INFO%(4)  'Save Position 4 Coefficient
590     GAINCOEF%(5) = INFO%(5)  'Save Position 5 Coefficient
1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

# SET GAIN COEFFICIENTS 56

## PURPOSE

Instructs Optomux to set the gain coefficients for specified module positions.

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 56. |
| *POSITIONS ARRAY:* | Contains the positions where Optomux is to set gain coefficients. |
| *INFO ARRAY:* | Contains gain coefficients for each module position specified in the POSITIONS array. The values are placed in the array elements corresponding to the module positions (coefficient for position 0 in first element of array). Coefficients are placed in the INFO array in the same form as they are returned by the Calculate Gain Coefficients command. |

## REMARKS

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

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

## EXAMPLE

This example sets gain coefficients for positions 0 through 8 on the Optomux at address 209. The gain coefficients are stored in an array named GAINCOEF%( ).

```
500     ADDR% = 209               'Optomux Address Is 209
510     CMD% = 56                 'Set Gain Coefficients Command
520     FOR I% = 0 TO 8           'Get Values For Positions 0 Through 8
530     INFO%(I%) = GAINCOEF%(I%) 'Read The Coefficient For Position
        Number I%
540     POSITIONS%(I%) = I%       'Specify Positions I%
550     NEXT
560     POSITIONS%(I%) = -1       'End Of List
570     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# CALCULATE AND SET GAIN COEFFICIENTS 57

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 57. |
| *POSITIONS ARRAY:* | Contains the positions where Optomux is to set and return gain coefficients. |
| *INFO ARRAY:* | The gain coefficients will be returned in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the gain coefficient that has been set for module position 2, etc.<br><br>Values returned are 10,000 times the actual gain coefficients. For example, a value of 14,000 represents a gain coefficient of 1.40. This is the same format that is used by the Calculate Gain Coefficients and Set Gain Coefficients commands. |

## REMARKS

The gain coefficient values are calculated using the current values of the inputs. Therefore, this command is effective only 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. The values obtained can be used during Optomux initialization. Optomux can be instructed to use these gain coefficients values by including them in the Set Gain Coefficients command.

### EXAMPLE

This example instructs the Optomux at address 260 to calculate and set the gain coefficients for positions 4 and 5. The gain coefficients are then saved in a sequential file for future use.

```
500     ADDR% = 260                             'Optomux Address Is 260
510     CMD% = 57                               'Calculate And Set Gain Coeff
                                                Command
520     POSITIONS%(0) = 4                       'Specify Position 4
530     POSITIONS%(1) = 5                       'Specify Position 5
550     POSITIONS%(0) = -1                      'End Of List
560     GOSUB 1000                              'Call The Driver
570     OPEN "GAINCOEF" FOR OUTPUT AS #2        'Open File To Save Values
580     PRINT #2, INFO%(4)                      'Save Position 4 Coefficient
590     PRINT #2, INFO%(5)                      'Save Position 5 Coefficient
600     CLOSE #2                                'Close The File

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000                'Test For Errors
1020    RETURN
```

# READ LOWEST VALUES 58

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 58. |
| *POSITIONS ARRAY:* | Contains the input positions for which Optomux is to return lowest values. |
| *INFO ARRAY:* | The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the fifth element of the INFO array will contain data for module position 4, etc. |

## REMARKS

Optomux will return the lowest reading for each position specified in the positions array. The readings will be the lowest Optomux has encountered since last receiving a Read And Clear Lowest Values or Clear Lowest Values command.

## EXAMPLE

This example instructs the Optomux at address 68 to return the lowest values of the analog inputs at positions 7 and 10. Assume that positions 7 and 10 contain 0- to 5-volt input modules. A message is displayed showing the lowest voltage encountered at each position.

```
500     ADDR% = 68                'Optomux Address Is 68
510     CMD% = 58                 'Read Lowest Values Command
520     POSITIONS%(0) = 7         'Specify Position 7
530     POSITIONS%(1) = 10        'Specify Position 10
540     POSITIONS%(2) = -1        'End Of List
560     GOSUB 1000                'Call The Driver
570     PRINT "Lowest voltage at position 7 is ";INFO%(7)/4095 * 5
580     PRINT "Lowest voltage at position 10 is ";INFO%(10)/4095 * 5

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# CLEAR LOWEST VALUES 59

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 59.

*POSITIONS ARRAY:* Contains the positions where Optomux is to clear lowest readings. This will allow Optomux to store the lowest value encountered in subsequent readings.

## EXAMPLE

This example instructs the Optomux at address 125 to clear the lowest values at positions 0 through 6. All other positions are left unchanged.

```
500     ADDR% = 125              'Optomux Address Is 125
510     CMD% = 59                'Clear Lowest Values Command
520     FOR I% = 0 TO 6
530     POSITIONS%(I%) = I%      'Specify Positions 0 Through 6
540     NEXT
550     POSITIONS%(I%) = -1      'End Of List
560     GOSUB 1000               'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# READ AND CLEAR LOWEST VALUES 60

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 60.

*POSITIONS ARRAY:* Contains the input positions where Optomux is to return and clear lowest values.

*INFO ARRAY:* The lowest values will be passed back in the INFO array parameter. These values are returned in the INFO array elements that correspond to the particular module positions; i.e., the lowest value for module position 0 is returned in the first element of the INFO array, the value for module position 4 is returned in the fifth array element, etc.

## REMARKS

The values returned will be the lowest Optomux has encountered since last receiving a Read And Clear Lowest Values or Clear Lowest Values command. Optomux will clear the lowest value for each of the specified positions by setting each position's lowest value to 3,000 Hex (extreme over-range). This will enable Optomux to store the lowest value encountered in subsequent readings.

## EXAMPLE

This example sends a Read And Clear Lowest Values command to the Optomux at address 77, requesting the lowest values it has encountered at positions 4 and 5.

```
110    ADDR% = 77              'Optomux Address Is 77
120    CMD% = 60               'Read And Clear Lowest Values Command
130    POSITIONS%(0) = 4       'Specify Position 4
140    POSITIONS%(1) = 5       'Specify Position 5
160    POSITIONS%(2) = -1      'End Of List
170    GOSUB 1000              'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# READ PEAK VALUES 61

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 61. |
| *POSITIONS ARRAY:* | Contains the input positions where Optomux is to return peak values. |
| *INFO ARRAY:* | The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the fifth element of the INFO array will contain the peak value for module position 4, etc. |

## REMARKS

Optomux will return the peak reading for each position specified in the POSITIONS array. The readings will be the highest Optomux has encountered since last receiving a Read And Clear Peak Values or Clear Peak Values command.

## EXAMPLE

This example instructs the Optomux at address 70 to return the peak values of the analog inputs at positions 5 and 6. A message is displayed showing the peak values encountered at each position.

```
500    ADDR% = 70               'Optomux Address Is 70
510    CMD% = 61                'Read Peak Values Command
520    POSITIONS%(0) = 5        'Specify Position 5
530    POSITIONS%(1) = 6        'Specify Position 6
540    POSITIONS%(2) = -1       'End Of List
560    GOSUB 1000               'Call The Driver
570    PRINT "Position 5 peaked at ";INFO%(5)
580    PRINT "Position 6 peaked at ";INFO%(6)

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# CLEAR PEAK VALUES 62

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 62.

*POSITIONS ARRAY:* Contains the positions for which Optomux is to clear peak readings. This will allow Optomux to store the peak value encountered in subsequent readings.

## REMARKS

Optomux will clear the peak values for each input position specified in the POSITIONS array. This will allow Optomux to store the highest value encountered in subsequent readings.

## EXAMPLE

This example instructs the Optomux at address 2 to clear the peak values at positions 0 through 15.

```
500     ADDR% = 2                 'Optomux Address Is 2
510     CMD% = 62                 'Clear Peak Values Command
520     FOR I% = 0 TO 15
530     POSITIONS%(I%) = I%       'Specify Positions 0 Through 15
540     NEXT
550     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# READ AND CLEAR PEAK VALUES 63

## PURPOSE

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

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 63. |
| *POSITIONS ARRAY:* | Contains the input positions where Optomux will return and clear peak values. |
| *INFO ARRAY:* | The peak values will be passed back in the INFO array parameter. These values are returned in the INFO array elements that correspond to the particular module positions; i.e., the peak value for module position 0 is returned in the first element of the INFO array, the value for module position 4 is returned in the fifth array element, etc. |

## REMARKS

The values returned will be the highest Optomux has encountered since last receiving a Read And Clear Peak Values or Clear Peak Values command. Optomux will clear the peak value for each of the specified positions by setting each position's peak value to -4,096 (extreme under-range). This will enable Optomux to store the highest value encountered in subsequent readings.

## EXAMPLE

This example instructs the Optomux at address 0 to Return And Clear Peak Value at position 4.

```
110     ADDR% = 0                 'Optomux Address Is 0
120     CMD% = 63                 'Read And Clear Peak Values Command
130     POSITIONS%(0) = 4         'Specify Position 4
160     POSITIONS%(1) = -1        'End Of List
170     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

## READ BINARY ON/OFF STATUS 64

### PURPOSE

Returns the ON/OFF status of all module positions.

### VERSIONS

Digital.

### PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 64.

*INFO ARRAY:* The return data will be passed back in position 0. The data will be expressed as one 16-bit binary number. Bit 15 (MSB) of the returned data corresponds to module position 15 and bit 0 (LSB) corresponds to module position 0.

### EXAMPLE

This example sends a Read Binary ON/OFF Status command to the Optomux at address 0 and displays a message indicating the return status.

```
110    ADDR% = 0                              'Optomux Address Is 0
120    CMD% = 64                              'Read Binary Status Command
130    GOSUB 1000                             'Call The Driver
140    PRINT "Return data is ";INFO%(0)       'Print The Return Data

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000           'Test For Errors
1020   RETURN
```

# WRITE BINARY OUTPUTS 65

## PURPOSE

Turns output modules ON and OFF.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 65. |
| *INFO ARRAY:* | Position 0 will contain a 16-bit number indicating the output positions that are to be turned ON and OFF. |

## REMARKS

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.

Bit 15, the most significant bit, corresponds to module position 15 and bit 0, the least significant bit, corresponds to module position 0.

## EXAMPLE

This example turns ON the output modules in positions 2 and 3 on the Optomux at address 160. All other output positions will be turned off.

```
500     ADDR% = 160               'Optomux Address Is 160
510     CMD% = 65                 'Write Binary Outputs Command
520     INFO%(0) = &H000C         'Turn On 2 And 3 And Turn Off The Rest
550     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# READ BINARY LATCHES 66

## PURPOSE

Returns data indicating which of the inputs have latched.

## VERSIONS

Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 66.

*INFO ARRAY:* The return data will be passed back in the INFO array position 0. The data will be expressed as one 16-bit binary number. Bit 15 (MSB) of the returned data corresponds to module position 15 and bit 0 (LSB) corresponds to module position 0.

## REMARKS

This command has no effect on the Optomux latches. Subsequent Read Latches commands will return consistent results.

## EXAMPLE

This example sends a Read Binary Latches command to the Optomux at address 240 and displays a message indicating the latched/unlatched status of each of the 16 module positions.

```
110     ADDR% = 240              'Optomux Address Is 240
120     CMD% = 66                'Read Binary Latches Command
130     GOSUB 1000               'Call The Driver
140     FOR I% = 0 TO 15         'Display Status Of All latches
150     IF (INFO%(0) AND 1)
        = 0 THEN GO TO 180       '0 = Unlatched, 1 = Latched
160     PRINT "Position ";I%;" has latched"
170     GO TO 190
180     PRINT "Position ";I%;" has not latched"
190     INFO%(0) = INFO%(0) \ 2  'Use Integer Divide, Not (/)
200     NEXT

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSIT%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020    RETURN
```

## READ AND CLEAR BINARY LATCHES 67

### PURPOSE

Reads the current latches and then sets the specified positions to unlatched state.

### VERSIONS

Digital.

### PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 67.

*INFO ARRAY:* Array position 0 will contain a 16-bit number indicating the latch positions that are to be cleared. On return from the driver, array element 0 will now contain the latched positions.

### EXAMPLE

This example clears the latches for modules in positions 2 and 3 on the Optomux at address 160. All other latches will be unchanged.

```
100    MACHINE1% = 160          'Address For Machine #1 Is 160

500    ADDR% = MACHINE1%        'Optomux Address Is Machine #1
510    CMD% = 67                'Read And Clear Binary Latches Command
520    INFO%(0) = &H000C        'Specify Positions 2 And 3 To Be Cleared
550    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSIT%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# HI RESOLUTION SQUARE WAVE 68

## PURPOSE

Initiates a continuous square wave at the specified output positions using the current Optomux timer resolution (see command 75).

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 68. |
| *POSITIONS ARRAY:* | Contains the positions that are to output continuous square waves. All other positions will not be affected. |
| *INFO ARRAY:* | The first two elements of the INFO array contain values defining the on and off times of the square wave. The on time is defined as the current timer resolution x [the first element]. The off time is defined as the current timer resolution x [the second element]. The maximum for on and off times is 256 x 2.56 seconds (10.92 minutes). |

## REMARKS

A 0 value in the INFO locations is equivalent to a value of 256.

This commands operates the same as the standard square wave command (command 27), except that it uses the current timer resolution (set by command 75) instead of a resolution of 2.56 seconds. The square wave will continue until it is turned off using command 27 or a time delay is set for that position. The Write Digital Outputs, Activate Digital Outputs, and Deactivate Digital Output commands will have no effect.

## EXAMPLE

This example assumes that the current timer resolution is the default value (10 ms). The Optomux at address 92 is to initiate a continuous square wave at position 2 with an on time of 0.05 seconds and an off time of 0.12 seconds.

```
100     ADDR% = 92                'Optomux Address Is 92
110     CMD% = 68                 'Hi Resolution Square Wave Command
120     POSITIONS%(0) = 2         'Specify Position 2
130     POSITIONS%(1) = -1        'End Of List
140     INFO%(0) = 5              'On Time Is 5 * 10 ms = 0.05 Seconds
150     INFO%(1) = 12             'Off Time Is 12 * 10 ms = 0.12 Seconds
160     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

## RETRIGGER TIME DELAY 69

### PURPOSE

Instructs Optomux to set the current Time Delay setting into the specified time delay outputs.

### VERSIONS

Digital.

### PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 69. |
| *POSITIONS ARRAY:* | Contains the module positions that will be retriggered. |

### REMARKS

This command along with the Set Time Delay command (command 25) allows you to dynamically change an active time delay.

### EXAMPLE

This example instructs the Optomux at address 137 to retrigger the time delays for positions 2 and 3.

```
100    ADDR% = 137              'Optomux Address Is 137
110    CMD% = 69                'Retrigger Time Delay Command
120    POSITIONS%(0) = 2        'Specify Position 2
130    POSITIONS%(1) = 3        'Specify Position 3
140    POSITIONS%(2) = -1       'End Of List
150    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# READ CONFIGURATION 70

## PURPOSE

Returns the input/output configuration of the specified Optomux I/O unit.

## VERSIONS

Analog and Digital.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 70.

*INFO ARRAY:* Return data will be passed back in the INFO array parameter. Each element of the INFO array will contain a 1 or a 0 indicating the input/output (1 = output, 0 = input) status of one module position. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain a 1 or a 0, indicating the configuration of module position 2, etc. Remember, module positions are numbered 0 through 15.

## EXAMPLE

This example sends a Read Configuration command to the Optomux at address 0 and displays a message indicating the configuration of each of the 16 module positions.

```
110     ADDR% = 0                 'Optomux Address Is 0
120     CMD% = 70                 'Read Configuration Command
130     GOSUB 1000                'Call The Driver
140     FOR I% = 0 TO 15          'Display Config. Of All 15 Positions
150     IF INFO%(I%)
        = 0 THEN GO TO 180        '1 = Output, 0 = Input
160     PRINT "Module position ";I%;" is an output"
170      GO TO 190
180     PRINT "Module position ";I%;" is an input"
190     NEXT

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET ENHANCED DIGITAL WATCHDOG 71

## 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.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 71.

*POSITIONS ARRAY:* Contains the positions to be activated when Optomux does a watchdog time-out.

*INFO ARRAY:* The first element of the INFO array parameter contains a value from 0 through 65,535, which specifies time delay interval, in increments of 10 milliseconds. A 0 value in the first element of the INFO array will disable the watchdog delay. Delay values of less than 200 milliseconds (except for 0) will result in a Limit Error, and the command will not be executed.

## EXAMPLE

This example instructs the Optomux at address 92 (hex) to turn on output positions 0, 5, and 7 and turn off all other output positions if no activity is detected on the Optomux network for 0.6 seconds.

```
100     ADDR% = 92                  'Optomux Address Is 92
110     CMD% = 71                   'Enhanced Digital Watchdog Delay Command
130     INFO%(0) = 60               'Set Watchdog Delay To 0.6 Seconds
140     POSITIONS%(0) = 0           'Specify Position 0
150     POSITIONS%(1) = 5           'Specify Position 5
160     POSITIONS%(2) = 7           'Specify Position 7
170     POSITIONS%(3) = -1          'End Of List
180     GOSUB 1000                  'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# GENERATE N PULSES 72

## PURPOSE

Instructs Optomux to output a counted string of pulses of a specified duration.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 72. |
| *POSITIONS ARRAY:* | Contains the module positions that will be affected by this command. |
| *INFO ARRAY:* | The on and off time of the pulse is passed in the first element of the INFO array; the number of pulses is passed in the second element. |

## REMARKS

The on and off time for this command is a value from 0 to 255. Each pulse will be on for INFO%(0) times the current timer resolution and the off time is equal to INFO%(0) times the current timer resolution. If the value in INFO%(0) is a 0, any time delay or pulse stream currently being output by the affected outputs will be canceled.

## EXAMPLE

This example instructs the Optomux at address 137 to generate 10 pulses of 10 ms resolution (10 ms on and 10 ms off) on positions 2 and 3.

```
520    ADDR% = 137              'Optomux Address Is 137
530    CMD% = 72                'Generate N Pulses Command
540    POSITIONS%(0) = 2        'Specify Position 2
550    POSITIONS%(1) = 3        'Specify Position 3
560    POSITIONS%(2) = -1       'End Of List
570    INFO%(0) = 1             'Set For 10 ms Resolution
580    INFO%(1) = 10            'Do 10 Pulses
590    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# START ON PULSE 73

## PURPOSE

Instructs Optomux to activate the selected modules for a specified period of time.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 73. |
| *POSITIONS ARRAY:* | Contains the module positions that will be pulsed on. |
| *INFO ARRAY:* | The delay length is specified in the first element of the INFO array. The range of possible delay lengths is 0 to 65,535 (in units of the current timer resolution). |

## REMARKS

If the value contained in the INFO array is a 0, this command does nothing. This command is retriggerable.

## EXAMPLE

This example sets positions 2 and 3 on the Optomux at address 137 to turn on for 1.2 seconds and then turn off (timer resolution is assumed at 10 ms).

```
100    ADDR% = 137              'Optomux Address Is 137
110    CMD% = 73                'Start On Pulse Command
120    POSITIONS%(0) = 2        'Specify Position 2
130    POSITIONS%(1) = 3        'Specify Position 3
140    POSITIONS%(2) = -1       'End Of List
150    INFO%(0) = 120           'Delay Length Is 1.20 Seconds
160    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# START OFF PULSE 74

## PURPOSE

Instructs Optomux to deactivate the selected modules for a specified period of time and then activate the modules.

## VERSIONS

Digital.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 74. |
| *POSITIONS ARRAY:* | Contains the module positions that will be pulsed off. |
| *INFO ARRAY:* | The delay length is specified in the first element of the INFO array. The range of possible delay lengths is 0 to 65,535 (in units of the current timer resolution). |

## REMARKS

If the value contained in the INFO array is a 0, this command does nothing. This command is retriggerable.

## EXAMPLE

This example sets positions 2 and 3 on the Optomux at address 137 to turn off for 500 milliseconds and then turn on (timer resolution is assumed at 10 ms).

```
520    ADDR% = 137              'Optomux Address Is 137
530    CMD% = 74                'Start Off Pulse Command
540    POSITIONS%(0) = 2        'Specify Position 2
550    POSITIONS%(1)= 3         'Specify Position 3
560    POSITIONS%(2) = -1       'End Of List
580    INFO%(0) = 50            'Delay Length Is 0.50 Seconds
590    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

## SET TIMER RESOLUTION 75

### PURPOSE

This command sets the timer resolution on the specified Optomux I/O unit.

### VERSIONS

Digital.

### PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 75. |
| *INFO ARRAY:* | The timer resolution is specified in the first element of the INFO array. The range of possible delay lengths is 0 to 255 (in units of the 10 ms). |

### REMARKS

If the value contained in the INFO array is a 0, the timer resolution will be 2.56 seconds. This command affects all time-related commands except for the watchdog commands and the Set Turnaround Delay command.

### EXAMPLE

This example sets the timer resolution to 50 milliseconds on the Optomux at address 137.

```
100    ADDR% = 137              'Optomux Address Is 137
110    CMD% = 75                'Set Timer Resolution Command
120    INFO%(0) = 5             'Set Timer Resolution To 50 ms
130    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# SET TEMPERATURE PROBE TYPE 76

## PURPOSE

Sets the temperature probe type for the read temperature command (command 77).

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 76. |
| *POSITIONS ARRAY:* | Contains the module positions that will be set to the desired probe type. |
| *INFO ARRAY:* | The probe type is specified in the first element of the INFO array. The range of possible probe types is 0 to 65,535 (only 0 through 9 are currently used). |

## REMARKS

The following lists the probe types:

0 - no temperature probe
1 - ICTD probe
2 - 10 ohm RTD probe
3 - 100 ohm RTD probe
4 - Type J thermocouple
5 - Type K thermocouple
6 - Type R thermocouple
7 - Type S thermocouple
8 - Type T thermocouple
9 - Type E thermocouple

## EXAMPLE

This example sets the probe type on modules 1, 5, and 7 to Type J thermocouple on the Optomux at address 137.

```
520    ADDR% = 137              'Optomux Address Is 137
530    CMD% = 76                'Set Probe Type Command
540    POSITIONS%(0) = 1        'Specify Position 1
550    POSITIONS%(1) = 5        'Specify Position 5
560    POSITIONS%(2) = 7        'Specify Position 7
570    POSITIONS%(3) = -1       'End Of List
580    INFO%(0) = 4             'Select Probe As Type J Thermocouple
590    GOSUB 1000               'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# READ TEMPERATURE INPUTS 77

## PURPOSE

Instructs Optomux to return the temperature, in 16th of a degree Celsius, at the specified input positions.

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 77. |
| *POSITIONS ARRAY:* | Contains the input positions whose temperatures are to be returned. |
| *INFO ARRAY:* | The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the temperature value for the module position 2, etc. |

## REMARKS

The returned temperature is in 16th of a degree. To convert to a floating point temperature, divide the returned value by 16. If you read a temperature from a position that has not had its probe type set, you will get back a value of -4,096.

## EXAMPLE

This example instructs the Optomux at address 86 to return the temperatures for the analog inputs at positions 7 and 10.

```
500    ADDR% = 86               'Optomux Address Is 86
510    CMD% = 77                'Read Temperature Inputs Command
520    POSITIONS%(0) = 7        'Specify Position 7
530    POSITIONS%(1) = 10       'Specify Position 10
540    POSITIONS%(2) = -1       'End Of List
560    GOSUB 1000               'Call The Driver
570    PRINT "Temperature at position 7 is ";INFO%(7)/16;" degrees C"
580    PRINT "Temperature at position 10 is ";INFO%(10)/16;" degrees C"

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000         'Test For Errors
1020   RETURN
```

# SET ANALOG WATCHDOG USER-DEFINED VALUES 78

## PURPOSE

This command, used with command 45, sets the data which Optomux will output upon a serial watchdog time-out.

## VERSIONS

Analog.

## PARAMETERS

*ADDRESS:* Contains the address of the Optomux I/O unit.

*COMMAND:* Contains the value 78.

*POSITIONS ARRAY:* Contains the output positions that are to have time-out data set for them. All other output positions will not be affected.

*INFO ARRAY:* Contains the values that will be written to the output positions included in the POSITIONS array. Values are passed in the INFO array elements that correspond to the particular module position; i.e., the value for module position 0 is contained in the first element of the INFO array, the value for module position 4 is included in the fifth array element, etc.

## REMARKS

Command 45, Set Analog Watchdog, must be called before the Set Analog Watchdog User-Defined Values command will work. Valid values for command 45 are 1 to 8 and 20 to 65,535. However, valid values for command 45 are 20 to 65,535 when command 78 is also being used. Please refer to command 45 for detailed information.

## EXAMPLE

This example sets the time-out for the Optomux at address 92 to go to half scale on output positions 0, 5, and 7.

```
100    ADDR% = 92              'Optomux Address Is 92
110    CMD% = 78               'Set Watchdog Time-out Command
120    POSIT%(0) = 0           'Specify Position 0
130    POSIT%(1) = 5           'Specify Position 5
140    POSIT%(2) = 7           'Specify Position 7
150    POSIT%(3) = -1          'End Of List
160    INFO%(0) = 2048         'Set To Half Scale
170    INFO%(5) = 2048         'Set To Half Scale
180    INFO%(7) = 2048         'Set To Half Scale
190    GOSUB 1000              'Call The Driver

1000   CALL OptoWare (ERRS%,ADDR%,CMD%,POSIT%(0),MODI%(0),INFO%(0))
1010   IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020   RETURN
```

# READ AVERAGE TEMPERATURE INPUTS 79

## PURPOSE

Instructs Optomux to return the average temperature, in 16th of a degree Celsius, at the specified input positions initiated by the Start Averaging Input command (command 47).

## VERSIONS

Analog.

## PARAMETERS

| | |
|---|---|
| *ADDRESS:* | Contains the address of the Optomux I/O unit. |
| *COMMAND:* | Contains the value 79. |
| *POSITIONS ARRAY:* | Contains the input positions whose average temperatures are to be returned. |
| *INFO ARRAY:* | The return data will be passed back in the INFO array parameter. The index into the INFO array parameter corresponds to the module position. For example, the third element of the INFO array will contain the average temperature value for the module position 2, etc. |

## REMARKS

The returned temperature is in 16th of a degree. To convert to a floating point temperature, divide the returned value by 16. If you read a temperature from a position that has not had its probe type set, you will get back a value of -4,096.

## EXAMPLE

This example instructs the Optomux at address 86 to return the average temperatures for the analog inputs at positions 7 and 10.

```
500     ADDR% = 86                'Optomux Address Is 86
510     CMD% = 79                 'Read Average Temperature Inputs Cmd
520     POSITIONS%(0) = 7         'Specify Position 7
530     POSITIONS%(1) = 10        'Specify Position 10
540     POSITIONS%(2) = -1        'End Of List
560     GOSUB 1000                'Call The Driver
570     PRINT "Temperature at position 7 is ";INFO%(7)/16;" degrees C"
580     PRINT "Temperature at position 10 is ";INFO%(10)/16;" degrees C"

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

## SET DRIVER PROTOCOL 100

### PURPOSE

Sets the driver to two-pass or four-pass protocol.

### VERSIONS

Analog and Digital.

### PARAMETERS

*COMMAND:* Contains the value 100.

*INFO ARRAY:* The first element of the INFO array contains 0 or 1:

```
INFO%(0) = 0            Specifies two-pass protocol
INFO%(0) = 1            Specifies four-pass protocol
```

### REMARKS

This command sets the protocol of the driver. It is necessary for the protocol of the driver to match that of Optomux in order to carry on valid message transactions. Optomux can be set to use either two- or four-pass protocol by using the Set Optomux Protocol command (command 4). On power up, Optomux protocol is determined by jumper B10 on the Optomux board. If you want to change the protocol that is currently being used on the Optomux network, change the protocol that Optomux is using via command number 4, and then change the protocol of the driver to match Optomux I/O unit. The driver defaults to two-pass protocol.

### EXAMPLE:

This example sends a Set Protocol command to the Optomux at addresses 0 through 25 instructing each Optomux to use four-pass protocol. It then sets the driver to four-pass protocol.

```
500     CMD% = 4                'Set Optomux Protocol Command
510     INFO%(0) = 1            'Specify Four-pass Protocol
520     FOR I% = 0 TO 25        'Send Command To Addresses 0 to 25
530     ADDR% = I%              'Set Address Parameter
540     GOSUB 1000              'Call The Driver
550     NEXT
560     CMD% = 100              'Set Command To Set Driver Protocol
570     GOSUB 1000              'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000        'Test For Errors
1020    RETURN
```

## SET TURNAROUND DELAY 101

### PURPOSE

Sets the length of time the driver will wait for a response from Optomux before returning an error code.

### VERSIONS

Analog and Digital.

### PARAMETERS

*COMMAND:* Contains the value 101.

*INFO ARRAY:* The first element of the INFO array contains the value indicating the desired delay length. The delay is given in 10 ms units. Maximum delay length is 32,000 ms. The driver default delay value is 3 seconds.

### REMARKS

The turnaround delay is strictly a software timing loop. If you have a machine other than a 4.77 MHz machine you will have to adjust the time-out value accordingly. Most AT type machines will have to increase the time-out values by 10 (a value of 3,000 will be close to a three second time-out).

### EXAMPLE

This example tells the driver to wait 1.5 seconds before issuing a turnaround time-out (-29) error in the event that Optomux does not respond to a command.

```
100     CMD% = 101                'Set Turnaround Delay Command
110     INFO%(0) = 150            'Set A Delay Of 1.5 Seconds (150 * 10 ms)
120     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET SERIAL PORT NUMBER 102

## PURPOSE

Tells the driver which serial port to use.

## VERSIONS

Analog and Digital.

## PARAMETERS

*COMMAND:* Contains the value 102.

*INFO ARRAY:* The first element of the INFO array contains the value indicating the desired serial port. A value of 0 specifies COM0, 1 specifies COM1, etc. The driver default is COM1.

For PC DOS-based machines, there are four possible serial ports: COM1 – COM4.

For the LC4, there are four possible serial ports: COM0 – COM3.

On the LC2, there are two serial ports: COM0 and COM1.

## REMARKS

This command should be performed before the Configure Serial Port command (command 104) is executed.

This information identifies the serial ports available on the PC:

| AC24<br>AC422 | PORT ADDRESS | INTERRUPT | JUMPERS | INFO |
|---|---|---|---|---|
| COM1 | 3F8HEX | 4 | COM1 | 1 |
| COM2 | 2F8HEX | 3 | COM2,A8 | 2 |
| COM3 | 348HEX | 2 | IRQ2, A4,A5,A7 | 3 |
| COM4 | 340HEX | 5 | IRQ5,A3,A4,A5,A7 | 4 |

LC4 Serial Ports:

| PORT | DESCRIPTION | INFO (0) |
|---|---|---|
| COM0 | Host Port | 0 |
| COM1 | Optomux Port | 1 |
| COM2 | EX2 Daughter Board Port | 2 |
| COM3 | EX2 Daughter Board Port | 3 |

LC2 Serial Ports:

| PORT | DESCRIPTION | INFO (0) |
|---|---|---|
| COM0 | Host Port | 0 |
| COM1 | Optomux Port | 1 |

### EXAMPLE

This example instructs the driver to use COM1 to communicate with Optomux I/O unit.

```
200     CMD% = 102                  'Select Serial Port Command
210     INFO%(0) = 1                'Specify COM Port 1
220     GOSUB 1000                  'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# SET NUMBER OF RETRIES 103

## PURPOSE

Sets the number of times the driver will attempt to send a command to Optomux before it returns an error message.

## VERSIONS

Analog and Digital.

## PARAMETERS

*COMMAND:* Contains the value 103.

*INFO ARRAY:* The first element of the INFO array parameter contains the desired number of attempts. The driver default is 1.

## REMARKS

A value of 0 in the INFO array will select the retry count to 65,536.

## EXAMPLE

This example instructs the driver to return error codes after the third unsuccessful attempt to communicate with Optomux I/O unit.

```
100     CMD% = 103                'Set Number Of Retries Command
110     INFO%(0) = 3              'Specify 3 Retries
120     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# CONFIGURE SERIAL PORT 104

## PURPOSE

Tells the driver to configure the current serial port.

## VERSIONS

Analog and Digital.

## PARAMETERS

*COMMAND:* Contains the value 104.

*INFO ARRAY:* The first element of the INFO array contains the value indicating the desired baud rate. Valid baud rates are 300, 600, 1200, 2400, 4800, 9600, 19200, and 38,400 baud. When setting the INFO element for 38,400 baud, you must use &H9600 (Hex representation) because 38,400 is outside the legal integer range.

## REMARKS

Besides setting the baud rate on the current port, the data word format for the serial port will be set to 8 data bits, 1 stop bit, and no parity. This command should follow the Set Serial Port Number command (command 102). Command 104 is not available on the LC2 and LC4 Local Controllers, since the baud rate is selected by hardware jumpers.

## EXAMPLE

This example instructs the driver to set the baud rate to 38,400 baud.

```
200     CMD% = 104                'Configure Serial Port Command
210     INFO%(0) = &H9600         'Specify 38.4 K Baud
220     GOSUB 1000                'Call The Driver

1000    CALL OptoWare (ERRS%,ADDR%,CMD%,POSITIONS%(0),MODI%(0),INFO%(0))
1010    IF ERRS% < 0 THEN GOSUB 2000          'Test For Errors
1020    RETURN
```

# APPENDIX A

## USING THE OPTOWARE DRIVER WITH INTERPRETED BASIC

To use the OptoWare driver with IBM's BASIC or Microsoft's GW-BASIC, you must first load the OptoWare driver subroutine into memory. This is done using two BASIC statements, DEF SEG and BLOAD.

The DEF SEG statement tells BASIC which segment to put the OptoWare driver in. The OptoWare driver must be loaded above BASIC's memory. A typical value for a system with 640 K of memory is 3,300 Hex.

The BLOAD statement tells the BASIC interpreter to actually load the OptoWare driver into the defined segment of memory allocated by the DEF SEG statement.

The CALL statement must contain a variable with a value that indicates where the subroutine is located relative to the current defined segment. A convenient name would be OptoWare because this name must be used when using the compiler version of BASIC.

An example of using the OptoWare driver in interpreted BASIC would be as follows:

```
@PROGRAM TEXT = 100              DEF SEF=&H3300 'Define Segment For Driver
110 BLOAD "DRIVER.COM", 0        'Load The OptoWare Driver with 0 offset
120 OptoWare = 0                 'Use OptoWare To CALL The Driver

600 CALL OptoWare( ERROR%, ADDRESS%, COMMAND%, POSITIONS%(0), MODIFIERS%(0),
INFO%(0))
```

For additional information on the DEF SEG and BLOAD statements, please refer to your BASIC manual.

# APPENDIX B

## SAMPLE APPLICATION PROGRAM

```
1010  '*
1020  '*
1030  '*
1040  '*   SAMPLE PROCESS CONTROL APPLICATION
1050  '*
1060  '*
1070  '*
1080  '*   SYSTEM CONFIGURATION
1090  '*
1100  '*         Host Computer: IBM PC/XT/AT
1110  '*
1120  '*         Serial communications:  COM1, 19200 baud
1130  '*
1140  '*         4-POINT DIGITAL (B1/PB4H)
1150  '*
1160  '*         POSITION     FUNCTION                TYPE
1170  '*
1180  '*           0          START BUTTON INPUT      INPUT
1190  '*           1          SPARE                   INPUT
1200  '*           2          TANK FILL VALVE         OUTPUT
1210  '*           3          TANK HEATER             OUTPUT
1220  '*
1230  '*
1240  '*
1250  '*         4-POINT ANALOG (B2/PB4AH)
1260  '*
1270  '*         POSITION     FUNCTION                TYPE
1280  '*
1290  '*           0          TANK TEMPERATURE        INPUT
1300  '*           1          SPARE                   INPUT
1310  '*           2          TANK LEVEL DETECTOR     INPUT
1320  '*           3          EMPTY TANK VALVE        OUTPUT
1330  '*
1340  '*
2010  '*
2020  '*
2030  '*
```

```
2040    '*    4-POINT DIGITAL OPTOMUX CONFIGURATION EQUATES
2050    '*
2060    DIGITAL% = 255          '* Address Of Digital OPTOMUX = 255
2070    START% = 0              '* Start Button Is Position 0
2080    FILLERVALVE% = 2                '* Valve To Fill Tank Is Position 2
2090    HEATER% = 3             '* Heater Is Module Position 3
2100    '*
2110    '*
2120    '*    4-POINT ANALOG OPTOMUX CONFIGURATION EQUATES
2130    '*
2140    '*
2150    '*
2160    ANALOG = 254            '* Address Of Digital OPTOMUX = 254
2170    TEMPERATURE = 0                 '* Tank Temperature Is Position 0
2180    LEVEL% = 2              '* Level Detector Is Position 2
2190    EMPTYVALVE% = 3         * Valve To Empty Tank Is Position 3
2200    '*
2210    '*
2220    '*
2230    '*    OPTOMUX COMMAND EQUATES
2240    '*
2250    '*
2260    PWRUPCLR% = 0                   '* Power Up Clear
2270    RESETOPMX% = 1                  '* Reset
2280    CONFIQURE% = 6                  '* Configure Positions As Input/Outputs
2290    ACTIVATE% = 10                  '* Activate Digital Outputs

2300    DEACTIVATE% = 11                '* Deactivate Digital Outputs
2310    READONOFF% = 12                 '* Read Digital On/Off Status
2320    READLATCHES% = 16               '* Read Latches
2330    CLEARLATCHES% = 18              '* Clear Latches
2340    READANLOUT% = 36                '* Read Analog Outputs
2350    READANLIN% = 37                 '* Read Analog Inputs
2360    SETLIMITS% = 40                 '* Set Analog Input Range
2370    TESTLIMITS% = 41                '* Read Over/Under Range Flags
2380    CLEARLIMITS% = 42               '* Clear Out-Of-Range Limits
2390    SETWAVE% = 43                   '* Set Wave Form
2411    '*
2412    '*    INITIALIZE CONSTANTS FOR TANK LEVELS AND TARGET TEMPERATURE
2413    '*
2416    TARGETTEMP% = (80+307.12)/0.1487109    '* Target Temp Is 80 Degrees F
2417    FULL% = 0.8 * 4095                     '* Tank Is Full At 80% Of Full Scale
2418    EMPTY% = 0.2 * 4095                    '* Tank Is Empty At 20% Of Full Scale
2419    CLOSED% = 0.2 * 4095                   '* Valve Is Closed At 20% Of Full Scale
2420    '*
3000    '*
3010    '*    DIMENSION AND INITIALIZE DRIVER PARAMETERS
3020    '*
```

```
3030    DIM POSITIONS%(15)              '* Dimension POSITIONS Array To 16 Elements
3040    DIM MODIFIERS%(1)               '* Dimension MODIFIERS Array To 2 Elements
3050    DIM INFO%(15)                   '* Dimension INFO Array To 16 Elements
3060    ERRORS% = 0                             '* Initialize ERRORS To 0
3070    ADDRESS% = 0                    '* Initialize Address To 0
3080    CMD% = 0                        '* Initialize Command To 0
3090    FOR I% = 0 TO 15                '* Set All Of The POSITIONS And INFO
3100    POSITIONS%(I%) = 0              '* Array Elements To 0
3110    INFO%(I%) = 0
3120    NEXT
3130    MODIFIERS%(0) = 0               '* Initialize The MODIFIERS Array To 0
3140    MODIFIERS%(1) =  0
3150    '*
3160    '*    LOAD THE DRIVER SUBROUTINE
3170    '*
3190    DEF SEG = &H3300                '* Define Segment Tells BASIC Where To Load
3200    BLOAD "DRIVER.COM",0            '* Load The Driver At Segment
3210    OPTOWARE = 0                    '* Use "OPTOWARE" To Call The Driver
3211    '*
3212    '*    CONFIGURE THE SERIAL PORT
3213    '*
3220    OPEN "com1:19200,n,8,1,cs,ds,rs,cd" AS #1
3230    CLOSE #1
4010    '*************************************************************************
4020    '*
4030    '*    MAIN SEQUENCER ROUTINE
4040    '*
4050    '*************************************************************************
4060    '*
4070    '*
4071    KEY OFF:CLS
4080    GOSUB 6100                              '* Go Initialize OPTOMUX
4090    IF ERRORS% < 0 THEN GOTO 4310           '* Test For Errors
4100    '*
4110    GOSUB 7080                              '* Wait For Start Button
4120    IF ERRORS% < 0 THEN GOTO 4310           '* Test For Errors
4130    '*
4140    GOSUB 8080                              '* Go Fill Tank
4150    IF ERRORS% < 0 THEN GOTO 4310           '* Test For Errors
4160    '*
4170    GOSUB 9080                              '* Go Heat Fluid
4180    IF ERRORS% < 0 THEN GOTO 4310           '* Test For Errors
4190    '*
4200    GOSUB 10070                             '* Go Empty Tank
4210    IF ERRORS% < 0 THEN GOTO 4310           '* Test For Errors
4220    '*
4230    CLS : GOTO 4110
4240    '*
4250    '*
```

```
4260    '*    ERROR RECOVERY CODE
4270    '*
4280    '*
4290    '*    DISPLAY MESSAGE INDICATING ERROR
4300    '*
4310    PRINT "Error number ";ERRORS%;" encountered at address ";ADDRESS%
4320    FOR I% = 1 TO 10: BEEP: NEXT              '* Make Some Noise To Wake Up
4330                                              '* The Operator
4340    '*
4350    '*    ATTEMPT TO REINITIALIZE DIGITAL AND ANALOG OPTOMUX
4360    '*
4370    GOSUB 6100                                '* Go Do OPTOMUX Initialization
4380    IF ERRORS% < 0 THEN GOTO 4310             '* Keep Trying Until Successful
4381    PRINT " SYSTEM HAS BEEN REINITIALIZED"
4390    PRINT " HIT RETURN TO RESTART SYSTEM ";
4400    IF INKEY$ = "" THEN GOTO 4400
4410    GOTO 4110                                 '* Go Look For Start
4430    '*
5010    '*************************************************************************
5020    '*
5030    '*    DRIVER CALLING ROUTINE
5040    '*
5050    '*************************************************************************
5060    '*
5070    '*
5080    CALL OPTOWARE(ERRORS%,ADDR%,CMD%,POSITIONS%(0),MODIFIERS%(0),INFO%(0))
5090    RETURN
5100    '*
5110    '*
6010    '*************************************************************************
6020    '*
6030    '*    INITIALIZE THE OTPOMUX CONTROLLERS
6040    '*
6050    '*************************************************************************
6060    '*
6070    '*
6080    '*    INITIALIZE DIGITAL OPTOMUX
6090    '*
6100    ADDRESS% = DIGITAL%                       '* Set The Address To Digital
6110    CMD% = PWRUPCLR%                                '* First Send Power Up Clear
6120    GOSUB 5080                                '* Call Driver To Send Command
6130    IF ERRORS% < 0 THEN RETURN                '* Return To Main Sequencer If We Have Errors
6150    CMD% = RESETOPMX%                         '* Now Send A Reset Command
6160    GOSUB 5080                                '* Call The Driver To Send Reset
6170    IF ERRORS% < 0 THEN RETURN                '* If Error Return To Main SequencerRoutine
6190    CMD% = CONFIQURE%                         '* Set Command To Configure
6200    POSITIONS%(0) = FILLERVALVE%                    '* Put Output Positions In
6210    POSITIONS%(1) = HEATER%                         '* The POSITIONS Array
6220    POSITIONS%(2) = -1                        '* End Of List
```

```
6230    GOSUB 5080                          '* Call Driver To Send Command
6240    IF ERRORS% < 0 THEN RETURN          '* Return To Main Sequencer If
6250                                        '* Any Errors Were Detected
6260    '*
6270    '*    INITIALIZE ANALOG OPTOMUX
6280    '*
6290    ADDRESS% = ANALOG%                  '* Set The Address To Analog
6300    CMD% = PWRUPCLR%                            '* First Send Power Up Clear
6310    GOSUB 5080                          '* Call Driver To Send Command
6320    IF ERRORS% < 0 THEN RETURN          '* Return To Main Sequencer If We Have Errors
6340    CMD% = RESETOPMX%                   '* Now Send A Reset Command
6350    GOSUB 5080                          '* Call The Driver To Send Reset
6360    IF ERRORS% < 0 THEN RETURN          '* If Error Return To Main Sequencer Routine
6380    CMD% = CONFIQURE%                   '* Set Command To Confiqure
6390    POSITIONS%(0) = EMPTYVALVE%         '* Put Output Positions In
6400    POSITIONS%(1) = -1                  '* The POSITIONS Array Followed By -1
6410    GOSUB 5080                          '* Call Driver To Send Command
6420    RETURN
6421    '*
7010    '*************************************************************************
7020    '*
7030    '*    WAIT FOR START BUTTON
7040    '*
7050    '*************************************************************************
7080    ADDRESS% = DIGITAL%                 '* Set Address To Digital
7090    CMD% = READLATCHES%                 '* Set Command To Read Latches
7100                                        '* Test For Start Button By
7110    '*
7120    GOSUB 5080                          '* Reading Latches
7120    IF ERRORS% < 0 THEN RETURN          '* Return To Main Sequencer If
7130                                        '* An Error Has Been Detected
7140    '*
7150    '*    TEST LATCH TO SEE IF THE BUTTON HAS BEEN PUSHED
7160    '*    IF IT HAS, RETURN - OTHERWISE KEEP WAITING
7170    '*
7180    IF INFO%(START%) = 0 THEN GOTO 7110
7181    '*
7182    '*    CLEAR THE START BUTTON LATCH
7183    '*
7185    CMD% = CLEARLATCHES%                '* Set Command To Clear Latches
7186    POSITIONS%(0) = START%              '* Put Start Button Position In
7187    POSITIONS%(1) = -1                  '* POSITIONS Array Followed By -1
7188    GOSUB 5080                          '* Call The Driver
7190    RETURN
7191    '*
8010    '*************************************************************************
8020    '*
8030    '*    FILL THE TANK
8040    '*
```

```
8050    '*************************************************************************
8071    '*    OPEN DIGITAL FILLER VALVE
8072    '*
8080    ADDRESS% = DIGITAL%                 '* Set Address To Digital
8090    CMD% = ACTIVATE%                    '* Set Command To Activate Outputs
8100    POSITIONS%(0) = FILLERVALVE%                '* Specify Filler Valve In POSITIONS
8110    POSITIONS%(1) = -1                  '* Array, -1 Marks End Of List
8120    GOSUB 5080                          '* Call Driver To Open Valve
8130    IF ERRORS% < 0 THEN RETURN          '* Return To Main Sequencer On Error
8140    '*
8150    '*    READ ANALOG LEVEL DETECTOR TO DETERMINE WHEN TANK IS FULL
8160    '*
8170    ADDRESS% = ANALOG%                  '* Set Address To Analog
8180    CMD% = READANLIN%                   '* Set Command To Read Analog Inputs
8190    POSITIONS%(0) = LEVEL%              '* Specify Level Detector In POSITIONS
8200    POSITIONS%(1) = -1                  '* Array, -1 Marks End Of List
8210    GOSUB 5080                          '* Call Driver To Read Level
8220    IF ERRORS% < 0 THEN RETURN          '* Return To Main Sequencer On Error
8230    '*
8240    '*    IF TANK IS NOT FULL THEN WAIT UNTIL IT IS
8250    '*
8260    IF INFO%(LEVEL%) < FULL% THEN GOTO 8210
8270    '*
8280    '*    TANK IS FULL SO CLOSE FILLER VALVE
8290    '*
8300    ADDRESS% = DIGITAL%                 '* Set Address To Digital
8310    CMD% = DEACTIVATE%                  '* Set Command To Deactivate Outputs
8320    POSITIONS%(0) = FILLERVALVE%                '* Specify Filler Valve In POSITIONS
8330    POSITIONS%(1) = -1                  '* Array, -1 Marks End Of List
8340    GOSUB 5080                          '* Call Driver To Close Valve
8350    RETURN
9010    '*************************************************************************
9020    '*
9030    '*    HEAT FLUID
9040    '*
9050    '*************************************************************************
9071    '*    TURN ON HEATER
9072    '*
9080    ADDRESS% = DIGITAL%                 '* Set Address To Digital
9090    CMD% = ACTIVATE%                    '* Set Command To Activate Outputs
9100    POSITIONS%(0) = HEATER%                     '* Specify Heater In POSITIONS
9110    POSITIONS%(1) = -1                  '* Array, -1 Marks End Of List
9120    GOSUB 5080                          '* Call Driver To Turn On Heater
9130    IF ERRORS% < 0 THEN RETURN          '* Return To Main Sequencer On Error
9140    '*
9150    '*    READ TEMPERATURE TO SEE IF IT'S TOO LOW
9160    '*
9170    ADDRESS% = ANALOG%                  '* Set Address To Analog
9180    CMD% = READANLIN%                   '* Set Command To Read Analog Inputs
```

```
9190    POSITIONS%(0) = TEMPERATURE%          '* Specify ICTD Temp Probe In POSITIONS
9200    POSITIONS%(1) = -1                    '* Array, -1 Marks End Of List
9210    GOSUB 5080                            '* Call Driver To Read Temperature
9220    IF ERRORS% < 0 THEN RETURN            '* Return To Main Sequencer On Error
9230    '*
9240    '*    IF TEMP IS TOO LOW THEN WAIT FOR IT TO RISE
9250    '*
9260    IF INFO%(TEMPERATURES%) < TARGETTEMP% THEN GOTO 9210
9270    '*
9280    '*    TURN OFF HEATER
9290    '*
9300    ADDRESS% = DIGITAL%                   '* Set Address To Digital
9310    CMD% = DEACTIVATE%                    '* Set Command To Deactivate Outputs
9320    POSITIONS%(0) = HEATER%                       '* Specify Heater In POSITIONS
9330    POSITIONS%(1) = -1                    '* Array, -1 Marks End Of List
9340    GOSUB 5080                            '* Call Driver To Close Valve
9360    '*
9370    '*    READ TEMPERATURE TO SEE IF WE WENT OVER THE TARGET
9380    '*
9390    ADDRESS% = ANALOG%                    '* Set Address To Analog
9400    CMD% = READANLIN%                     '* Set Command To Read Analog Inputs
9410    POSITIONS%(0) = TEMPERATURE%          '* Specify ICTD Temp Probe In POSITIONS
9420    POSITIONS%(1) = -1                    '* Array, -1 Marks End Of List
9430    GOSUB 5080                            '* Call Driver To Read Temperature
9440    IF ERRORS% < THEN RETURN              '* Return To Main Sequencer On Error
9450    '*
9460    '*    IF TEMP IS TOO HIGH THEN WAIT FOR IT TO DROP BEFORE WE RETURN
9470    '*
9480    IF INFO%(TEMPERATURE%) < TARGETTEMP% THEN GOTO 9430
9490    RETURN
10010   '*************************************************************************
10020   '*
10030   '*    EMPTY TANK
10040   '*
10050   '*************************************************************************
10060   '*
10062   '*    RAMP ANALOG VALVE OPEN OVER A 16.4 SECOND PERIOD
10063   '*
10070   ADDRESS% = ANALOG%                    '* Set Address To Analog
10080   CMD% = SETWAVE%                       '* Set Up For Waveform Command
10090   MODIFIERS%(0) = 10                    '* Select 16.4 Second Period
10100   MODIFIERS%(1) = 2                     '* Select Continuous Ramp Up
10110   POSITIONS%(0) = EMPTYVALVE%           '* Specify Tank Empty Valve
10120   POSITIONS%(1) = -1                    '*  End Of List
10121   INFO%(0) = 255                        '* Specify High Scale For High Limit
10122   INFO%(1) = 0                          '* Specify 0 Scale For Low Limit
10130   GOSUB 5080                            '* Call Driver To Open Valve
10140   IF ERRORS% < 0 THEN RETURN            '* Return To Main Sequencer If
10150                                         '* Any Errors Were Detected
```

```
10160  '*
10170  '*    READ THE TANK LEVEL TO SEE IF IT'S EMPTY
10180  '*
10190  CMD% = READANLIN%                    '* Set Command To Read Analog Inputs
10200  POSITIONS%(0) = LEVEL%               '* Put Position Of Level Detector
10210  POSITIONS%(1) = -1                   '* In POSITIONS Array
10230  GOSUB 5080                           '* Call Driver To Read Level
10240  IF ERRORS% < 0 THEN RETURN           '* Return If There Were Errors
10241  '*
10242  '*    KEEP READING THE TANK LEVEL UNTIL IT'S EMPTY
10245  '*
10250  IF INFO%(LEVEL%) < EMPTY% THEN GOTO 10230   '* If Tank Not Empty, Wait
10260  '*
10270  '*    RAMP THE ANALOG VALVE CLOSED NOW THAT THE TANK IS EMPTY
10280  '*
10290  CMD% = SETWAVE%                      '* Setup For Wave Form Command
10300  MODIFIERS%(0) = 14                   '* Select 8.2 Second Period
10310  MODIFIERS%(1) = 6                    '* Select Continuous Ramp Down
10320  POSITIONS%(0) = EMPTYVALVE%          '* Specify Tank Empty Value
10330  POSITIONS%(1) = -1                   '* End Of List
10340  GOSUB 5080                           '* Call Driver To Open Valve
10350  IF ERRORS% < 0 THEN RETURN           '* Return To Main Sequencer If
10360                                       '* Any Errors Were Detected
10370  '*
10380  '*    READ VALUE OF EMPTY VALVE OUTPUT TO SEE IF VALVE IS CLOSED
10390  '*
10400  CMD% = READANLOUT%                   '* Set Command To Read Analog Outputs
10410  GOSUB 5080                           '* Call Driver To Read Value Of Output
10420                                       '* Controlling The Empty Valve
10430  IF ERRORS% < 0 THEN RETURN           '* Return To Main Sequencer If
10440                                       '* Any Errors Were Detected
10450  '*
10460  '*    IF VALVE IS NOT CLOSED, WE HAVE TO WAIT
10470  '*
10480  IF INFO%(EMPTYVALVE%) < CLOSED% GOTO 10410
10490  RETURN
```

# I/O Configuration for this Example

### 4–POINT DIGITAL OPTOMUX

| | | |
|---|---|---|
| Address 255 | Position 0 | Input; Start Button |
| | Position 1 | Spare |
| | Position 2 | Input; Tank Filler Valve |
| | Position 3 | Tank Heater |

### 4–POINT ANALOG OPTOMUX

| | | |
|---|---|---|
| Address 254 | Position 0 | Input; ICTD Temperature Sensor |
| | Position 1 | Spare |
| | Position 2 | Input; 0-5 Volt Level Detector |
| | Position 3 | Output; 0-5 Volt Valve |

# Overview of this Sample Application

This Application Repeatedly Goes through the Steps Below:

| | |
|---|---|
| Step 1 | Initialize Optomux |
| Step 2 | Wait for Operator Start |
| Step 3 | Fill tank with fluid |
| Step 4 | Heat fluid to temperature |
| Step 5 | Empty tank |
| Step 6 | Go back to 'Wait for operator start' |

Optomux errors are also checked when communicating to Optomux I/O. If an Optomux error occurs, the operator is alerted and the I/O is re-initialized.

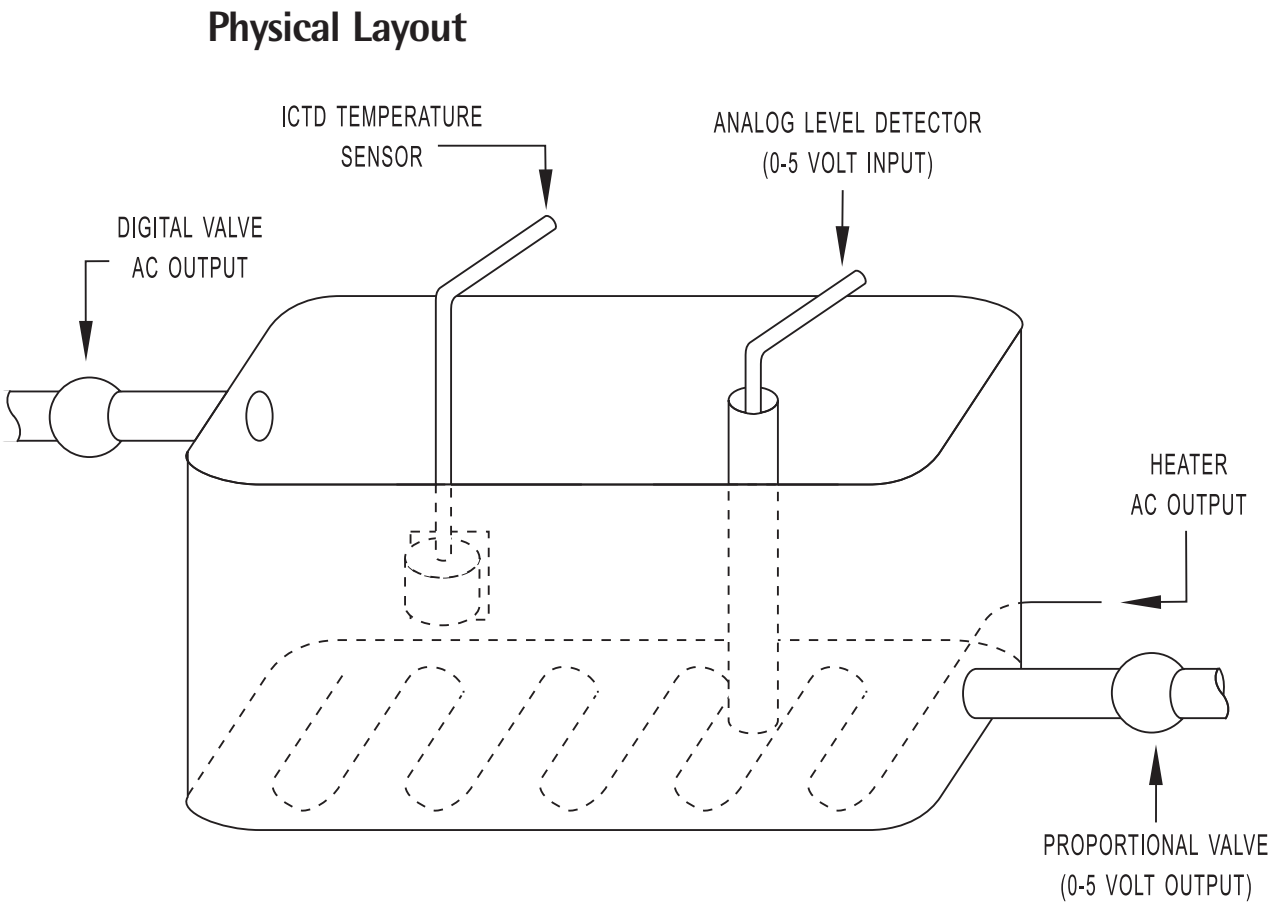


**Figure 1-1: Physical Layout for this Sample Application**

# APPENDIX C

## USING OPTO 22'S UTILITY PROGRAMS

The following three programs are located in the Opto Driver Toolkit.

### HOST

The HOST program allows the user to send ASCII Hex characters out the serial port to the Optomux network. This is useful in checking out the Optomux hardware and configuration and/or debugging your own Optomux driver. To run the program, type HOST at the DOS prompt. A menu of selections will be displayed on the screen. Below is a copy of the HOST screen.

The HOST program automatically enables the CAPS LOCK key, because the Optomux commands use mostly upper-case alphabets. The program defaults to COM1, 19200 baud, two-pass protocol, repeat count of one, and error detect on and checksum enabled.

The following is a summary of the commands to use HOST:

1. Press 1 to enter Optomux commands into the buffer. The program puts the start of command character (>) at the beginning of the line. If you use the default settings, the program also computes the checksum of that command line. To complete the command line, enter a carriage return. To continue entering additional commands, type the next Optomux command on the next line. To get back to the main menu, enter a carriage return on a blank line.
2. Press 2 to get into the Immediate mode. In the Immediate mode, the command you enter on the screen will be transmitted immediately when you enter a carriage return. The program puts the start of command character (>) at the beginning of the command and, if the checksum enable is on, computes the checksum of that command line. Pressing the ESC key will return you to the main menu.
3. Press 3 to display the command(s) in the command buffer on the screen.
4. Press 4 to change the repeat count. This selects how many iterations the command buffer will be send to the Optomux network.
5. Press 5 to toggle between two- and four-pass protocol. This must match the jumper setting of the B10 jumper on the Optomux Brain Board (B1 or B2).
6. Press 6 to select one of four combinations listed below.

**Table C-1: Error Checking Options**

| ERROR DETECT | CHECKSUM ENABLE |
|---|---|
| ON | ON |
| OFF | OFF |
| ON | OFF |
| OFF | ON |

With the ERROR DETECT off, the audible error beep and the pause functions are disabled. This is useful when you want to put an oscilloscope on the communication line. With the CHECKSUM ENABLE off, the program does not compute the checksum of the Optomux command; you must compute the checksum or enter two question marks (??).

Press P to cycle through the six possible serial port selections. The serial port's Hex address is also displayed on the main menu. The fifth and sixth serial port addresses are for communicating from Paragon LC to an LC4 as COM 3 and 4.

Press B to cycle through the possible baud rate selections. The baud rates are: 300, 600, 1,200, 2,400, 4,800, 9,600, 19,200, 38,400, 57,600, and 115,200.

Press Q to quit the HOST program and get back to DOS.

Press the carriage return key to transmit the command(s) in the command buffer. Some computers label the carriage return key as "Enter" or "Return."

For information on the Optomux commands, please refer to the *Optomux B1 And B2 Digital And Analog Brain Boards Operations Manual,* Form 203.

## Optoscan

OptoScan is a utility program that reads your Optomux network and responds back with the settings of all the Optomux stations on the network. It also allows you to enter the parameters of a given Optomux station to poll that station for information and allow you to make changes to that station.

The following are tutorial/demos on how to use OptoScan on your Optomux network. To run the program, type OS at the DOS prompt.

***Example one:***

After receiving your Optomux system, set your Group A and B jumpers on the brain boards.

1. At the main menu, move the cursor to the baud and enter the desired baud rate.
2. Move the cursor and select the desired protocol (two- or four-pass, [2] or [4]), station address ([0] to [255]), link mode ([M]ultidrop or [R]epeat mode), and last board ([Y]es or [N]o).
3. After making the appropiate selections, the Group A and B Jumpers Sections will display the correct jumper settings for Groups A and B.
4. To get new settings, repeat steps 1 through 3.

***Example two:***

After completing example one and wiring the Optomux communications network, you would like to test the communications link.

1. At the main menu, move the cursor to PORT and enter the serial port number of the Optomux network (1 thru 4). Press the F4 function key, and enter the desired delay (use the default value). The OptoScan program will communicate to the Optomux network starting at station address 0 and 300 baud. OptoScan will increment the station address from 0 to 255 at 300 baud, sending a Power Up Clear command to the Optomux board address. If the board responds with an acknowledge, it will send an identify Optomux Type command. However, if the board responds with a -28 error (invalid return data), OptoScan will change to four-pass protocol and re-issue the Power Up Clear command. If the board responds with an acknowledge, OptoScan will record the settings for the station address, switch back to two-passs protocol, and increment to the next station address. After incrementing to station address 255, OptoScan will double the baud rate and start at station address 0. OptoScan will increment up to station address 255 at 38,400 baud and stop. On the right side of the main menu, OptoScan will display the station address, Optomux type, baud rate, and protocol settings of all the Optomux stations from which OptoScan received an acknowledge response.

2. Check your configuration with the display on the right side of the main menu, if you need to change any of your Optomux station settings refer to example one.

***Example three:***

After completing examples one and two and all the field wiring, you would like to read from and write to field devices on a particular Optomux station.

1. Set the settings for that Optomux station on the upper left side of the main menu (BAUD, DELAY, PROTOCOL, and ADDRESS). Toggle to AUTO scan by pressing the F3 function key and then press the F2 function key to SCAN the particular Optomux station. The status of all 16-points is displayed in the middle of the main menu.

2. Configure the 16-points using the up and down arrow keys and the F9 function key to toggle between input and output module. OptoScan displays 16 positions whether you can use a four, eight, or 16-point I/O mounting rack. For a digital input module, when the field input turns on the LED on the I/O mounting rack, it will turn on and OptoScan will change from OFF to ON at that module position. For a digital output module, you can turn ON/OFF the output module by pressing the F9 function key with the cursor positioned at the module position under the STATE column. Press the F10 function key to get this command back to the main menu.

3. For analog modules, step 2 is still valid. However, the VALUE column displays the raw count (0 to 4,095) for the analog modules instead of the ON/OFF status. To enter data, you need to press the F9 function key and then an integer number from 0 to 4,095 (0 to 100 percent of scale).

4. If you want a record of any errors, press the F5 function key to active the error logging routine. The errors are logged onto a file named ERROR.TXT.

***Note:*** *Press the F1 function key for help and follow the instructions on the bottom of the screen. Press the F10 function key to get to the previous screen or to exit from OptoScan.*

## USER.EXE

The USER program allows you to use the OptoWare driver to communicate to the Optomux network. The USER program allows you to enter data into the different variables (ERRORS, ADDRESS, COMMAND, POSITIONS, MODIFIERS, and INFO) used by the OptoWare driver and then call the driver to send the command to the Optomux network. The following is a demostration of how to use the USER program. To run the program, type USER at the DOS prompt.

1. To use the USER program, you must first set the serial port number and then configure the serial port. To set the serial port number from the main menu, use the arrow keys to move the cursor over to Driver title and press the carriage return. This displays the Driver menu with the list of driver commands. Use the arrow keys to move the cursor down to the Set Port Number command (command 102) and press the carriage return. The OptoWare command menu will be displayed. The OptoWare Command menu is divided into two parts. The upper half describes the command you have selected and contains an example using the command in IBM BASIC. Use the PgUp and PgDn keys to page up and down the descriptions half of the screen.

The lower half of the screen contains the variables used by OptoWare. Refer to the upper half of the screen or the OptoWare manual regarding how to use the command. Use the arrow keys to move the cursor to the different variables and variable array elements. To set the serial port number, use the arrow keys to move the cursor to INFO ARRAY (0) and enter the appropriate serial port number (1 to 4).

Once the OptoWare command is set correctly, press the F2 function key to call the OptoWare driver and execute the command. If there is an error in the OptoWare driver or communication to Optomux, an error number and error message will be displayed. Otherwise, the ERRORS variable is set to 0 and the Driver Status displays a "Command finished" message.

2. The next command is to configure the serial port. Press the ESC key to get back to the Driver menu and select the Configure Serial Port command (command 104). Press carriage return to get to the OptoWare command menu. To configure the serial port, enter the baud rate in INFO ARRAY (0) (i.e., 300, 600, ..., 38,400.).

Once the OptoWare command is set correctly, press the F2 function key to call the OptoWare driver and execute the command. If there is an error in the OptoWare driver or communication to Optomux, an error number and error message will be displayed. Otherwise, the ERRORS variable is set to 0 and the Driver Status displays a "Command finished" message.

3. This concludes setting up the serial port for communication to the Optomux network. The next step would be to issue System commands (i.e., Power Up Clear, Set Turn Around Delay, etc.) to each Optomux board. After the System commands, the Configure commands must be issued. The Configure commands informs the OptoWare driver which positions are inputs, outputs, or temperature probes.

4. After issuing the Driver, System, and Configure commands, you are ready to read and write to the I/O modules and setup the Latches, Time Delays, Waveforms, etc. commands.

# APPENDIX D

The Optomux.DLL driver allows a Windows application to communicate with Optomux I/O. The driver is implemented as a Windows DLL providing a single "C" API (function) called "optoware" and may be used with any Windows language that supports DLLs such as Visual BASIC or C/ C++ compilers. The driver uses any port that is compatible with the Windows serial communications drivers such as a normal RS-232 port, any of Opto 22's RS-485 cards or 3rd part cards.

## INSTALLATION

The file Optomux.Dll should be copied to either the application directory or the Windows System directory. Windows searches for a DLL in the following sequence: [1] in memory, [2] the current working directory [3] Windows System directory.

Header files (Optomux.H for C and Optomux.Bas for BASIC) should be copied to the directory where the application source code is located.

The file Optomux.Lib is provided as an import library. This tells the linker that the Optomux API is part of a DLL. The LIB file is not required for Visual BASIC.

## SIMPLE EXAMPLE

This simple example initializes the dirver and issues a "power-up-clear" command to an Optomux board. If communications are successful then an error of zero is returned.

```
#include "Optomux.H"
#define SET_TURN_AROUND_DELAY 101
#define SET_SERIAL_PORT_NUMBER 102
#define CONFIGURE_SERIAL_PORT 104
#define POWER_UP_CLEAR          0


void main()

{

    int Positions[16];
    int Modifiers[16];
    int Info[16];
    int Error;
    int MyComPort = 1;
    int MyBaud = 19200;
    int MyAddress = 0;
```

```
Info[0] = 250;

Error = optoware(
    MyAddress, SET_TURN_AROUND_DELAY, Positions, Modifiers, Info );

Info[0] = MyComPort; // Select port COM1

Error = optoware(
    MyAddress, SET_SERIAL_PORT_NUMBER, Positions, Modifiers, Info );

Info[0] = MyBaud; // Set baud rate

Error = optoware(
    MyAddress, CONFIGURE_SERIAL_PORT, Positions, Modifiers, Info );

Error = optoware(
    MyAddress, POWER_UP_CLEAR, Positions, Modifiers, Info );

} // main()
```

# ERROR CODES

A complete list of error codes with a brief description is provided in the Optomux.H file. Common errors are:

| Error number | Description |
|---|---|
| 0 | No error |
| -29 | No response. Address may be wrong or cable unplugged. |
| -31 | Checksum error. Possible noise on the communication line. |

# APPENDIX E

## USING OPTO 22 I/O IN 32 BIT WINDOWS

Both Optomux and Mistic I/O may be accessed in Win32 using OptoMwd.DLL. This DLL allows multiple applications to access I/O simultaneously. This DLL provides a set of "C" APIs and may be used with any 32-bit Windows language that supports DLLs such as Visual Basic or C/ C++ compilers.

### PORT LOCKING AND MULTIPLE APPLICATIONS

Port locking is implemented so that multiple applications can share the same port. If an application needs exclusive access to a port to perform multiple commands without interruption, an API is provided to lock and unlock a port.

### PORT TYPES SUPPORTED

This driver supports these ports:

- RS-232 - Uses the Windows serial driver to access a standard RS-232 port or any Opto 22 serial card such as an AC24AT or AC422AT.
- Opto22 AC37 card - Can operate up to 115,200 baud and has a coprocessor to buffer communications.

### HANDLES

OptoMwd.DLL uses handles to access ports. A handle is provided when a port is opened and thereafter the handle is used to access a port.

### INSTALLATION

Run the OptoDriver Toolkit installation to install I/O drivers, examples and on-line documentation.

### EXAMPLES

Examples are provided in the OptoDriver Toolkit for several languages. These examples are installed when the OptoDriver Toolkit is installed.

## API LIST

For most applications, only 3 APIs in the library are required as follows:

- Open a port and get a port handle
- Read or write data
- Close the port and release the port handle

See the code fragment below shows a simple example without error handling.

## Simple Code Fragment Using These APIs

```
// Step [1] - Open a port and get a port handle.
// There's a separate 'open' API for Arcnet, AC37 and a COM port.
ErrorCode = opto22MwdPortOpenXXX( &MwdHandle, ...);

// Step [2a] - For Mistic/Snap, call SendMIO to interact with I/O.
ErrorCode = SendMIO( MwdHandle, ...);

// Step [2b] - For Optomux I/O, call SendOptomux to interact with I/O.
ErrorCode = SendOptomux( MwdHandle, ...);
printf( "\"%s\", %d chars, Error:%d\n", ReceStr,iActualLen,iErrorCode );

// Step [3] Close the port when the application ends.
opto22MwdPortClose( MwdHandle );
```

Notes about APIs:

- The APIs are listed below with the function prototypes and a description.
- Refer to OptoMwd.H for actual function prototypes or OptoMwd.BAS for function declarations.
- In general, an API that returns an "int" returns an error number where 0 indicates "no error".
- Each API name starts with "opto22Mwd" except for SendMIO.

## opto22MwdGetVersion, opto22MwdGetVersion2

```
char* FAR PASCAL opto22MwdGetVersion( void );

int opto22MwdGetVersion2(
  char* versionArg,
  UINT  maxLenArg );
```

The "get version" APIs get a version string of the form "R1.9z". The purpose of these functions is to allow an application to check the version of the DLL file. **opto22MwdGetVersion** returns a pointer to a string and **opto22MwdGetVersion2** copies the version string to a buffer provided by the caller.

## SendMIO

```
O22_ERROR_CODE SendMIO(
  int               iHandle,  // Handle provided by 'port open'
  int               iAddress, // Address of brick
  unsigned int      iCommand, // Command number i.e., 202=set output
  unsigned int far* PositionArray,  // Position array - 2 elements
  long far*         SendDataArray,  // Send data array - 16 elements
  long far*         ReceDataArray); // Recv data array - 16 elements
```

Sends a command to Mistic I/O and gets the corresponding response. Returns an Opto 22 error code where a value of zero indicates no error.

## SendOptomux

```
O22_ERROR_CODE SendOptomux(
  int               iHandle,     // Handle provided by 'port open'
  int               iAddress,    // Address of brick
  unsigned int      iCommand,    // Command number i.e., Max is 79
  long far*         cPosition,   // 16 element array
  unsigned int far* ModifierArray,  // Modifier array - 2 elements
  long far*         DataArray);  // Send Data Array, 16 elements
```

Sends a command to Optomux I/O and gets the corresponding response. Returns an Opto 22 error code where a value of zero indicates no error.

## opto22MwdPortOpen for AC37 and WinApi

```
int opto22MwdPortOpenAC37(
  int*  handle,
  UINT  ioPort, // I/O address such as 3F8 (IRQ not required)
  DWORD Baud,   // Baud rate such as 115200
  float timeOut,
  UINT  retry,
  int   protocolType,
  int   dataCheckType );

int opto22MwdPortOpenWinApi(
  int*  handle,
  UINT  comPort,       // COM port number. 1 for COM1.
  DWORD Baud,          // Baud rate up to 38400
  float timeOut,
  UINT  retry,
  int   protocolType,    // Protocol type: ASCII or Binary
  int   dataCheckType ); // Data check type (should be CRC)
```

The "PortOpen" APIs open a port and assign a handle number to be used by the application to access the port.

Parameters common to all "PortOpen" APIs:

**timeOut** is the amount of time to wait, in seconds, for a response. A typical number to use for ARCNET or high speed serial is 0.75 seconds. Larger values are needed for lower baud rates or if several applications are accessing the same port.

**retry** is the quantity of times the driver will retry a command if an error occurs. The driver always tries once but the number of reties depends on this value. A typical number is 1.

**protocolType** should be either **ProtocolTypeBinary** or **ProtocolTypeAscii** which are defined in OptoMwd.H. Windows 3.1 on an RS232 port will not support Binary mode. Typically, Binary is better because twice as many characters need to be sent for ASCII. ASCII may be required if communicating via modem.

**dataCheckType** should be **DataCheckTypeCrc16** which is defined in OptoMwd.H.

### opto22MwdPortClose

```
void opto22MwdPortClose( int HandleArg );
```

This API closes a port and releases the handle. If this API is not called by an application before exiting, the port handle will not be released until the DLL is released from memory. Visual Basic should call this API when the main form unloads.

## VISUAL BASIC FUNCTIONS

### StringAsLong, StringAsFloat

```
void FAR PASCAL StringAsLong(long* numArg, char* StringArg);
void FAR PASCAL StringAsFloat(float* numArg, char* Stringarg);
```

These two APIs are used by Visual Basic to extract Longs or Floats from a response string that contains binary data. A response from mistic can be ASCII (such as "1.234E12") or binary (which would appear as 4 bytes of garbage characters) depending upon the command. Host Words allows ASCII or Binary mode to be selected. "C" doesn't need these two functions because C can use type casts to convert the data.

## STATUS OR ERROR CODES

Refer to OptoErr.RH for a list of error codes. OptoErr.H and OptoErr.BAS list some APIs that provide an error message given an error code