
GALIL OPTIMA

Application Note

This document describes how to configure a Red Lion Controls operator interface terminal to allow communications with a Galil Optima. The communications protocol supports access to pertinent parameters. Please read this document carefully before attempting to configure communications with these devices.

Introduction

The configuration software has been designed to allow the user to enter commands in a manner that should be familiar to the user of the Galil Optima.

There are some idiosyncrasies in interfacing with the Optima, especially with regard to reading and writing item that use decimal points. Please be aware of the number of decimal places each item uses, and configure the display properly.

Communications Mode

The default configuration is:

Port 2 - RS232

19200 baud

8 bits

No Parity

1 Stop Bit

Axis Selection

In Comms Devices, one can select the Axis as an address. The valid choices are A through H, where A,B,C, and D are equivalent to X,Y,Z, and W, respectively. The driver automatically sets Echo to Off, when first encountering a new device.

One can also define multiple axis in a Comms Devices address using two to eight letters, such as BD, in order to send a value to multiple axes for those commands that support such an instruction. E.g. the programmer might address Device 1 as A, Device 2 as B, and Device 3 as BD. Then by writing a value using the AC command to Device 3, the driver will send that value to both B and D in one instruction, without affecting the acceleration setting for axes A and C.

The complementary read operation performed on a device with multiple axes selected will read the value from the axis with the letter closest to the beginning of the alphabet, that is, if the programmer defines a device address as EDG, only axis D will be read for that device selection.

Certain commands, such as Tell Position (TP) require an axis designator be included with the command. These commands will ignore the Address setting in Comms Devices, and use the programmed selection.

Accessing Data

The programmer selects the desired parameter, either via the drop down list, or by typing in the acronym. For some commands, an axis letter is required. For the programmers convenience, <R> has been appended to the description to indicate read-only items, [R/W] has been appended to indicate items that can be read, and written, and (W) for write-only items. Note that reading a (W) item will result in a 0 being returned.

Position Format and Variable Format

Position Format and Variable Format selections are provided to allow the programmer to adjust the number of decimal places as needed. The default setting of the controller, and the driver, is 6.4, six to the left and 4 to the right of the decimal point. If the controller setting is different, or a change is attempted by the driver, a read of PF and/or VF is required in order to match the format in the driver with that of the controller. When writing a value to either of them, only the least three significant decimal digits will be used. For example, if either Format is to have 3 decimal places, and 7 digits to the left, set the value to 73. While the controller supports formats with up to 14 characters (10 to the left/4 to the right), the driver cannot support numbers greater than 32 bits, (range is 2,147,483,647 to -2,147,483,648). Therefore the driver limits the programming of PF and VF to a total of 10 digits. A value of 10 to 100 is acceptable, and any number from 0 to 4 for the last digit is acceptable, just as long as the sum is less than 10. If a number is returned by the controller that exceeds the range of the driver, the driver will return the positive or negative limit (2,147,483,647 or -2,147,483,648).

Read Operations

Read operations will be performed only on one axis at a time.

Read only commands RL, RP, SC, TD, TE, TP, TS, TT, TV require an axis identifier. These commands ignore the address setting in Comms Devices.

Read only commands that address the controller directly, for example Tell Inputs (TI), also ignore the device address setting, so can be programmed just one time on any device.

Commands that are programmed to write one value to multiple axes, will request the value from the axis with the letter closest to the beginning of the alphabet. Examples are Acceleration (AC) and Increment Position (IP). See "Axis Selection" above.

If an item is read that has decimal places, the driver will correct the incoming value to match the number of decimal places. For example, if the programmer specifies PF as 53, and the drive sends 1.23, the driver will multiply the result by 10 so that the display is correct when 3 decimal places are shown, so as to match the PF. If the controller returns more decimal places, the driver will round the resulting truncated number. See "Position Format and Variable Format" above.

Write Operations

Two command instructions have been separated into two parts, each. They are:
Abort which comprises AB as Abort Motion and Program, and ABA as Abort Motion only, and,
Analog Feedback which comprises AF as Enable Digital Feedback and AFA as Enable Analog Feedback.

These instructions are executed by writing a non-zero value, and are axis independent.

ECAM Master Axis (EA) is a command that requires an axis selection be made when programming. This is executed by writing a non-zero value to the command.

Writes to any commands controlled by Position Format will use the Position Format. Writes to Variables will always use the Variable Format setting. All other writes will use the default settings which are listed below.

Commands using Position Format:

BL DE DP ER FL IP PA
PR RL TD TE TP

Commands with default decimal places, and the fixed number of places:

Instruction	Decimal Places
FA	.00
GR	.0000
IL	.0000
IT	.0000
KD	.00
KI	.00
KP	.00
OF	.0000
TL	.0000
TT	.0000

All other commands transmit the number with no decimal places.

The commands BURN (BN), BURN VARIABLES (BV), and RESET (RS), take longer to respond than the normal timeout for a response. The driver will not transmit anything else to the controller for several seconds while those are executing.

PROGRAMMING COMMANDS

Items that are designed specifically for use in the controller's internal program, change the mechanical configuration of the controller, change the controller's internal program or display lists of program item, have been omitted. Examples are After Distance (AD), Edit (ED), Configure (CN), and List Program (LS). But see Device Command, below, which provides some flexibility for the programmer.

VARIABLES

Ten read/write variables have been provided, YP through YY. The programmer uses them as desired. If more are needed, dimension an array using some other variable name, transfer the data into one of the ten provided, and use another of the ten for the index. An example program line would be:

YP = YourArray[YQ]

In effect, this will provide an "unlimited" number of user variables that the driver can access from the controller. Any decimal position variances from the Variable Format setting must be taken care of by the programmer.

Device Command

A "Device Command Response" has been provided for items that do not fit into the normal driver operation. This is a read-only value, defined as a string, and thus must be programmed in a separate Comms Block from all other items, if Comms Blocks are used. A string of up to 63 characters can be retrieved from the drive using Device Command, and the response will be stored in Device Command Response.

For example, if for some reason, the programmer wanted the Copyright command executed, the programmed Action would be:

```
DeviceCommand( 1, "CW ?", 0 )
```

Where the 1 is the line number of the Device in Comms Devices, and the 0 is the parameter that indicates a response is expected. If that parameter is not 0, the driver will not expect a response from the controller. This can be used to send data or commands to the controller. Refer to the Optima manual to create the string command. The driver will add the Carriage Return to the end, and retrieve the response. Device Command Response (YZ) will then contain the returned string, up to the first 63 characters. The programmer can manipulate the string data using the various functions available.

Knowledge of Unit Operation Is Assumed

In all cases, the simple principle of 'pass-through' is maintained: there is no attempt to validate a value in terms of the end use of the unit: both familiarity with the Drive functions and knowledge of system operation are assumed.

RS232 Connection

Paradigm RS232 Port	OPTIMA RS232 Port
Terminal 1	3
Terminal 2	2
Terminal 5	5

In addition, connect a jumper between 7 and 8 on the Optima, and a jumper between 3 and 4 on the Paradigm.