
IAI ROBOCYLINDER

Application Note

This document describes how to configure a Paradigm operator interface terminal to allow communications with an IAI Robocylinder. The protocol supports access to pertinent parameters. Please read this document carefully before attempting to configure communications with these devices.

Introduction

The EDICT-97 configuration software has been designed to allow the user to enter commands in a manner that should be familiar to the programmer of a Robocylinder.

Addressing Units

The Default Device Address is 0. The Device Address can be set, in Comms Devices, to any value between 0-9 and A-F.

Communications Mode

The default configuration for Edict97 is:

Port 3 - RS-485

9600 baud

8 bits

No Parity

1 Stop Bit

The Baud Rate for the driver is fixed at 9600, the default rate for the Robocylinder upon power up. Changes made in the configuration will not affect the actual driver communication speed.

Accessing Data

The programmer will select the desired function by using the drop-down list. Programmers experienced with the driver will be able to type in the name of the function, directly.

WRITE OPERATIONS

NOTE THAT SOME PARAMETERS WILL REQUIRE SCALING BASED ON THE CONTROLLER CONSTANTS AND THE SIZE OF THE LEAD SCREW, IF ENGINEERING UNITS ARE TO BE USED. IT IS THE RESPONSIBILITY OF THE PROGRAMMER TO REFER TO THE ROBOCYLINDER CONTROLLER MANUAL, AND THE ACTUATOR MANUAL, IN ORDER TO GENERATE THE PROPER NUMBER OF ENCODER COUNTS.

Q3 - Position Move

Position Move requires 2 parameters to be sent. The first is the Point Number, and this will be required in the top 16 bits of the value. The second is the Position number which will be placed in the lower 16 bits of the value. To program a Position Move to Point 1, Position number 3, set the value to $0x10000+3$.

o - Home Command

Set the proper value for the direction the motor is to move.

a - Absolute Move

Set the value to the desired number. Remember to convert to encoder counts.

v - VEL/ACC Command

For this command three writes are required. The first two will be to write the desired Velocity into the VEL selection, and the desired Acceleration into the ACC selection. Then write a non-zero to the VELACC location to execute the command. Remember to convert to encoder counts.

q - Servo ON/OFF

Write a non-zero to the value to turn the servo on, a zero to turn it off.

m - Increment Move

Set the value to the desired number. Remember to convert to encoder counts.

d - Stop Motion

Write a non-zero to the value to send the Stop Motion command.

WRITE DATA TO THE POINT TABLE

This command comprises 5 operations, 4 to set the values, 1 to execute. To execute the command, write a non-zero to the value POINT. But first, fill the following locations with the desired values.

Q1 - A->B Transfer

A->B Transfer requires 2 parameters to be sent. The first is the Point Number, and this will be required in the top 16 bits of the value. The second is the Position number which will be placed in the lower 16 bits of the value. To program a Position in Position number 3, set the value to $0x10000+3$.

T4 - Address Allocation

Write the desired address, in Hex, to this location. E.g. Velocity is 0x404.

W4 - Data Write

Write the desired data to this location. Remember to convert to encoder counts.

V5 - B->A Transfer

Program this similarly to Q1 - A->B Transfer, above.

When a non-zero is written to POINT, the 4 commands will be sent in the listed sequence. Repeat the sequence for each additional data item.

READ OPERATIONS**n - Status Inquiry**

This location holds the raw data for Status, Alarms, IN, and OUT. The values for each separate item will be found in RSTATUS, RALARMS, RIN, and ROUT.

R4 - Position Inquiry

This location holds the value returned by the Position Inquiry command.

RSTATUS, RALARMS, RIN, and ROUT

These values do not send a request directly to the Robocylinder. Instead, they are updated whenever a Status Inquiry is made, or as a result of the response to any write command that returns that data.

Additional Read Information

Programming write operators for reading will return the most recent value written.

IMPORTANT:

The driver does not attempt to validate any parameter or data value. It is the total responsibility of the programmer to ensure that valid parameters and data values are transmitted.

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 Robocylinder functions and knowledge of system operation are assumed.

RS485 Connection

| Paradigm RS485 Port | ROBOCYLINDER |
|----------------------------|---------------------|
| Terminal 6 (TxA) & 8 (RxA) | 1 |
| Terminal 7 (TxB) & 9 (RxB) | 2 |
| Terminal 10 (Comm.) | No Connection |

In addition, connect a 1K resistor between 9 and 10 of the Paradigm.

Connect 4 and 8 on the Robocylinder controller to disable E-Stop.