

---

# **INDRAMAT ECODRIVE**

## **Application Note**

---

This document describes how to configure a Red Lion operator interface terminal to allow communications with an Indramat Ecodrive. 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 Ecodrive. Please note that the configuration software permits the selection of parameters that might not be available in the drive. It is the responsibility of the programmer to ensure that only valid parameter numbers are selected.

## Communications Mode

The default configuration for Edict97 is:

Port 2 - RS232

19200 baud

8 bits

No Parity

1 Stop Bit

## Accessing Data

The programmer selects the desired parameter, either via the drop down list, or by typing in the prefix, followed by the address in decimal. Note that an address can be selected that might not exist in the model being programmed. It is up to the programmer to verify the requested address exists.

**IMPORTANT NOTE:** The driver has been designed to permit the transfer of Position, Velocity, Acceleration, and Jerk List data, and access to individual numeric data items. Other than the lists mentioned, any item that returns more than 4 bytes (32 bits) of data such as the IDN lists of operation data, or non-numeric values such as the Manufacturer Version, will likely result in incomprehensible displays, or slowed communications. It is assumed that the drive configuration is known by the programmer so that these types of values are unneeded. See below for special instructions on writing parameters to Lists.

## Supported Formats:

### Standard Parameters from 0 to 4095

"S" representing the current value. These may be selected as Read/Write, although not all can necessarily be written.

"nS" representing the minimum value. These are read only.

"xS" representing the maximum value. These are read only.

### Product Specific Parameters from 0 to 4095

"P" representing the current value. These may be selected as Read/Write, although not all can necessarily be written.

"nP" representing the minimum value. These are read only.

"xP" representing the maximum value. These are read only.

**Block Parameter Lists - up to 64 values allowed**

"4006-01" Process Block Position List

"4007-01" Process Block Velocity List

"4008-01" Process Block Acceleration List

"4009-01" Process Block Jerk List

**Writing List Parameters**

The protocol requires that the complete List be written, even if only one value is changed. The best way to ensure this is done correctly is to set up a Communications Block with Access set to Both, and Update set to Manual, for each Process Block List. Then import the list values, either by a User Program that loads the communications block with the correct number of initial values, or by a ReadBlock() on a list that is already set up, perhaps by the Ecodrive configuration software. In the latter case, the programmer will most likely prefer to use the System Initialised event to perform the ReadBlock() Action. After that, unless the list can be changed by something other than the panel, it should not be necessary to read the list again. Another ReadBlock() can be performed at any time to refresh the data, if necessary.

When an item in the list needs to be changed, write the data to the communications block position, and perform a WriteBlock() function. On character based units, that WriteBlock() can be the Action associated with the event "Enter (Raise, Lower) Key Pressed". On a touch screen, an Accept selection should be provided, either a Soft-Key press or a button on the display. Writing a list that is shorter than needed will cause the drive to reset its list size to the new value, eliminating access to list parameters beyond that point.

For items that do not belong to the Position List, Velocity List, Acceleration List, or Jerk List, the above procedure is not needed.

**Command Error Selection**

For diagnostic purposes, "Z1" is a selection that will contain the most recently returned error value. Refer to the Ecodrive manual for information on the error code meanings.

**Decimal Point Handling**

Parameters using decimal points, are assigned as defined in document

**DOK-ECODR3-SMT-02VRS\*\*-PA01-EN-P**

A decimal point is inserted in the value written to the driver, at the position indicated. Items that use the Position Scaling parameters ( S-76, S-78, S-79 ), the Velocity Scaling parameter ( S-46 ), and the Acceleration Scaling parameter (S-162), automatically read the appropriate parameters before placing the decimal point in the value to be written. Items with fixed decimal positions defined in the above document use that value. For example, P-132, a parameter that depends on Position Scaling, is to be written, and the driver is sent the value 12345. S-76 will be read to determine if no scaling, linear scaling, or rotational scaling is required. If one of the latter, S-78 or S-79 will be read to determine the proper position. If 4 decimal places would be required, 1.2345 will be sent as the value. The programmer should be aware of the current decimal point setting in the drive, and adjust

the value by the appropriate power of 10, when necessary. As another example, S-100 is fixed at 1 decimal position. So a proportional gain of 8 must be sent to the driver as 80, so that 8.0 is transmitted to the drive.

All Program Block List items are written with the same number of decimal positions which will be determined before the first one is written. The programmer must adjust all values to the same resolution.

Read operations ignore any decimal points in the received numeric string. The programmer should know how to display the value appropriately.

### **IMPORTANT PROGRAMMING CONSIDERATION**

The Ecodrive protocol, except for Program Block Lists, requires a separate request for each parameter. When Communication Blocks are used in Edict97, the programmer assigns a parameter number, and a quantity. The selected range is permitted to span unassigned parameters, but a request will be issued, and an invalid IDN error will be returned by the drive. Similarly, if Automatic Blocks are used in Edict97, the configuration software will attempt to group parameters in a block of 16, when possible. For example, if a programmer sets a value to [S349], and a second to [S360], the Automatic Block will generate 12 read operations, S349 through S360. This will result in a number of invalid ID number errors, and will tend to slow communications. The programmer will generally have more control if Communication Blocks are used. Minimizing the number of read operations results in the fastest possible throughput of data.

### **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 to Indramat IKB0005 cable.**

<b>RS232 Port</b>	<b>ECODRIVE</b>
Terminal 1 - Tx	3 - Rx
Terminal 2 - Rx	2 - Tx
Terminal 5 - 0V	5 - 0V

In addition, connect a jumper between 3 and 4 of the Paradigm.

The Red Lion part P895047Z may also be used to connect to the Indramat IKB0005 cable.

#### Revision History

1 - 13 Nov. 2002 - Added Instructions for writing list parameters.