

INDRAMAT CLC DRIVE

Information Sheet for Crimson v2.0

Compatible Devices

- Indramat CLC Drive

Verified Device

- DKS01.1 – W030B-D

Driver Options

None

Device Options

A Network Address is selectable from 0 - 15.

Important Information

The Maximum value displayed in the selection dialog box does not imply the actual number of accessible points for any particular device. The programmer must be aware of the limits of the device that is to be connected.

No attempt is made to verify any data value being written. It is the responsibility of the programmer to ensure valid data are written.

Note: The driver verifies that the response to any command matches the network address, the class, the subclass, the set or bit number, the address, and checksum, before returning any data of a response with no error.

The following table uses these conventions:

pppp	a parameter number
ss	a one or two digit set number
bb	a one or two digit bit number
n	a single digit integer value

The programmer selects the values in the range specified by the minimum and maximum in the configuration dialog box.

Accessible Data

Prefix	Description	Format	Notes
VI	Integer Parameters	VIpppp(ss)	1
VR	Real Parameters	VRpppp(ss)	2
GI	Global Integer Parameters	GIpppp(ss)	1
GR	Global Real Parameters	GRpppp(ss)	2
IW	I/O Register Words	IWpppp	3
IB	I/O Register Bits	IBpppp(bb)	4
AX	Axis Parameters	AXpppp(ss)	2
CD	Card Parameters	CDpppp(ss)	2
TP	Task Parameters	TPpppp(ss)	2
SC	SERCOS Drive Parameters	SCpppp(ss)	2
ERR	Error Text	ERRn	5
TWC	User Write Command Text	TWCn	6
TXW	Send User Write Command	TXW	8
TRC	User Read Command Text	TRCn	6
TXR	Send User Read Command	TXR	8
RXD	User Command – Read Response	RXDn	7

Note 1: These items handle integer data in the range –2147483648 to 2147483647.

Note 2: These items handle real number data. Not all parameters accessed by these selections are real. For a read, the driver will convert response values to real number displays. For a write, the driver determines the data type of the target parameter, and converts the real data into the proper format.

Note 3: IW writes 16 bits of I/O data to the register specified.

Note 4: IB will read and write the one register bit specified in the configuration. Write 1 to set the bit, 0 to clear it.

Note 5: ERR0-ERR9 These items contain the text string most recently returned by the drive if an error occurs when attempting to access a parameter value. A write to any of these will clear the string. See Appendix A for details.

Note 6: TWC and TRC These items are included to permit the programmer to send commands and read data for items that are not provided for in the dialog box. Each selection holds 4 characters of a string to be sent to the drive, with the 0 selection being the first characters sent. A maximum of 32 characters may be used. See Appendix B for details.

Note 7: RXD Each **RXD[n]** contains 4 characters of the string of the data returned from the drive in response to a **TRC** command. The programmer must provide the interpretation if necessary. See Appendix B.

Note 8: TXW and TXR These items execute the sending of the corresponding command string when a 1 is written to it. When read, a zero is returned for each.

Appendix A – The Error Response

The drive will return an error string if a command cannot be executed. Each **ERR[n]** will contain 4 characters of the string.

For example, a request for network address 1, axis 0, parameter 0 would be:

```
>1 AP 0.0<CR><LF>
```

and the response would be:

```
>1 AP 0.0 !52 Invalid axis (0) $xx<CR><LF>
```

52 is the drive's error number, and xx is the checksum.

The text in **ERR[]** starts with the network address and includes up to the character before the \$ (or <CR> if the checksum is turned off).

The following table shows the data in **ERR[0]** through **ERR[7]** for the above example.

ERR[0]	'1' (0x31)	SP (0x20)	'A' (0x41)	'P' (0x50)
ERR[1]	SP (0x20)	'0' (0x30)	'.' (0x2E)	'0' (0x30)
ERR[2]	SP (0x20)	'!' (0x21)	'5' (0x35)	'2' (0x32)
ERR[3]	SP (0x20)	'I' (0x49)	'n' (0x4E)	'v' (0x76)
ERR[4]	'a' (0x61)	'l' (0x6c)	'i' (0x69)	'd' (0x64)
ERR[5]	SP (0x20)	'a' (0x61)	'x' (0x78)	'i' (0x69)
ERR[6]	's' (0x73)	SP (0x20)	'(' (0x28)	'0' (0x30)
ERR[7]	')' (0x29)	SP (0x20)	0	0

Assign to **ERR[0]** a string variable of size 40, with Encoding set to Packed High-To-Low. This will display up to 39 characters of the error string (the last character is always set to 0).

Appendix B – Formatting User Commands

A thorough knowledge of the command/data structure is necessary as the driver does not process the text in any way. The driver automatically generates the header, adds the entered **TRC** or **TWC** string, and then adds the tail.

Read Command – The command data entered into the TRC buffers must take the following form:

The high byte of **TRC[0]** is the hex value of the letter representing the Class.

The next higher byte is the hex value of the Subclass.

The next higher byte is hex 20 (space).

The low byte of **TRC[0]** is the first digit of the set, bit, or handle, number.

Example: To read User Integer Variable 1234, handle 0, the programmer-entered string is:

IP 0.1234

TRC[0] – **TRC[2]** should hold the following values:

	High Byte	Bits 23 – 16	Bits 15 - 8	Low Byte
TRC[0]	'I' (0x49)	'P' (0x50)	SP (0x20)	'0' (0x30)
TRC[1]	'.' (0x2E)	'1' (0x31)	'2' (0x32)	'3' (0x33)
TRC[2]	'4' (0x34)	0	0	0

To transmit the string, set the value of **TXR** to 1.

The full response from the unit would be (Data = 56789, xx = checksum):

>0 IP 0.1234 56789 \$xx<CR><LF>

RXD[0] will contain 0x35363738 ("5678").

RXD[1] will contain 0x39200000 ("9 "), the data followed by the space, then 0's.

A string variable of size 32 assigned to **RXD[0]**, with Encoding set to Packed High-To-Low will display up to 32 characters of the response.

Write Command – The command data entered into the **TWC** buffers must take the following form:

The high byte of **TWC[0]** is the hex value of the letter representing the Class.

The next higher byte is the hex value of the Subclass.

The next higher byte is hex 20 (space).

The low byte of **TWC[0]** is the first digit of the set or bit number.

Example: The programmer-entered string to set User Integer Variable 1234 in set 0 to 98765 is:

IP 0.1234 98765

The programmer will fill **TWC[0]** – **TWC[3]** with the following values:

	High Byte	Bits 23 - 16	Bits 15 - 8	Low Byte
TWC[0]	'I' (0x49)	'P' (0x50)	SP (0x20)	'0' (0x30)
TWC[1]	'.' (0x2E)	'1' (0x31)	'2' (0x32)	'3' (0x33)
TWC[2]	'4' (0x34)	SP (0x20)	'9' (0x39)	'8' (0x38)
TWC[3]	'7' (0x37)	'6' (0x36)	'5' (0x35)	0

To transmit the string, set the value of **TXW** to 1.

The full response from the unit would be:

>0 IP 0.1234<CR><LF>

RXD[n] remains unchanged after any write.

Cable Information

RS-422

FROM RLC UNIT	Serial Port
1 (TXB)	5
2 (TXA)	6
3 (RXA)	4
4(RXB)	1
6 (0V)	7

RS-485

FROM RLC UNIT	Serial Port
1+4 (RXB+TXB)	4
2+3 (RXA + TXA)	5
6 (0V)	7

RS-232

FROM RLC UNIT	Serial Port
2 (RxD)	2 (TxD)
5 (TxD)	3 (RxD)
3/4 (0V)	7