KEB DIN66019II Protocol

Application Note

This document describes how to configure a Paradigm operator interface terminal to allow communications with a device using the DIN66019II Protocol. 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 a command mnemonic and number in a manner that should be familiar to the user of a device using the DIN66019II Protocol. The driver allows the exchange of data with the Drive.

Accessing Data

The programmer obtains the best results by selecting the desired services using the pull-down list in Communications Blocks. Once the format of a given command is known, the command string may also be typed in directly. All data fields are either 32 bit or 64 bit, depending upon the command. The format is S, the number of the service, and, when required, additional specification fields, separated from the Sn by a space.

Note that the Address is always specified in hexadecimal digits.

The available commands are:
S0 - Access a parameter, specifying a Set number and Address.
Example : Set 1, Address 123 is entered/displayed as S0 1.123

S2A - Access a parameter's Lower Limit, specifying the Address
S2B - Access a parameter's Higher Limit, specifying the Address
Example : Address 123 is entered/displayed as S2A 123 and S2B 123

S3 - Access a parameter's Default Value, specifying the Address
Example : Address 123 is entered/displayed as S3 123

S4A - Access a parameter's Characteristics 1, specifying the Address
S4B - Access a parameter's Characteristics 2, specifying the Address
Example : Address 123 is entered/displayed as S4A 123 and S4B 123

S5A - Access a parameter's Display Divisor/Multiplier, specifying the Address
S5B - Access a parameter's Display Offset/Flags, specifying the Address
Example : Address 123 is entered/displayed as S5A 123 and S5B 123

S16A- Access the Process Data Type 1 (2*32Bit), Value 1
S16B- Access the Process Data Type 1 (2*32Bit), Value 2
Example : S16A and S16B

S17A- Access the Process Data Type 2 (4*16Bit), Values 1/2
S17B- Access the Process Data Type 2 (4*16Bit), Values 3/4
Example : S17A and S17B

It is very important that Items S2, S4, S5, S16, and S17, be put in the communications block in the order SnA, then SnB. When a read is requested, the A command is executed, with the second 32 bit word saved for when the B data is requested. When a write is to be done, first write the A value, then the B. The driver waits for the B value to be written before sending both 32 bit words.

There is one additional command provided for debugging the program. That command is: ERR - Access the latest error. When the device returns an error code, instead of data, the parameter address for S0, or the command number (2, 3, 4, 5, 16, or 17) appear in the upper 16 bits of the ERR value, and the error number returned is in the lower 16 bits.

Unit Addressing

The Unit Address is entered as a decimal number from 0 to 255. 255 is treated as a broadcast address, 240 - 254 are treated as multicast addresses. If items for any of these addresses generate a read command, the value of 0 will be returned.
Verify the valid addresses for the device to be used.

Responses

To clarify what happens with each successful command:

S0 and S3 - reads/writes the single 32 bit Default Value for the specified Address.

For S2 (with S4, S5, S16, and S17 working in the same way) :

Read - When S2A is read, the driver sends the command for Service 2. The device returns 16 hexadecimal digits. The driver returns the 32 bit value represented by the first 8 digits, and saves the value of the second 8 digits. When S2B is read, that value is retrieved from the driver without sending a second command to the device.

Write - When S2A is written, the driver stores the value, and waits for a write operation to be done on S2B. When a write is performed on S2B, the driver retrieves the S2A value, and sends the 16 hexadecimal digits in one command.

NOTE : Each A and each B command has one internal storage location, regardless of what address is chosen. For Services 16 and 17, this will not be any problem, as there is no address associated with those. For Services 2, 4, and 5, however, the programmer will have to process just one pair on any single communications scan, in order to read or write the proper value. As it is not likely these commands will be changing, or need to be changed often, it only requires care to copy the read data into another location before reading another, or writing just one pair per communication scan.

MAPPING Parameter Addresses

The F5 Combiviss Group List selections map to Parameter values as follows:

'sy' ---> 0xx	'dr' ---> 6xx	'di' ---> Bxx
'ru' ---> 2xx	'cn' ---> 7xx	'do' ---> Cxx
'op' ---> 3xx	'ud' ---> 8xx	'le' ---> Dxx
'pn' ---> 4xx	'fr' ---> 9xx	'in' ---> Exx
'uf' ---> 5xx	'an' ---> Axx	'cs' ---> Fxx

Remember that the Group List address must be converted to Hex, e.g. SO 1.210 selects ru16, cn12 is chosen by SO 1.70C.

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. Requesting an invalid parameter number may cause a failure to accumulate data from valid addresses.

Communications Connections

Paradigm Terminal	**KEB F5 Inverter- DB9**
Pin 1 (RS232 Tx)	3
Pin 2 (RS232 Rx)	2
Pin 3 (RTS)	
Pin 4 (CTS)	
Pin 5 (Common)	7
Pin 6 (RS485 TxA)	4
Pin 7 (RS485 TxB)	5
Pin 8 (RS485 RxA)	8
Pin 9 (RS485 RxB)	9
Pin 10 (Common, if necessary)	7

DO NOT CONNECT either Pin 1 or Pin 6 of the KEB's DB9 connector to anything on the Paradigm.

Using RS232, a link must be fitted between Pin 3 (RTS) and Pin 4 (CTS) on the Paradigm.

Using RS485, a 1K resistor between Pin 9 (RxB) and 10 (Common) on the Paradigm may sometimes be required.