Bulletin 1203-Gk5
Bulletin 1336-Gm5
Single Parameter Read And Write

APPLICATION NOTE **JUNE 24, 1997**

PURPOSE

The purpose of this document is to provide guidelines for wiring and control schemes for SCANport devices including Bulletin 1305 and 1336 PLUS AC Drives. This document is a suggestion only. Users must ensure that installations meet applicable codes and are suitable for the existing conditions.

WHAT THIS NOTE CONTAINS

This document contains information and examples of a PLC-5 ladder program to perform single parameter reads and writes using the 1771-SDN and 1203-GK5 or 1336-GM5. For information on reading or writing multiple parameter values and links refer to the application note titled “Scattered Read and Write”.

INTENDED AUDIENCE

This application note should be used by personnel familiar with the hardware components and programming procedures necessary to operate DeviceNet and SCANport devices. It is also assumed that the user has some familiarity with the PLC-5, 1771-SDN scanner and ladder programming.

WHERE IT IS USED

The diagrams, parameter settings and auxiliary hardware used in this application note are designed to address specific issues in many different applications. Some changes by the user may be necessary to apply the concepts of this document to a specific application.

TERMS AND DEFINITIONS

BTR - Block Transfer Read

BTW - Block Transfer Write

Datalink - a pointer used by some SCANport devices to allow parameters to be mapped to adapter I/O.

EEPROM - the memory that stores drive parameters when power is not applied.

Explicit Messaging - a DeviceNet messaging standard.

Parameter Read - a method of reading a single parameter.

Parameter Write - a method of writing a single parameter.

Scattered Read - a method of reading multiple parameters or links in any order.

Scattered Write - a method of writing multiple parameters or links in any order.

DESCRIPTION

These examples use a PLC-5, a 1771-SDN DeviceNet scanner and a 1203-GK5 to read and write parameter values in a SCANport device. For clarity, only one transaction is active at any time in these examples.

The first program will execute a BTW and a BTR under manual control to perform a parameter read or write depending on the information contained in the BTW data file.

The second program will execute a BTW and then execute multiple BTR’s until the DeviceNet message completes.

Complex applications may require additional block transfers to poll the DeviceNet scanner to check for message completion and may have multiple active transactions.

APPLICATION CONSIDERATIONS

These example ladder programs were written to be simple and clear examples of DeviceNet messaging. They contain no error handling. Consult the PLC-5, 1771-SDN and 1203-GK5 manuals for more information.

Explicit messages will complete faster if the 1771-SDN scanner is first placed in the idle mode. This may be worthwhile if the application requires reading or writing a large number of parameters in the SCANport device (e.g., configuring a drive system for a different product).

Using explicit messaging to make frequent changes to a parameter will eventually result in the failure of the SCANport device's EEPROM (if so equipped). If an application requires frequent changes of only a few parameters, the parameters should be written using the adapter's Datalink function since this does not cause EEPROM writes to occur.

PARAMETER OBJECT

The 1203-GK5 and 1336-GM5 DeviceNet to SCANport bridges with FRN1.6 or later firmware can read a parameter from any SCANport product with a single DeviceNet message. The message uses the standard DeviceNet Parameter Object which has a class code of 15 (0Fh).

The 1203-GK5 and 1336-GM5 support the following DeviceNet messages:

Service	Class	Instance	Attribute	Data	Description
0Eh	0Fh	0	Attrib #		Read Attribute specified for the Parameter Object
0Eh	0Fh	Parm #	Attrib #		Read Attribute specified for the Parameter specified
10h	0Fh	Parm #	1	Value	Write a value to Parameter specified
01h	0Fh	Parm #	0		Read all Attributes for Parameter specified
4Bh	0Fh	Parm #	Value / Bit #		Read Enum String for specified value or bit number of specified Parameter

CLASS ATTRIBUTES

The Parameter Object has several pieces of information available. Each of these pieces of information is called an "attribute". Each attribute can be read individually (with the 0Eh service code).

Attribute Number	Description	Data Length in Bytes
2 (02h)	The highest parameter number in this device.	2
8 (08h)	Descriptor	2
9 (09h)	Configuration Assembly Instance (always returns 0 -- not used)	2
10 (0Ah)	Language ID	1

The definition of the bits in the Parameter Object Descriptor word are shown below.

Descriptor Bit	Description (when set)
0	Device contains parameters
1	Supports full parameter attributes
2	Must do non-volatile storage save command
3	Parameters are stored in non-volatile storage
4-15	Reserved

The Parameter Object Configuration Assembly Instance is not supported and returns a value of zero.

The Parameter Object Language ID values are as shown below.

Language ID	Description
0	English
1	French
2	Spanish (Mexican)
3	Italian
4	German
5	Japanese
6	Portugese
7	Mandarin Chinese

PARAMETER ATTRIBUTES

Each parameter has a number of attributes. Each attribute can be read individually (with the 0Eh service code) or all at once (with the 01h service code).

Attribute Number	Description	Data Length in Bytes
1 (01h)	Parameter Value	2
2 (02h)	Link Path Size	1
3 (03h)	Link Path	0
4 (04h)	Descriptor -- see table	2
5 (05h)	Data Type -- see table	1
6 (06h)	Data Size -- number of bytes in Parameter Value	1
7 (07h)	Parameter Name String (length byte plus 16 characters)	17
8 (08h)	Units String (length byte plus 4 characters)	5
9 (09h)	Help String (length byte plus 0 characters)	1
10 (0Ah)	Minimum Value	2
11 (0Bh)	Maximum Value	2
12 (0Ch)	Default Value	2
13 (0Dh)	Scaling Multiplier -- see scaling formula	2
14 (0Eh)	Scaling Divisor -- see scaling formula	2
15 (0Fh)	Scaling Base -- see scaling formula	2
16 (10h)	Scaling Offset -- see scaling formula	2
17 (11h)	Multiplier Link -- parameter containing multiplier value	2
18 (12h)	Divisor Link -- parameter containing divisor value	2
19 (13h)	Base Link -- parameter containing base value	2
20 (14h)	Offset Link -- parameter containing offset value	2
21 (15h)	Decimal Precision -- see scaling formula	1

The definition of the bits in the Parameter Descriptor word are shown below.

Descriptor Bit	Description (when set)
0	Link path can be set
1	ENUM
2	Supports scaling
3	Supports scaling links
4	Read only
5	Updated in real time by device
6	Supports extended precision scaling
7-15	Reserved

The Data Type attribute for each parameter is defined below.

Data Type Value	Description
1	16-bit word
2	16-bit unsigned integer
3	16-bit signed integer
4	Boolean
5	Short integer
6	Double integer
7	Long integer
8	Unsigned short integer
9	Unsigned double integer
10	Unsigned long integer
11	Single floating point (IEEE 754)
12	Double floating point (IEEE 754)
13	Duration (short)
14	Duration
15	Duration (high-resolution)
16	Duration (long)
17	Date
18	Time of day
19	Date and Time
20	String (8-bit characters)
21	String (16-bit characters)
22	String
23	Short String
24	Byte (8-bits)
25	Double word (32-bits)
26	Long word (64-bits)

There are four scaling formulae -- two for use with extended precision scaling and two for normal scaling. The decimal precision variable is always used to locate the decimal point for a display by counting from the rightmost digit. In extended precision scaling the decimal precision variable is also used in the scaling formula.

$$\text{Engineering Value} = \frac{(\text{Internal Value} + \text{Offset}) * \text{Multiplier} * \text{Base})}{\text{Divisor} * 10^{\text{Decimal Precision}}}$$

$$\text{Internal Value} = \frac{\text{Engineering Value} * \text{Divisor} * 10^{\text{Decimal Precision}}}{\text{Multiplier} * \text{Base}} - \text{Offset}$$

$$\text{Engineering Value} = \frac{(\text{Internal Value} + \text{Offset}) * \text{Multiplier} * \text{Base})}{\text{Divisor}}$$

$$\text{Internal Value} = \frac{\text{Engineering Value} * \text{Divisor}}{\text{Multiplier} * \text{Base}} - \text{Offset}$$

HOW TO FORMAT THE EXPLICIT MESSAGE TRANSACTION BLOCK

Ten 32-word transaction blocks within the scanner module are reserved for Explicit Message Program Control. The transaction blocks accommodate both the download of Explicit Message Requests and the upload of Explicit Message Responses.

The scanner module can accommodate one request or response for each transaction block and can transfer two blocks for each upload and download. You must format each transaction block as shown:

	Format of 64-word Block Transfer Write for Explicit Message Request		Format of 64-word Block Transfer Read for Explicit Message Response		
	15 ... 0		15 ... 0		
Transaction #1 Header (3 words)	TXID	COMMAND	TXID	STATUS	word 0
	PORT	SIZE	PORT	SIZE	
	SERVICE	MAC ID	SERVICE	MAC ID	
	CLASS		SERVICE RESPONSE DATA		
	INSTANCE		"		
	ATTRIBUTE		"		
	SERVICE DATA		"		
	"		"		word 31
Transaction #2 Header (3 words)	TXID	COMMAND	TXID	STATUS	word 32
	PORT	SIZE	PORT	SIZE	
	SERVICE	MAC ID	SERVICE	MAC ID	
	CLASS		SERVICE RESPONSE DATA		
	INSTANCE		"		
	ATTRIBUTE		"		
	SERVICE DATA		"		
	"		"		word 63

Transaction Blocks are divided into two parts:

- **transaction header** - contains information that identifies the transaction to the scanner and processor
- **transaction body** - in a request, this contains the DeviceNet Class, Instance, Attribute and Service Data portion of the transaction. In a response, this contains only the response message.

Each of the data attributes in the transaction header are one byte in length:

- **COMMAND** - for each download, a command code instructs the scanner how to administer the request:

Command Code	Description
0	Ignore transaction block (block empty)
1	Execute this transaction block
2	Get status of transaction TXID
3	Reset all client/server transactions
4-255	Reserved

-
- **STATUS** - for each upload, the status code provides the processor with status on the device and its response:

Status Code	Description
0	Ignore transaction block (block empty)
1	Transaction completed successfully
2	Transaction in progress (not ready)
3	Error - slave not in scan list
4	Error - slave off-line
5	Error - DeviceNet port disabled or off-line
6	Error - transaction TXID unknown
7	Unused
8	Error - Invalid command code
9	Error - Scanner out of buffers
10	Error - Other client/server transaction in progress
11	Error - could not connect to slave device
12	Error - response data too large for block
13	Error - invalid port
14	Error - invalid size specified
15	Error - connection busy
16-255	Reserved

- **TXID** Transaction ID - when you create and download a request to the scanner, the processor's ladder logic program assigns a TXID to the transaction. This is a one-byte integer in word 31 the range of 1 to 255. The scanner uses this value to track the transaction to completion, and returns the value with the response that matches the request downloaded by the processor. The ladder logic program monitors rollover and usage of TXID values.
- **SIZE** The size of the transaction body in bytes. The transaction body can be up to 29 words (58 bytes) in length. If the size exceeds 29 words, an error code will be returned.
- **PORT** The DeviceNet port where the transaction is routed. The port can be zero (Channel A) or one (Channel B).

- **MAC ID** The DeviceNet network address of the slave device where the transaction is sent. This value can range from 0 to 63. The port and MAC ID attributes coupled together identify the target slave device. The slave device must be listed in the scanner module's scan list and be on-line for the Explicit Message transaction to be completed.
- **SERVICE** The service attribute contains the service request and response codes that match the corresponding request for the TXID.

HOW THE PROCESSOR AND SCANNER MODULE MANAGE MESSAGES

Block transfer operations between the processor and the scanner always originate in the processor. The scanner module can only wait for the processor to download a transaction block to the module or request an upload of a transaction block from the module.

Once an Explicit Message Request transaction block is downloaded to the scanner module, a ladder logic program in the processor polls the scanner module for the transaction block containing the Explicit Message Response for that request. This is done by the processor with a Block Transfer Read on the scanner module. Depending on the network load, the scanner could take a few seconds to complete the request. When a response is loaded, bit 15 of the module status register is set to 1. The program may have to poll the scanner module a number of times before the scanner returns a Response Transaction Block.

SIMPLE PARAMETER OBJECT MESSAGE EXAMPLE

The PLC-5 program shown in Figure 1 performs all the parameter functions depending on the data table values. Operation is as follows:

1) I:001/00 is set true to enable a single block transfer write to transfer the command and data to the 1771-SDN.

2) The 1771-SDN sends a DeviceNet message to the 1203-GK5.

3) The 1203-GK5 decodes the message and communicates with the SMP-3 via SCANport to complete the command it received.

4) The 1203-GK5 sends a DeviceNet message back to the 1771-SDN.

5) I:001/01 is set true to enable one block transfer read to transfer the response and data from the 1771-SDN.

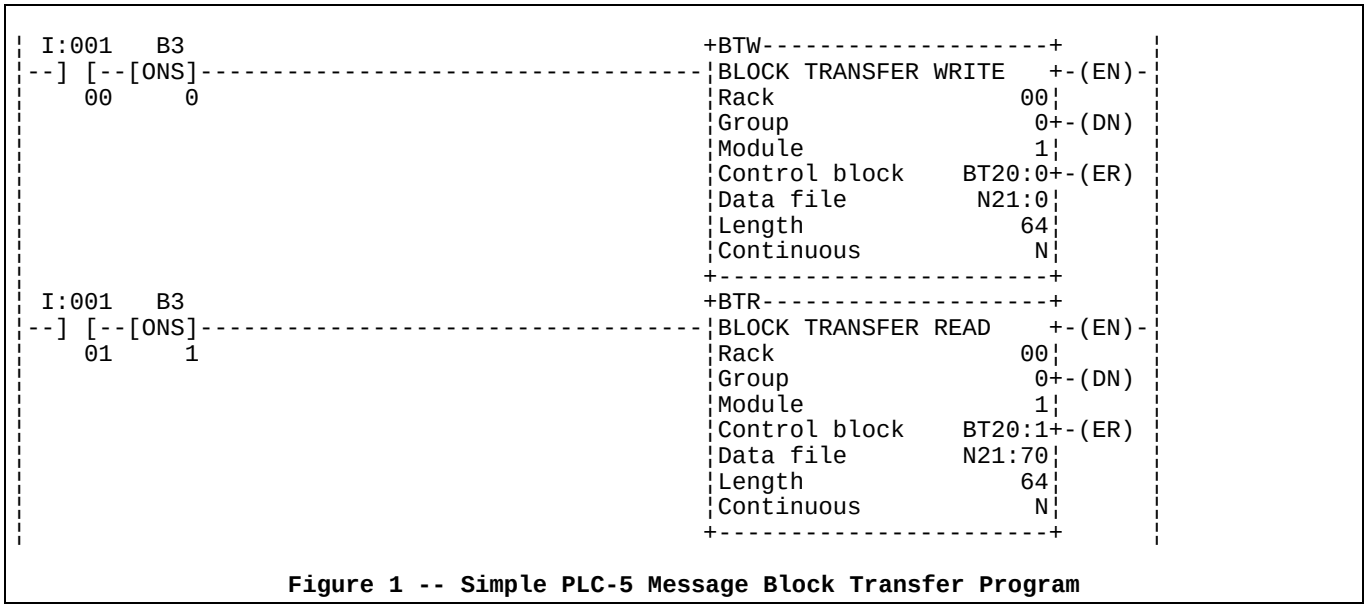

Figure 1 -- Simple PLC-5 Message Block Transfer Program

BLOCK TRANSFER DATA TABLES

Figures 2 through 13 show the data table values required to send DeviceNet messages to a 1336 PLUS and the response from the 1336 PLUS. The messages to the 1336 PLUS start at N21:0 and the response messages start at N21:70. All values are shown in hexadecimal unless identified otherwise.

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	0E01	000F	0000	0002	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	0002	8E01	00DB	0000	0000	0000	0000	0000	0000
N21:80	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 2 -- Read Maximum Instance Number

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	0E01	000F	0000	0008	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	0002	8E01	000F	0000	0000	0000	0000	0000	0000
N21:80	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 3 -- Read Parameter Class Descriptor

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	0E01	000F	0000	0009	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	0002	8E01	0000	0000	0000	0000	0000	0000	0000
N21:80	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 4 -- Read Configuration Assembly Instance

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	0E01	000F	0000	000A	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	0001	8E01	0000	0000	0000	0000	0000	0000	0000
N21:80	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 5 -- Read Language ID

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	0E01	000F	0005	0001	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	0002	8E01	0006	0000	0000	0000	0000	0000	0000
N21:80	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 6 -- Read Value of Parameter 7

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	0E01	000F	0005	0007	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	0011	8E01	4610	6572	2071	6553	656C	7463	3120
N21:80	2020	0020	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 7 -- Read Parameter Name String from Parameter 7

Address	0	1	2	3	4	5	6	7	8	9
N21:0	\01\01	\00\06	\0E\01	\00\0F	\00\05	\00\07	\00\00	\00\00	\00\00	\00\00
N21:10	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:20	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:30	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:40	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:50	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:60	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:70	\01\01	\00\11	Ä \01	F \10	e r		q	e S	e l	t c 1
N21:80		\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:90	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:100	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:110	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:120	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:130	\00\00	\00\00	\00\00	\00\00						

Figure 8 -- Read Parameter Name String from Parameter 7 (ASCII Display)
(ASCII strings appear byte-swapped when displayed with PLC programming software)

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0008	1001	000F	0005	0001	0007	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	0000	9001	0000	0000	0000	0000	0000	0000	0000
N21:80	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 9 -- Write a value of 7 to parameter 5

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	0101	000F	0005	0000	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	002E	8101	0007	0200	0200	0D02	7246	7165	5320
N21:80	6C65	6365	2074	0031	0100	1200	0600	0100	0100	0100
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 10 -- Read Full from parameter 5

Address	0	1	2	3	4	5	6	7	8	9
N21:0	\01\01	\00\06	\01\01	\00\0F	\00\05	\00\00	\00\00	\00\00	\00\00	\00\00
N21:10	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:20	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:30	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:40	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:50	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:60	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:70	\01\01	\00 .	ü \01	\00\07	\02\00	\02\00	\0D\02	r F	q e	S
N21:80	l e	c e	t	\00 1	\01\00	\12\00	\06\00	\01\00	\01\00	\01\00
N21:90	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:100	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:110	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:120	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:130	\00\00	\00\00	\00\00	\00\00						

Figure 11 -- Read Full from parameter 5 (ASCII Display)
(ASCII strings appear byte-swapped when displayed with PLC programming software)

Address	0	1	2	3	4	5	6	7	8	9
N21:0	0101	0006	4B01	000F	0005	0006	0000	0000	0000	0000
N21:10	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:20	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:30	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:40	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:50	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:60	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:70	0101	000A	CB01	4109	6164	7470	7265	3120	0000	0000
N21:80	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:90	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
N21:130	0000	0000	0000	0000						

Figure 12 -- Read ENUM for a value of 6 in parameter 5

Address	0	1	2	3	4	5	6	7	8	9
N21:0	\01\01	\00\06	K \01	\00\0F	\00\05	\00\06	\00\00	\00\00	\00\00	\00\00
N21:10	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:20	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:30	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:40	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:50	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:60	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:70	\01\01	\00\0A	\CB\01	A \09	a d	t p	r e	1	\00\00	\00\00
N21:80	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:90	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:100	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:110	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:120	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00	\00\00
N21:130	\00\00	\00\00	\00\00	\00\00						

Figure 13 -- Read ENUM for a value of 6 in parameter 5 (ASCII Display)
(ASCII strings appear byte-swapped when displayed with PLC programming software)

INTERLOCKED BTW/BTR PARAMETER READ AND WRITE EXAMPLE

Figure 14 shows one method of interlocking the block transfer write and read functions so that a message to the 1203-GK5 is automatically monitored for a response. This program uses the same data table values as shown in Figures 2 and 3.

When N21:70 no longer contains a value of zero the message has completed. If the value in N21:70 matches the value in N21:0 the message was successfully completed. At that time, the data table contains the response message and another message can be sent to the 1203-GK5. If the values do not match, an error has occurred.

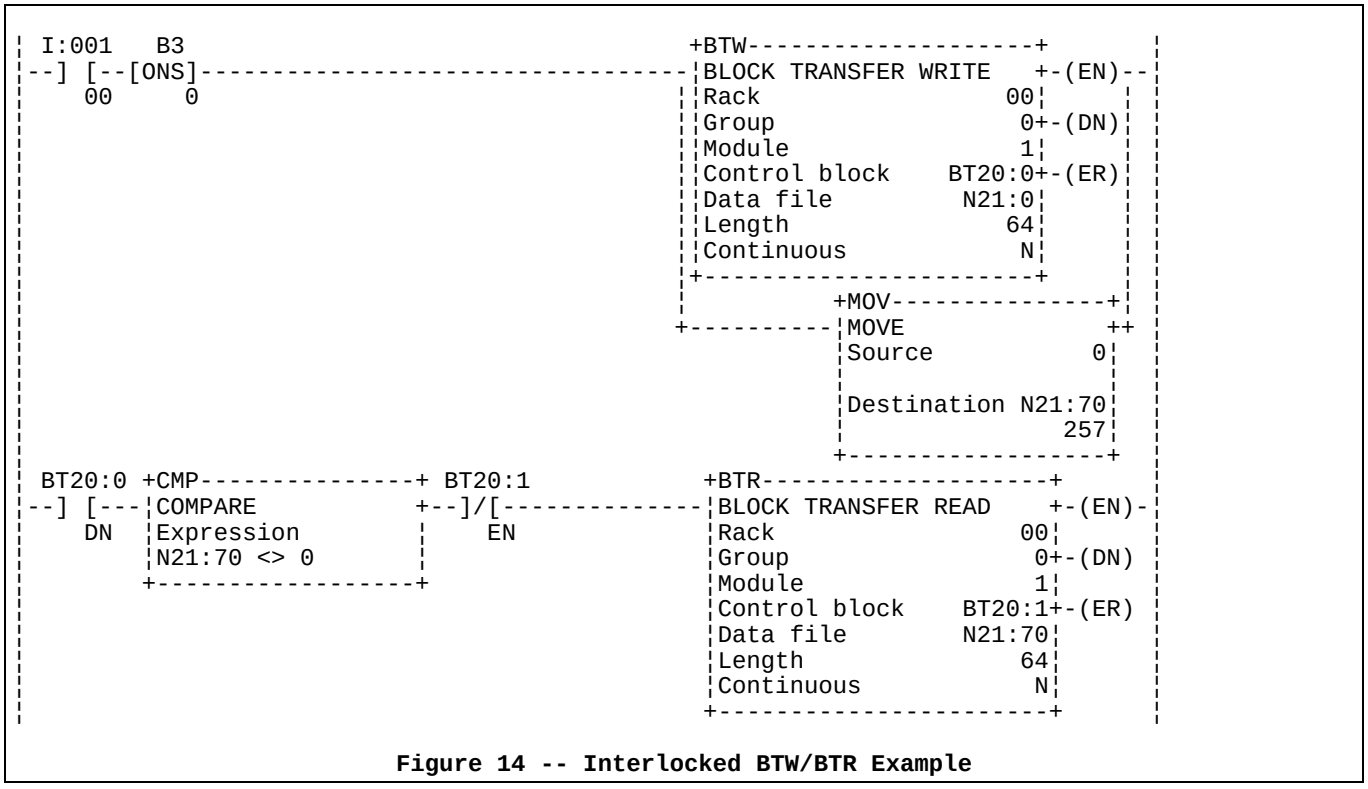

Figure 14 -- Interlocked BTW/BTR Example