Bulletin 1203-Gk5
Bulletin 1336-Gm5
Scattered 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 scattered reads and writes of parameter values or links using the 1771-SDN and 1203-GK5 or 1336-GM5. For information on reading or writing single parameters refer to the application note titled “Single Parameter 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 and links 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 scattered 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.

SCATTERED PARAMETER VALUE READ

The 1203-GK5 and 1336-GM5 DeviceNet to SCANport bridges with FRN1.8 or later firmware can read a group of scattered parameters from any SCANport product with a single DeviceNet message. The DeviceNet message uses vendor specific classes reserved for Allen-Bradley SCANport devices.

The DeviceNet class is 147(93H) for a parameter value read. The service code is 50(32H), which is a class specific service code. The instance and attribute codes are zero (access to the general class and not a specific instance).

The data for this command is a list of two words for each parameter to be read. The first word contains the parameter number. The second word is a place holder and has no required value. The format of DeviceNet command is:

DeviceNet Message Structure
Message Header, MAC ID, XID, Frag
RR=0, Service Code=50(32 hex)
Class = 147 decimal, 93 hex
Instance LSB = 0
Instance MSB = 0
Attribute = 0
First Parameter Number LSB
First Parameter Number MSB
Response data place holder LSB = 0
Response data place holder MSB = 0

The parameter list can contain up to 24 parameters. However, not all DeviceNet products can support messages this long. The maximum transmission length from the DeviceNet Server/Master may therefore be the limiting factor. (NOTE: This message requires implementation of the DeviceNet Fragmentation protocol.)

The response data is in a format similar to the command data. Each parameter requested is returned in two words. If the parameter read was successful the first word will contain the parameter number and the second word the parameter value. If the parameter read was unsuccessful the high bit of the parameter number in the first word will be set as an error flag and the second word will contain an error code. In most situations, this will be a value of 4 or "Instance Not Supported." Each parameter read can succeed or fail independently.

SCATTERED PARAMETER VALUE WRITE

FRN1.9 and later firmware can write a scattered list of parameters. The DeviceNet class is 147(93H) with a service code of 52(34H) (another object specific service). As with the scattered read, the data associated with the command consists of two words for each parameter. The first word contains the parameter number. The second word contains the parameter value to be written.

The response message is similar to the scattered read. If a write was successful, the data returned is zero. If a write was unsuccessful, the most significant bit of the parameter number in the first word is set and the second word contains an error code. The likely error responses are:

1	An attempt was made to write to a read only variable
4	Parameter number is out of range (Instance not supported).
6	Data to be written was out of range
7	Object State Conflict (e.g., an attempt to change a variable that is not changeable in the run state).

SCATTERED PARAMETER LINK READ OR WRITE

FRN1.9 and later firmware can also read and write a scattered list of parameter links in SCANport products that support parameter links. The scattered parameter link read/write function is the same as the scattered parameter read/write function except that the DeviceNet class is 153(99H) instead of 147(93H). The same object specific service codes are used.

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 SCATTERED PARAMETER READ AND WRITE EXAMPLE

The PLC-5 program shown in Figure 1 performs all the scattered read and write function 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.

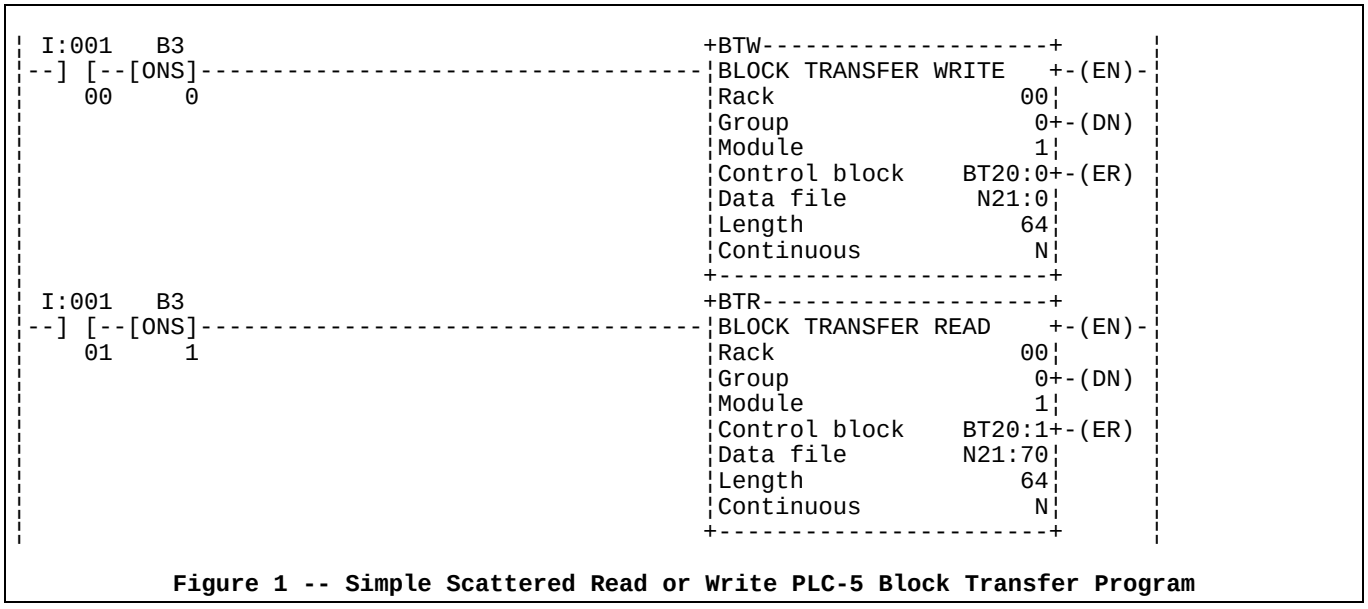

Figure 1 -- Simple Scattered Read or Write PLC-5 Block Transfer Program

BLOCK TRANSFER DATA TABLES

Figure 2 shows the data table values required to read a scattered list of parameters from the SMP-3. This example will read eight parameters.

Address	0	1	2	3	4	5	6	7	8	9
N21:0	257	38	12801	147	0	0	1	0	8	0
N21:10	2	0	7	0	3	0	6	0	4	0
N21:20	5	0	0	0	0	0	0	0	0	0
N21:30	0	0	0	0	0	0	0	0	0	0
N21:40	0	0	0	0	0	0	0	0	0	0
N21:50	0	0	0	0	0	0	0	0	0	0
N21:60	0	0	0	0	0	0	0	0	0	0
N21:70	257	32	-19967	1	0	8	3597	2	100	7
N21:80	4	3	0	6	4	4	0	5	100	0
N21:90	0	0	0	0	0	0	0	0	0	0
N21:100	0	0	0	0	0	0	0	0	0	0
N21:110	0	0	0	0	0	0	0	0	0	0
N21:120	0	0	0	0	0	0	0	0	0	0
N21:130	0	0	0	0						

Figure 2 -- Scattered Read Data Table Example

Figure 3 shows the data table values required to write a scattered list of parameters to the SMP-3. This example attempts to write eight parameters but only succeeds with parameter 7 (the other parameters are read-only).

Address	0	1	2	3	4	5	6	7	8	9
N21:0	257	38	13313	147	0	0	1	0	8	0
N21:10	2	0	7	0	3	0	6	0	4	0
N21:20	5	0	0	0	0	0	0	0	0	0
N21:30	0	0	0	0	0	0	0	0	0	0
N21:40	0	0	0	0	0	0	0	0	0	0
N21:50	0	0	0	0	0	0	0	0	0	0
N21:60	0	0	0	0	0	0	0	0	0	0
N21:70	257	32	-19455	-32767	1	-32760	1	-32766	1	7
N21:80	0	-32765	1	-32762	1	-32764	1	-32763	1	0
N21:90	0	0	0	0	0	0	0	0	0	0
N21:100	0	0	0	0	0	0	0	0	0	0
N21:110	0	0	0	0	0	0	0	0	0	0
N21:120	0	0	0	0	0	0	0	0	0	0
N21:130	0	0	0	0						

Figure 3 -- Scattered Write Data Table Example

Figure 4 shows the data table values required to read a scattered list of parameter links from a 1336 FORCE drive with a Standard Adapter board. This example will read seven parameter links.

Address	0	1	2	3	4	5	6	7	8	9
N21:0	257	34	12801	153	0	0	28	0	104	0
N21:10	370	0	373	0	376	0	101	0	379	0
N21:20	0	0	0	0	0	0	0	0	0	0
N21:30	0	0	0	0	0	0	0	0	0	0
N21:40	0	0	0	0	0	0	0	0	0	0
N21:50	0	0	0	0	0	0	0	0	0	0
N21:60	0	0	0	0	0	0	0	0	0	0
N21:70	257	28	-19967	28	355	104	358	370	146	373
N21:80	182	376	0	101	365	379	146	0	0	0
N21:90	0	0	0	0	0	0	0	0	0	0
N21:100	0	0	0	0	0	0	0	0	0	0
N21:110	0	0	0	0	0	0	0	0	0	0
N21:120	0	0	0	0	0	0	0	0	0	0
N21:130	0	0	0	0						

Figure 4 -- Scattered Link Read Data Table Example

Figure 5 shows the data table values required to write a scattered list of parameter links to a 1336 FORCE drive with a Standard Adapter board. This example will write seven parameter links.

Address	0	1	2	3	4	5	6	7	8	9
N21:0	257	34	13313	153	0	0	28	355	104	358
N21:10	370	146	373	182	376	0	101	365	379	146
N21:20	0	0	0	0	0	0	0	0	0	0
N21:30	0	0	0	0	0	0	0	0	0	0
N21:40	0	0	0	0	0	0	0	0	0	0
N21:50	0	0	0	0	0	0	0	0	0	0
N21:60	0	0	0	0	0	0	0	0	0	0
N21:70	257	28	-19455	28	0	104	0	370	0	373
N21:80	0	376	0	101	0	379	0	0	0	0
N21:90	0	0	0	0	0	0	0	0	0	0
N21:100	0	0	0	0	0	0	0	0	0	0
N21:110	0	0	0	0	0	0	0	0	0	0
N21:120	0	0	0	0	0	0	0	0	0	0
N21:130	0	0	0	0						

Figure 5 -- Scattered Link Write Data Table Example

INTERLOCKED BTW/BTR PARAMETER READ AND WRITE EXAMPLE

Figure 6 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.

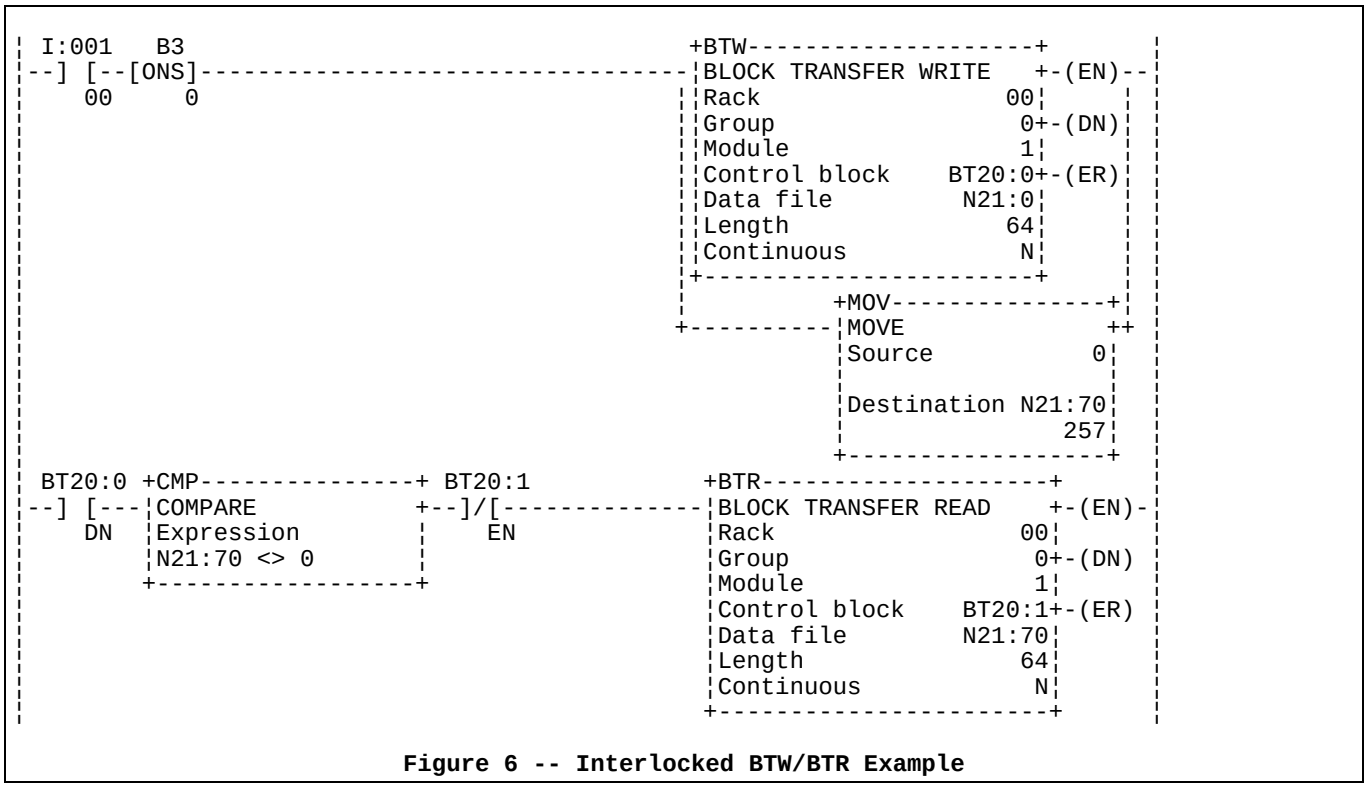

Figure 6 -- Interlocked BTW/BTR Example