

AB DF1

Module (Driver) Documentation

Author: Glenn Turnock

Date: 7-Aug-03

Revision: 0.1

Table of Contents

Overview	1
Capabilities and Functionality	1
Add Mappings.....	1
PLC-5 (Master)	1
SLC (Master)	1
Error Handling	1
Design.....	2
Communications Interface	2
Protocol.....	2
Network Interface	2
Server Interface.....	2
Functional Overview: Active Station List.....	2
Functional Overview: Network Reads	3
Functional Overview: Network Writes	3
Network Configuration And Mappings	4
Network Configuration Parameters	4
Mapping Parameters	5
Mapping Address Formatting Rules	5
Status Counters.....	6
“Hardware” Status Counters.....	7
Error And Notice Messages	8
Critical Error Messages.....	8
Error Messages	8
Notice Messages	9
DF1 and TX_Delay.....	10

Overview

This document provides design details and other information for the successful configuration and use of the AB-DF1 client module (driver) for the Pronghorn system. The main purpose of the client module is to provide connectivity to Allen-Bradley PLC's/Devices via the DF1 network. Details on this protocol are detailed in the "DF1 Protocol and Command Set" Reference Manual (Allen-Bradley publication 1770-6.5.16).

Capabilities and Functionality

The main objective of this project is to provide a full functioning ABDF1 Pronghorn client module. The objective of the first milestone was to develop an ABDF1 master module capable of communicating with a PLC5 device on an AB DF1 communications network. Additional DF1 devices can be supported by addition of new addressing and messaging functionality to the AB Code Libraries.

Add Mappings

As part of the startup process, a mapping .csv file is read and mappings are added to the Pronghorn Daemon task list. The client module must interpret all network specific mapping data. As part of the process the virtual function 'Add Mappings' will be called. The client module is required to interpret each external mapping item and add an internal mapping structure to the mapping list. Additionally an address link association is added, which is later used for determining which mappings need to be executed. This is required because a single internal (Pronghorn) address can be associated with multiple external addresses and vice versa.

PLC-5 (Master)

These are the requirements for the client module when running in this mode or communicating with this type of PLC. Both Network Writes and Network Reads are solicited from the slave device. To accomplish this the following DF1 commands are implemented:

- The commands required to communicate to a PLC-5 slave
 - a. Typed Read (CMD=0F, FNC=68): Read a block of data from a PLC address
 - b. Typed Write (CMD=0F, FNC=67): Write a block of data to the a PLC address

SLC (Master)

These are the requirements for the client module when running in this mode or communicating with this type of PLC. Both Network Writes and Network Reads are solicited from the slave device. To accomplish this the following DF1 commands are implemented:

- The commands required to communicate to a SLC-500 series PLC
 - a. Protected Read (CMD=0F, FNC=A2): Read data from a logical address on a SLC slave
 - b. Protected Write (CMD=0F, FNC=AA): Write data to a logical address on the a SLC slave

Error Handling

The client module must be able to accept and decipher error conditions from the slave nodes. Conversely it must be able to report exception status to the master node. The DF1 protocol handles this by using a STS (Status) byte in the command header and an EXT STS (Extended Status) code in the trailer if there is an error detected. An error table in the AB Libraries is used to map all the Allen-Bradley error codes to Pronghorn error codes and vice versa.

Design

The Pronghorn client module is to be named abdf1. All code conforms to the Pronghorn Object Oriented programming model, based on the sample code that was provided as a guideline. All code is modular and written in C++. The supplied Pronghorn library functions are used wherever possible.

Communications Interface

The DF1 network requires that a serial interface be used for the physical communications layer. The Pronghorn library contains a wrapper class to access the serial hardware interface that provides the following functions:

- 1) Configure: A method for client to pass configuration parameters (i.e. Station address, port, irq, baud rate etc)
- 2) Open: A method for opening the interface
- 3) Close: A method for closing the interface
- 4) Read: A method for the client module to read data from the interface
- 5) Write: A method for the client module to write data to the interface

The DF1 Module incorporates and makes use of this Pronghorn Library wrapper class to access the serial communications port.

Protocol

The DF1 Module makes use of a common set of Allen-Bradley functions/structures that exist in the Pronghorn common library.

Network Interface

A new C++ class (CAbDf1) has been created (derived from the serial API, CCOM) so that it handles:

1. DF1 command set
2. Packet assembly
3. Packet disassembly
4. Read/Write to the network interface

Server Interface

The Pronghorn Library contains a set of common library classes to provide the interaction between the AB DF1 Module and the Pronghorn Server Daemon. In addition, a set of functions has been implemented (*i.e.* DoNetWrite and DoNetRead *virtual functions*), which process commands from the Pronghorn Daemon.

Functional Overview: Active Station List

The DF1 module maintains a continuously updated Active Station List, which maintains a list of which stations on the DF1 network are currently active. On module startup, all stations are assumed to be active.

Each time a network transaction occurs, the Active Station List is consulted. If the station pertaining to the current transaction is considered active, then the transaction proceeds as normal. If the target station is considered to be inactive, a single Master Poll (DLE ENQ) message is sent to the slave. If the slave responds (with either a stale message, or a EOT), then the slave is considered active, the Active Station list is updated to reflect this fact, and the transaction proceeds. If, on the other hand, an error, or no reply is received within a valid time frame, the station is still considered to be inactive and the current transaction for this inactive slave is aborted with error.

The active station list also records details of the last transaction received from the particular slave. This is used to determine duplicate messages and also to handle stale messages in the event of command timeouts.

Functional Overview: Network Reads

When the Pronghorn Daemon requests that the DF1 Module perform a read operation on a DF1 Slave device, the Active Station List is consulted.

Once we determine that we are trying to communicate with an active station, a DF1 application layer packet is constructed, based on the target slave Station ID, and the requested address to access on the slave using common Pronghorn AB functions. Next, the DF1 data link layer frame is constructed by the 'CreateMasterMessageFrame' function. This function assembles the master message frame, computing the BCC/CRC as it goes, and also performs DLE stuffing on the data, if required.

The master message frame is transmitted to the target DF1 slave. What the master now does is look for an acknowledge reply (DLE ACK) from the slave. If an error occurs, or an acknowledge is not received within a valid time frame, then a retry count is incremented and the master message frame is retransmitted. If the number of retries exceeds a configured limit (NAK retries), then the message is aborted with error.

Once a successful acknowledge is received, then the Master must poll the slave to obtain the response message. The 'CreateMasterPollingFrame' function creates the ENQ message to be sent to the target slave. Once the polling frame is sent to the slave, the processing is much the same as when sending the master message frame, with one exception. Since we know we sent the slave a message, polling continues until we either receive a reply or until the transaction times out (cmd timeout). If the transaction times out it is likely that the slave is taking too long to process the reply. If an error occurs, or a reply message (either an EOT saying that there is no message for the slave to send, or a valid reply message) is not received within a valid time frame, then a retry count is incremented and the master polling frame is retransmitted. If the number of retries exceeds a configured limit (ENQ retries), then the message is aborted with error.

Once that we have determined that we have received a valid message frame (good BCC/CRC, etc) from the DF1 slave, the 'DispatchAbDf1Message' function is called. This function performs a check on the received message frame from the Slave to ensure that its contents are valid. Based on this, we can then unpack the reply using the common Pronghorn AB routines to determine the format and type of data that the reply message carries. Once this is determined, then the data can be transferred to Pronghorn shared memory.

Functional Overview: Network Writes

When the Pronghorn Daemon requests that the DF1 Module perform a write operation on a DF1 Slave device, the Active Station List is consulted.

Once we determine that we are trying to communicate with an active station, a DF1 application layer packet is constructed, based on the target slave Station ID, and the requested address to access on the slave using common Pronghorn AB functions. Next, the data, based on the Pronghorn request and the referenced mapping is copied from Pronghorn shared memory to a temporary transmit data buffer. Next, the DF1 data link layer frame is constructed by the 'CreateMasterMessageFrame' function. This function assembles the master message frame, inserting the data from the temporary transmit buffer, computing the BCC/CRC as it goes, and also performs DLE stuffing on the data, if required.

The master message frame is transmitted to the target DF1 slave. What the master now does is look for an acknowledge reply (DLE ACK) from the slave. If an error occurs, or an acknowledge is not received within a valid time frame, then a retry count is incremented and the master message frame is retransmitted. If the number of retries exceeds a configured limit (NAK retries), then the message is aborted with error.

Once a successful acknowledge is received, then the Master must poll the slave to obtain the response message. The 'CreateMasterPollingFrame' function creates the ENQ message to be sent to the target slave. Once the polling frame is sent to the slave, the processing is much the same as when sending the master message frame, with one exception. Since we know we sent the slave a message, polling continues until we either receive a reply or until the transaction times out (cmd timeout). If the transaction times out it is likely that the slave is taking too long to process the reply. If an error occurs, or a reply message (either an

EOT saying that there is no message for the slave to send, or a valid reply message) is not received within a valid time frame, then a retry count is incremented and the master polling frame is retransmitted. If the number of retries exceeds a configured limit (ENQ retries), then the message is aborted with error.

Once that we have determined that we have received a valid message frame (good BCC/CRC, etc) from the DF1 slave, the 'DispatchAbDf1Message' function is called. This function performs a check on the received message frame from the Slave to ensure that its contents are valid.

Network Configuration And Mappings

Network Configuration Parameters

The network configuration parameters are created using spreadsheets (i.e. Excel) that are then exported to a Comma Separated Values (CSV) file. The Pronghorn uses these files as a configuration interface to the system. There already exist library functions to handle this. New fields have been defined in order to configure the DF1 Client module and network interface. The DF1 network requires the following fields to be added to the network configuration structure:

Field Identifier	Valid Range	Default Value	Description
Station	[0 – 254]	5	The device (or Node) number that identifies the module on the AB-DF1 network
Mode	“Master”, “Slave” or “Peer”	“Master”	Specifies the AB-DF1 module operational mode
PLC_Type	“PLC2”, “PLC3”, “PLC5” or “SLC”	“PLC5”	The type of PLC that the AB-DF1 module is to emulate on the AB-DF1 network
Error_Code	“BCC” or “CRC”	“CRC”	The type of error-detection to be used on the AB-DF1 network. “BCC” specifies a simple 8-bit checksum is to be used for error-detection. “CRC” specifies that a 16-bit Cyclic-Redundancy-Check code is to be used for error-detection.
NAK_Retries	[0 – 10]	3	The maximum number of allowed failed attempts to be performed when transmitting the Master Command message to the slave.
ENQ_Retries	[0 – 10]	3	The maximum number of allowed failed attempts to be performed when transmitting the Master ENQ Poll message to the slave.
Cmd_Timeout	[500 – 5000] ms	1000 ms	The expected time that a complete command cycle (Command Send, Slave ACK, ENQ Poll, Slave Reply) will be completed.
Com_Port	[1 – 8 (or more depending on CCOM support)]	1	The serial communications port that this module will use
Baud	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	9600	The transmission speed of the DF1 Network, in bits per second.
Data_Bits	[5 – 8]	8	Specifies the number of data-bits per transmitted character.
Stop_Bits	[1 – 2]	1	Specifies the number of stop-bits per transmitted character.
Parity	“Odd”, “Even” or “None”	-	Specifies the type of parity checking performed

Field Identifier	Valid Range	Default Value	Description
Timeout	[0 – 360000] ms	1000 ms	Specifies the time, in ms, to wait for a response from a remote DF1 device.
TX_Delay	[0 – 65535] ms	0	A time delay in milliseconds that the driver will wait before transmitting data. Typically used only on half-duplex (2-wire) DF1 networks.
Handshake	“Yes” or “No”	“No”	Specifies if Hardware Handshaking (CTS/RTS) is being used

Mapping Parameters

Like the network configuration parameters, mappings are also created using spreadsheets (i.e. Excel) that are then exported to a Comma Separated Values (CSV) file. The Pronghorn uses these files as a configuration interface to the system. There already exist library functions to handle this. New fields have been defined in order to implement a mapping for accessing the DF1 data. There are already some common fields used by Pronghorn (Address, Command, Type and Tag). The DF1 network requires the following additional fields to be added to the mapping structure:

Field Identifier	Valid Range	Description
Station	[0 – 254]	The device (or Node) number that the Master will transmit commands to.
PLC_Type	“PLC2”, “PLC3”, “PLC5” or “SLC”	The type of PLC/Device that the DF1 Module will be communicating with (since the command sets differ for differing PLC Types)
Address	See Description	This field specifies the register address that the data is being read from (in the case of a read command) or being written to (in the case of a write command). See below for the common Address Formatting Rules.

Mapping Address Formatting Rules

There are a number of different possible address formats supported by each type of PLC. The address has the following common format, in general:

[File Type][File Number];[Element]/{[Sub-Element]}

where

File Type is a letter representing the type of PLC data. Valid file types and their associated identifiers are listed in the following table:

File Type Identifier	Default File Number	File Data Type
O	0	Outputs
I	1	Inputs
S	2	Status
B	3	Binary
T	4	Timer
C	5	Counter
R	6	Control
N	7	Integer
F	8	Floating Point

File Number is a numeric value representing the File in the PLC's memory. File Numbers 0-8 are reserved and are associated with the File Types as shown in the table, above. File Numbers 9-999 are user configurable.

Element is the offset of the required element to access in the File.

Sub-Element is an optional addressing field which may be required when accessing certain File Types.

Status Counters

The AB-DF1 Module maintains a list of status counters for recording the operational status of the module and the DF1 communications network. The following table lists the ID #, Mnemonic and a description of each status counter implemented in the AB-DF1 Module.

ID	Description	Mnemonic
2525	Message Packets transmitted – The sum of all Message Packets sent by this module	CNT_MSG_PACKETS_SENT
2526	Message Packets received – The sum of all Message Packets Received by this module	CNT_MSG_PACKETS_RCVD
2527	Command Messages transmitted – The number of AB-DF1 Command Messages sent to DF1 devices on the network	CNT_CMD_MSGS_SENT
2528	Command Messages received – The number of AB-DF1 Command Messages received from DF1 devices on the network	CNT_CMD_MSGS_RCVD
2529	Reply Messages transmitted – The number of AB-DF1 Reply Messages sent to DF1 devices on the network	CNT_REPLY_MSGS_SENT
2530	Reply Messages received – The number of AB-DF1 reply messages received from DF1 Devices on the network.	CNT_REPLY_MSGS_RCVD
2531	Master Poll Messages transmitted – The number of Master Poll Messages sent to DF1 Devices on the network.	CNT_POLL_MSGS_SENT
2532	Master Poll Messages received – The number of Master Poll Messages received from the DF1 Master	CNT_POLL_MSGS_RCVD
2533	NAK's transmitted – The number of NAK Symbols transmitted on the DF1 network	CNT_NAKS_SENT
2534	NAK's received – The number of NAK Symbols received from the DF1 network	CNT_NAKS_RCVD
2535	ACK's transmitted – The number of ACK Symbols transmitted on the DF1 network	CNT_ACKS_SENT
2536	ACK's received – The number of ACK Symbols received from the DF1 network	CNT_ACKS_RCVD
2537	EOT's transmitted – The number of EOT Symbols transmitted on the DF1 network	CNT_EOTS_SENT
2538	EOT's received – The number of EOT Symbols received from the DF1 network	CNT_EOTS_RCVD
2539	Error Messages Sent – The number of AB-DF1 error messages transmitted on the DF1 network	CNT_ERROR_MSGS_SENT
2540	Error Messages Received – The number of AB-DF1 error messages received from the DF1 network	CNT_ERROR_MSGS_RCVD
2541	NAK retries – The number of Command/Reply	CNT_NAK_RETRIES

ID	Description	Mnemonic
	Message retries attempted due to either no ACK Symbol received or a received NAK Symbol, or other communications error	
2542	NAK retry limit exceeded – The number of times that the maximum NAK retry limit threshold has been exceeded due to too many errors and the message aborted.	CNT_NAK_LIMIT_EXCEEDED
2543	ENQ retries – The number of Master Poll Message retries attempted due to either no ACK Symbol received or a received NAK symbol, or other communications error	CNT_ENQ_RETRIES
2544	ENQ retry limit exceeded – The number of times that the maximum ENQ retry limit threshold has been exceeded due to too many errors and the message aborted	CNT_ENQ_LIMIT_EXCEEDED
2545	Command Timeouts – The number of times that the expected total transaction time was exceeded (Master). In this case a NAK Symbol is sent to the remote device to terminate processing of the last command sent.	CNT_COMMAND_TIMEOUTS
2546	BCC Errors Detected – The number of BCC errors detected in the received data stream.	CNT_BCC_ERRORS
2547	CRC Errors Detected – The number of CRC errors detected in the received data stream.	CNT_CRC_ERRORS
2548	Stale Messages Received – The number of stale (leftover) messages received from DF1 devices on the network	CNT_STALE_MSGS_RCVD
2549	Duplicate Messages Received – The number of duplicate messages (due to lost/corrupted ACK Symbols) received from DF1 devices on the network	CNT_DUPLICATE_MSGS_RCVD
2550	Invalid Messages Received – The number of Invalid (bad data, incorrect addressing, etc.) messages received from DF1 devices on the network	CNT_INVALID_MSGS_RCVD
2551	Communication Read Errors – The number of communication port read errors that have occurred while attempting to read data from the serial communications port.	CNT_COM_READ_ERRORS

“Hardware” Status Counters

The AB-DF1 Module maintains two “hardware” status counters that keep track of how many characters have been transmitted and received on the assigned serial communications port. The following table lists the ID #, Mnemonic and a description of each “hardware” status counter implemented in the AB-DF1 Module.

ID	Description	Mnemonic
2560	Number of Characters Transmitted – The sum of all transmitted characters sent by this module over the serial communications port	CRD_CHAR_SENT
2561	Number of Characters Received – The sum of all characters received by this module from the serial communications port	CRD_CHAR_RECV

Error And Notice Messages

Critical Error Messages

The following critical error messages may be displayed during the parsing of the AB-DF1 “Net” parameter file. If any of the network configuration parameters are in error, a critical error is deemed to have occurred and the AB-DF1 module will not start. The following table lists the ID #, Mnemonic and description of each module specific Critical Error Message implemented in the AB-DF1 Module.

ID	Description	Mnemonic
2552	Invalid station number (%d) must be less than %d – The supplied station number parameter is not in the valid range [0 - 254].	CRT_BAD_STATION_ID
2553	Invalid Driver Mode (%s), should be one of %s, %s, or %s – The supplied driver mode parameter is not one of “Master”, “Slave” or “Peer”.	CRT_BAD_DRIVER_MODE
2554	Unsupported Driver Mode (%d), Supported Mode(s): %s – The supplied driver mode parameter is not supported in this version of AB-DF1 module.	CRT_UNSUPPORTED_MODE
2555	Invalid or Unsupported PLC Type (%s) should be one of %s, %s, %s, %s – The supplied PLC-Type parameter is either Invalid or Unsupported.	CRT_BAD_PLC_TYPE
2556	Invalid Error Code (%s), should be either %s, or %s – The supplied error code parameter is not one of “BCC” or “CRC”.	CRT_BAD_ERROR_CODE
2557	Invalid NAK Retries (%d) should be in range [0, %d] – The supplied NAK retries parameter is not in the valid range shown.	CRT_BAD_NAK_RETRIES
2558	Invalid ENQ Retries (%d) should be in range [0, %d] – The supplied ENQ retries parameter is not in the valid range shown.	CRT_BAD_ENQ_RETRIES
2559	Invalid Cmd Timeout (%d) should be in range [%d, %d] – The Command Timeout parameter is no in the valid range shown.	CRT_BAD_CMD_TIMEOUT

Error Messages

The following error messages may be displayed at any time during the operation of the AB-DF1 Module. Some may appear during the parsing of the AB-DF1 “Map” parameter file, while others may appear during module operation. The following table lists the ID #, Mnemonic and description of each module specific Error Message implemented in the AB-DF1 Module.

ID	Description	Mnemonic
2501	Item: %d, Invalid station number (%d) must be less than %d – The supplied station number parameter for one of the mappings in the mapping configuration file is not within the valid range [0 - 254].	ERR_BAD_STATION
2502	Item: %d, Invalid or Unsupported PLC Type (%s) should be one of %s, %s, %s, %s – The supplied PLC-type parameter for one of the mappings in the mapping configuration file is invalid or unsupported.	ERR_BAD_PLC_TYPE

ID	Description	Mnemonic
2503	Item: %d, Invalid Address Syntax (%s) – The supplied Address parameter for one of the mappings in the mapping configuration file does not comply with the Mapping Address Formatting Rules.	ERR_ADDRESS_SYNTAX
2504	Item: %d, Address is out of range (%s, len=%d) – The supplied Address parameter for one of the mappings in the mapping configuration file is not within a valid address range.	ERR_ADDRESS_RANGE
2505	Response with wrong command code (slv=%d, actual=%d, expected=%d) – A reply was received from a DF1 device on the network with a Bad Function Code.	ERR_BAD_FUNCTION_CODE
2506	Response with wrong CRC (actual=0x%02x%02x, expected=0x%02x%02x) – A reply was received from a DF1 device on the network with a bad CRC. This is an indication of network communication problems.	ERR_BAD_CRC
2507	Response with wrong BCC (actual=0x%02x%02x, expected=0x%02x%02x) – A reply was received from a DF1 device on the network with a bad BCC. This is an indication of network communication problems.	ERR_BAD_BCC
2508	Command Timed Out – A Command Message was sent to a DF1 Slave, but no valid reply was received in a timely manner from that slave.	ERR_COMMAND_TIMEOUT
2509	Slave Station (%d) Inactive – If a sufficient number of errors occurs while attempting to communicate with a DF1 Slave, then that Slave will be marked inactive and removed from the Active Station List.	ERR_INACTIVE_STATION
2510	Wrong Message Received Src:0x%02x, Dst:0x%02x, Tns:0x%04x – An unexpected or wrong message was received from a DF1 device on the network.	ERR_WRONG_MESSAGE
2511	Received NAK Symbol - A NAK symbol was received from a DF1 Device on the network.	ERR_RCVD_NAK_SYMBOL

Notice Messages

The following notice messages may be displayed at any time during the operation of the AB-DF1 Module. The following table lists the ID #, Mnemonic and description of each module specific Notice Message implemented in the AB-DF1 Module.

ID	Description	Mnemonic
2512	Received Message Too Long (> %d bytes) – The last message received was greater than the maximum allowed AB-DF1 message	NTC_RCVD_MSG_TOO_LONG
2513	Received Message Too Short (< %d bytes) – The last message received did not contain enough characters to constitute a valid AB-DF1 message.	NTC_RCVD_MSG_TOO_SHORT
2514	Invalid Symbol received after DLE - (0x%02x) – The next character read after DLE was not a valid symbol (should be one of SOH, STX, ETX, EOT, ENQ, ACK, DLE or NAK)	NTC_INVALID_SYMBOL
2515	ACK Expected (last symbol received was 0x%02x) – An ACK Symbol was expected but was not received.	NTC_ACK_EXPECTED

ID	Description	Mnemonic
2516	EOT or STX Expected (last symbol received was 0x%02x) – An EOT Symbol or an STX Symbol was expected but was not received.	NTC_EOT_STX_EXPECTED
2517	Received NAK Symbol – A NAK Symbol was received from a DF1 device on the network.	NTC_RCVD_NAK_SYMBOL
2518	Incomplete Message, No ETX received – A message Starting with SOH/STX was received but no ETX was received to signify the end of the message.	NTC_NO_ETX_RCVD
2519	NAK Retry Limit (%d) exceeded – The NAK Retry Limit Threshold has been exceeded for this message, due to excessive errors.	NTC_NAK_LIMIT_EXCEEDED
2520	ENQ Retry Limit (%d) exceeded – The ENQ Retry Limit Threshold has been exceeded when attempting to Poll for a reply, due to excessive errors.	NTC_ENQ_LIMIT_EXCEEDED
2521	NAK Retry #%d, due to error %d – A message error occurred when attempting to send a Message to a DF1 device on the network, a retry will be attempted	NTC_NAK_RETRY
2522	ENQ Retry #%d, due to error %d – A message error occurred when attempting to send a Poll Message to a DF1 device on the network, a retry will be attempted.	NTC_ENQ_RETRY
2523	Reply Message received from Slave %d – A successful reply message was received from a remote DF1 slave Device	NTC_REPLY_MSG_RCVD
2524	Duplicate Message received from Slave %d – A duplicate reply message was received from a remote DF1 slave Device (possibly due to a lost or corrupted ACK Symbol)	NTC_DUPLICATE_MSG_RCVD

DF1 and TX_Delay

Since a single Pronghorn Read/Write transaction consists of many DF1 message transactions, there are one or two things to keep in mind when selecting a TX_delay parameter in the network configuration .csv file. A single Pronghorn Read/Write transaction can consist of the following serial communications messages:

- 1) (possibly) a poll to determine the status (active/inactive) of a slave
- 2) one (or possibly more, depending on network conditions) master message frames followed by the associated DF1 slave acknowledge (DLE ACK)
- 3) one (or possibly more, depending on network conditions) master polling frames (DLE ENQ) followed by the associated DF1 slave response message.

Since TX_Delay comes into effect prior to EACH module DF1 transmission, it is quite possible to build up a significant delay in the processing of a single transaction. It is possible, when using polled messages to build up a sufficient delay that one or more messages may miss their polling “window”. This, in general, will only affect 2-wire (half-duplex) DF1 networks where multiple slaves are being ‘controlled’ by one master. For full-duplex networks (i.e. 4-wire), TX_Delay may be safely set to zero with little or no adverse results.

If this situation occurs, the best policy is to reduce the TX_Delay parameter to a minimum value for reliable operation.