Technical Note

AN-X2-GENI Genius Datagrams Rev 1.1

Purpose

This technical note shows how to send Datagrams through an AN-X2-GENI module from a ControlLogix processor.

Requirements

AN-X2-GENI module. Referred to as "AN-X" in this document.

AN-X2-GENI-MAS firmware version 4.3.1 or above.

Genius I/O System and Communications User's Manual GEK90486F-1 November 1994. This manual is referred to as "GeniComm" in this document.

Background

The AN-X uses Genius Datagrams as part of its normal configuration and fault monitoring functions.

In firmware rev 4.3.1 or later, the AN-X can send Datagrams using an unscheduled CIP Generic message on Ethernet (MSG instruction on ControlLogix).

Two types of Datagrams are supported.

1. Send Datagram with No Reply.
2. Send Datagram with Reply.

The AN-X Genius CIP Object (0xc4)

In firmware rev 4.3.1 or later, the AN-X has a "Genius CIP Object", referred to as GeniObj in this document.

The GeniObj has two commands that send Genius Datagrams. These commands are sent to **Instance 1**.

0x4c: Send Datagram with No Reply

This command triggers the AN-X to send a Datagram and not wait for a reply. A CIP response is returned immediately (MSG DN).

0x4d: Send Datagram with Reply

This command triggers the AN-X to send a Datagram on the Genius network, wait for a reply, then send the reply in the CIP response (MSG DN).

Command/Response Buffers

Ofs (SINT)	Name	Description
0	Len	Length of Datagram including Dst, Fnc, Src and Sub (3 to 132)
1	Dst	Destination Node (0-31)
2	Fnc	Function Code (usually 0x20)
3	Src	Source Node - Filled in by AN-X
4	Sub	Sub-Function
5..255	Data	Data - Byte 0-nn in GeniComm

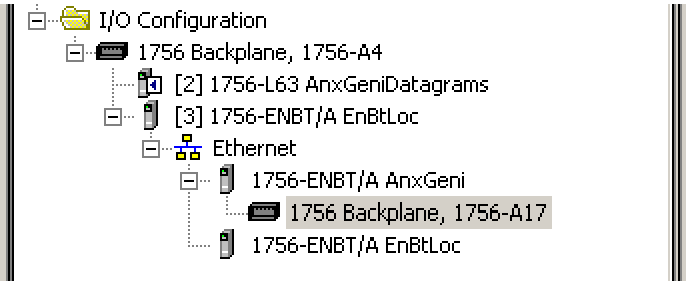
Error Responses

Gen Sts	Ext Sts	Description
0x08	N/A	Service Not Supported
0x14	N/A	Attribute Not Supported
0x13	N/A	Not enough data received - 4 bytes minimum for Datagram
0x15	N/A	Datagram Len > (CIP data len - 1)
0x0c	N/A	Object in Bad State. This can happen if Datagrams are being sent while an AN-X Auto-configure is performed.
0x01	0x207	Datagram Timeout

Sample Ladder Logic

I/O Configuration

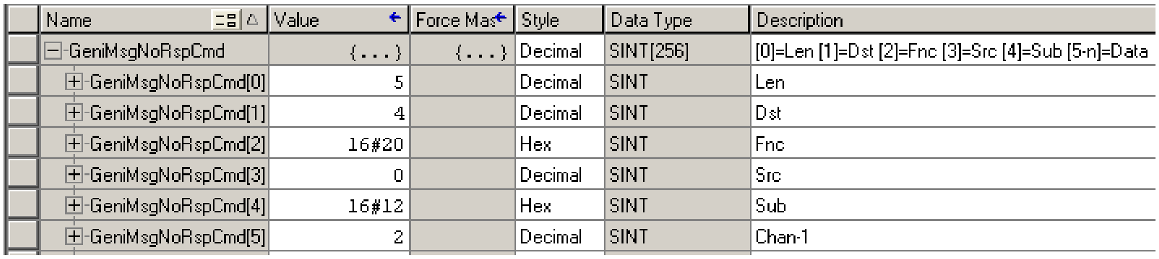
Note that scheduled connections to the AN-X are not necessary, but can be configured if the AN-X is to scan I/O and/or exchange Global Data as well.



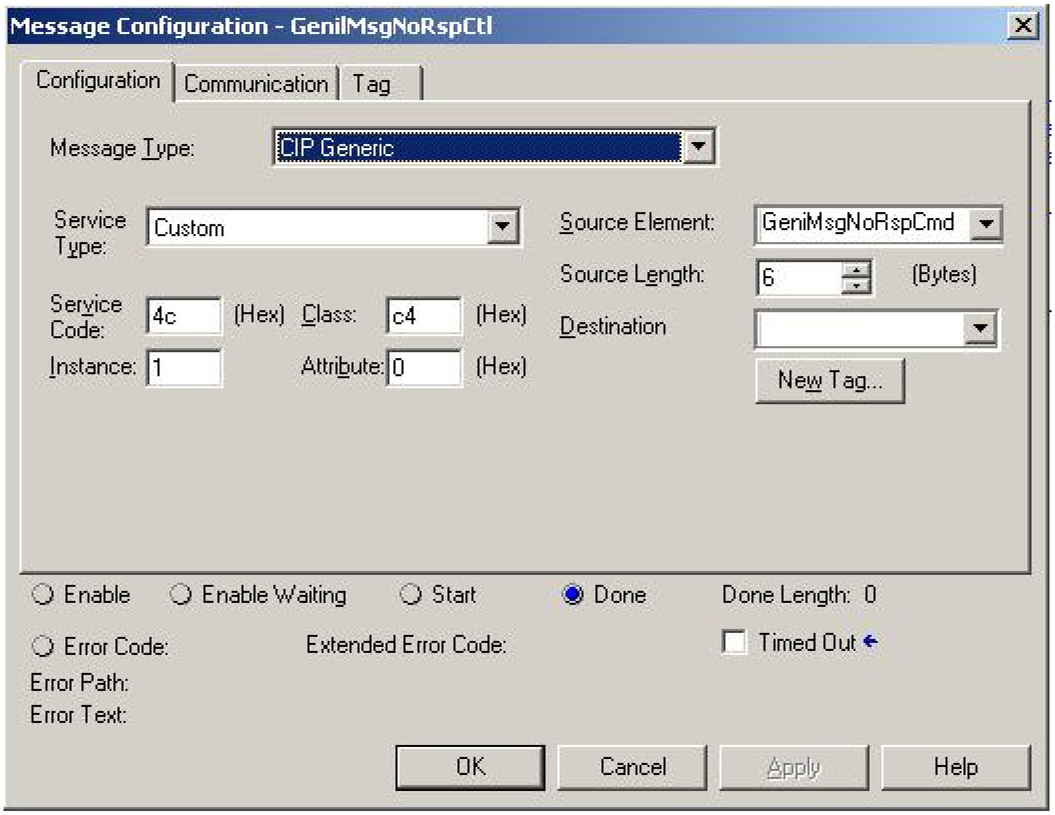
```
I/O Configuration
  1756 Backplane, 1756-A4
    [2] 1756-L63 AnxGeniDatagrams
    [3] 1756-ENBT/A EnBtLoc
      Ethernet
        1756-ENBT/A AnxGeni
          1756 Backplane, 1756-A17
        1756-ENBT/A EnBtLoc
```

Rung 0 - Send Datagram with No Reply

In this example we send a "Clear Circuit Fault" Datagram. See GeniComm pg 3-28.



Name	Value	Force Mas	Style	Data Type	Description
⊟ GeniMsgNoRspCmd	{...}	{...}	Decimal	SINT[256]	[0]=Len [1]=Dst [2]=Fnc [3]=Src [4]=Sub [5-n]=Data
⊞ GeniMsgNoRspCmd[0]	5		Decimal	SINT	Len
⊞ GeniMsgNoRspCmd[1]	4		Decimal	SINT	Dst
⊞ GeniMsgNoRspCmd[2]	16#20		Hex	SINT	Fnc
⊞ GeniMsgNoRspCmd[3]	0		Decimal	SINT	Src
⊞ GeniMsgNoRspCmd[4]	16#12		Hex	SINT	Sub
⊞ GeniMsgNoRspCmd[5]	2		Decimal	SINT	Chan-1



Message Configuration - GenilMsgNoRspCtl

Configuration | Communication | Tag

Message Type: CIP Generic

Service Type: Custom

Source Element: GeniMsgNoRspCmd

Source Length: 6 (Bytes)

Destination:

Service Code: 4c (Hex) Class: c4 (Hex)

Instance: 1 Attribute: 0 (Hex)

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

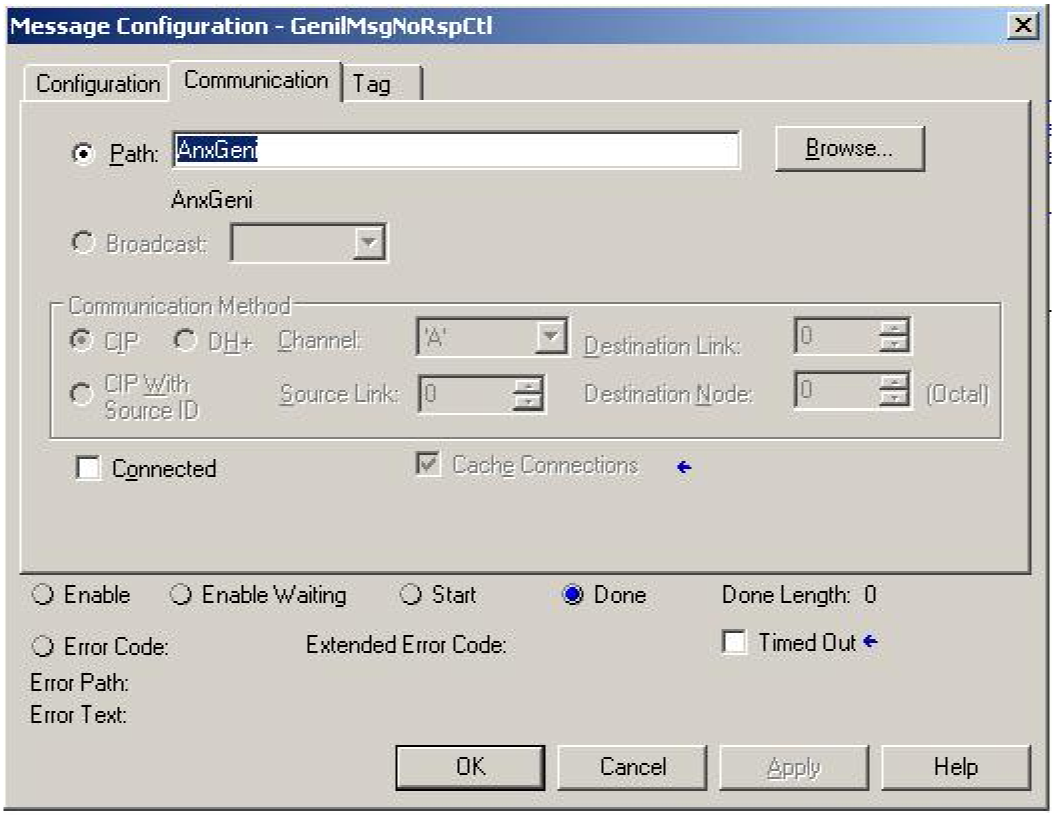
○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

Note that the Source Length in the dialog is 6 since it includes the Datagram Len byte.

Connected messaging is recommended for high message rates.



Message Configuration - GenilMsgNoRspCtl

● Path: AnxGeni

AnxGeni

○ Broadcast:

Communication Method

● CIP ○ DH+ Channel: 'A' Destination Link: 0

○ CIP With Source ID Source Link: 0 Destination Node: 0 (Octal)

☐ Connected ☑ Cache Connections

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

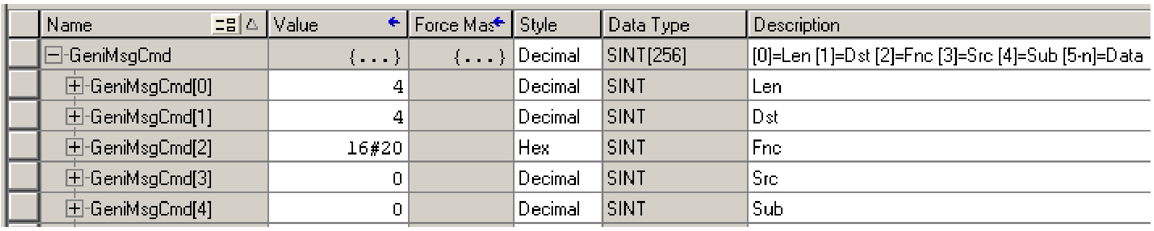
○ Error Code: Extended Error Code: ☐ Timed Out

Error Path:

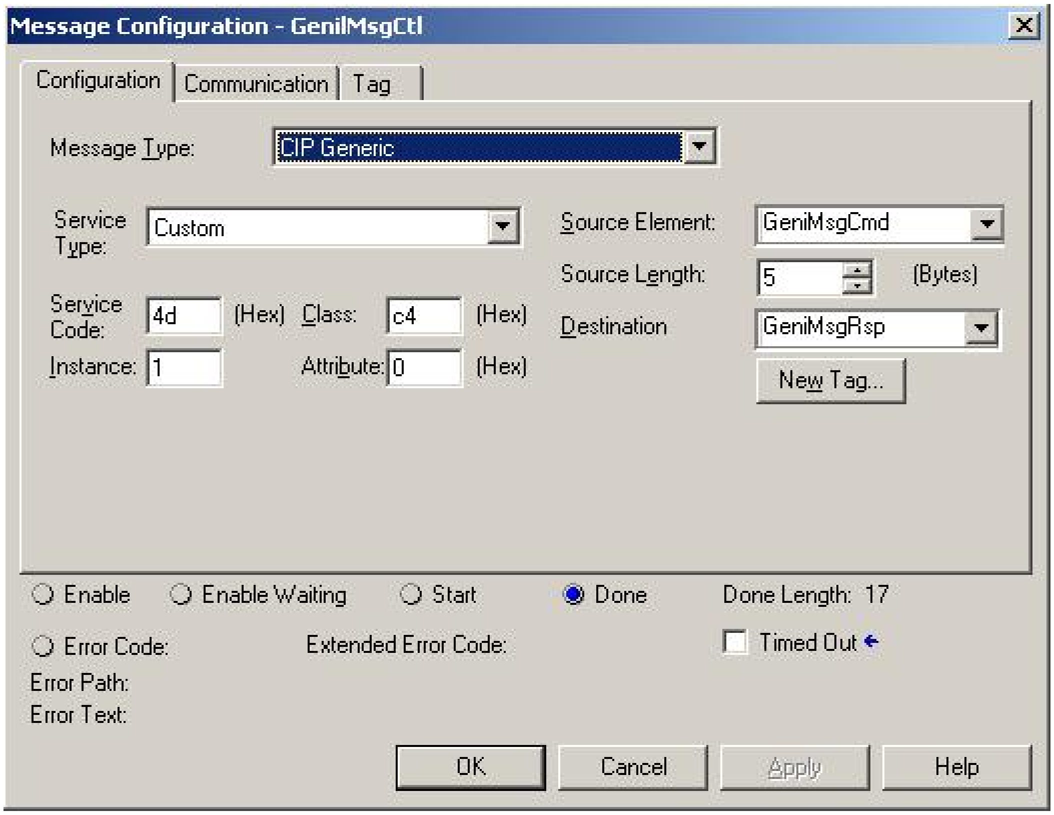
Error Text:

Rung 1 - Send Datagram with Reply

In this example we send a "Read Identification" Datagram. See GeniComm pg 3-8.



Name	Value	Force Mask	Style	Data Type	Description
⊟-GeniMsgCmd	{...}	{...}	Decimal	SINT[256]	[0]=Len [1]=Dst [2]=Fnc [3]=Src [4]=Sub [5-n]=Data
⊞-GeniMsgCmd[0]	4		Decimal	SINT	Len
⊞-GeniMsgCmd[1]	4		Decimal	SINT	Dst
⊞-GeniMsgCmd[2]	16#20		Hex	SINT	Fnc
⊞-GeniMsgCmd[3]	0		Decimal	SINT	Src
⊞-GeniMsgCmd[4]	0		Decimal	SINT	Sub



Message Configuration - GeniIMsgCtl

Message Type: CIP Generic

Service Type: Custom

Source Element: GeniMsgCmd

Source Length: 5 (Bytes)

Service Code: 4d (Hex) Class: c4 (Hex)

Destination: GeniMsgRsp

Instance: 1 Attribute: 0 (Hex)

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 17

○ Error Code: Extended Error Code: □ Timed Out

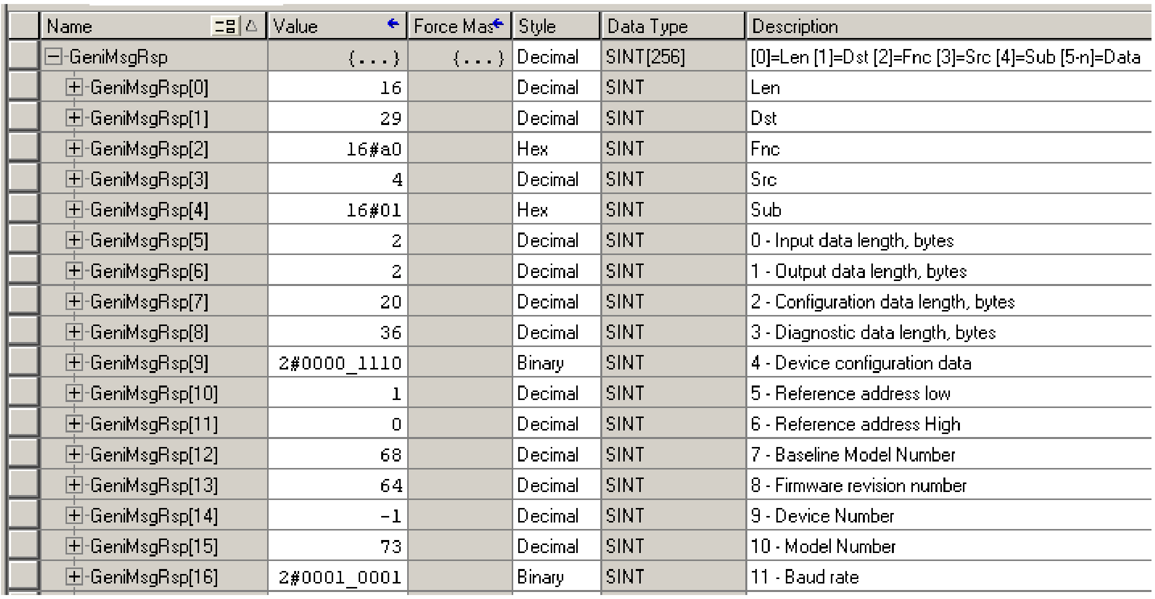
Error Path:

Error Text:

Note that the Source Length in the dialog is 5 since it includes the Datagram Len byte.

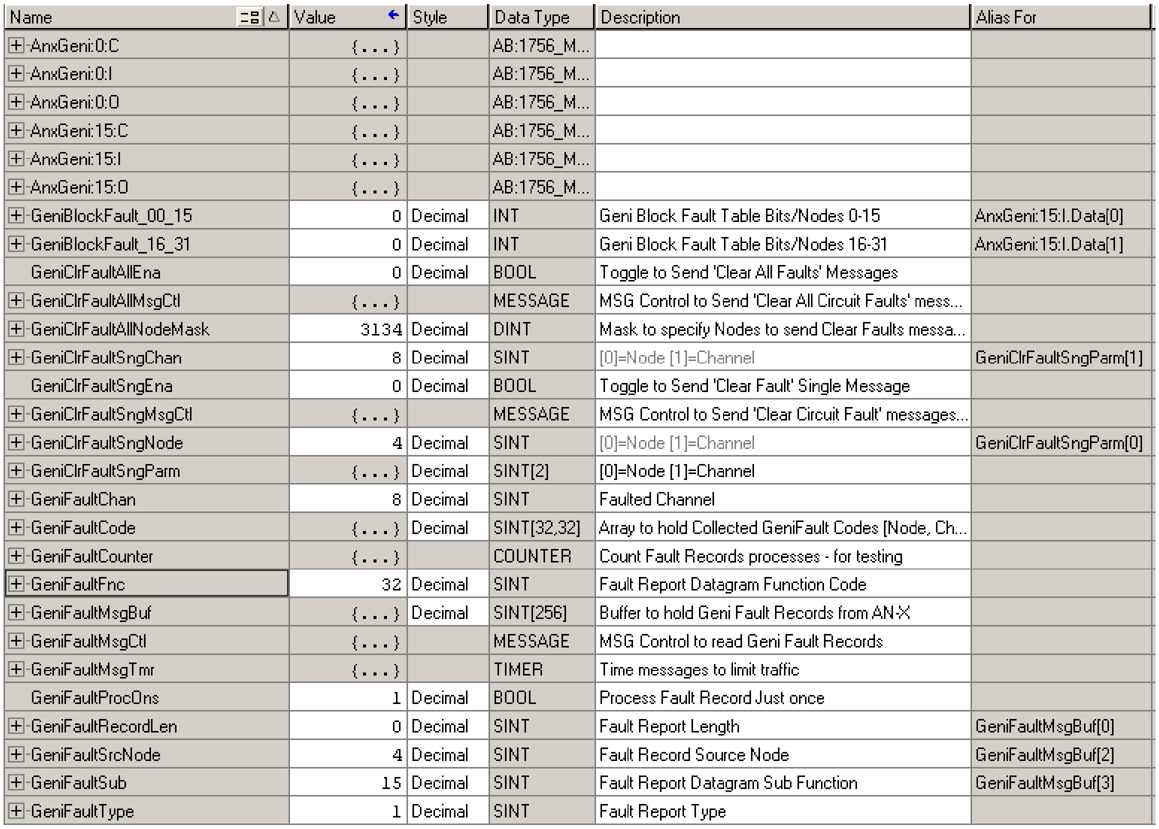
The communication tab is the same as "GeniMsgNoRspCtl".

Here is the response from Node 4 (IC660BBD020 24/48VDC 16 Ckt Source I/O Block).

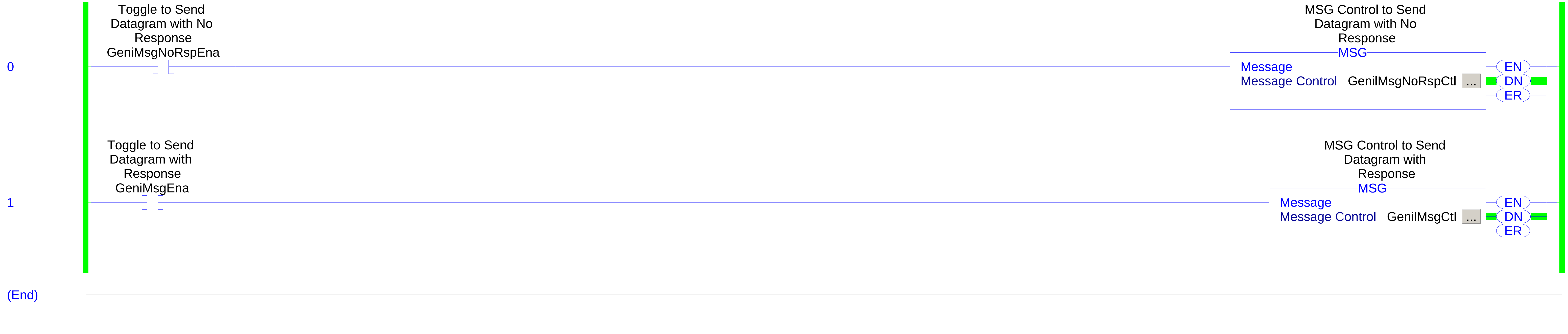


Name	Value	Force Mas	Style	Data Type	Description
GeniMsgRsp	{...}	{...}	Decimal	SINT[256]	[0]=Len [1]=Dst [2]=Fnc [3]=Src [4]=Sub [5-n]=Data
GeniMsgRsp[0]	16		Decimal	SINT	Len
GeniMsgRsp[1]	29		Decimal	SINT	Dst
GeniMsgRsp[2]	16#a0		Hex	SINT	Fnc
GeniMsgRsp[3]	4		Decimal	SINT	Src
GeniMsgRsp[4]	16#01		Hex	SINT	Sub
GeniMsgRsp[5]	2		Decimal	SINT	0 - Input data length, bytes
GeniMsgRsp[6]	2		Decimal	SINT	1 - Output data length, bytes
GeniMsgRsp[7]	20		Decimal	SINT	2 - Configuration data length, bytes
GeniMsgRsp[8]	36		Decimal	SINT	3 - Diagnostic data length, bytes
GeniMsgRsp[9]	2#0000_1110		Binary	SINT	4 - Device configuration data
GeniMsgRsp[10]	1		Decimal	SINT	5 - Reference address low
GeniMsgRsp[11]	0		Decimal	SINT	6 - Reference address High
GeniMsgRsp[12]	68		Decimal	SINT	7 - Baseline Model Number
GeniMsgRsp[13]	64		Decimal	SINT	8 - Firmware revision number
GeniMsgRsp[14]	-1		Decimal	SINT	9 - Device Number
GeniMsgRsp[15]	73		Decimal	SINT	10 - Model Number
GeniMsgRsp[16]	2#0001_0001		Binary	SINT	11 - Baud rate

Controller Tags



Name	Value	Style	Data Type	Description	Alias For
AnxGeni:0:C	{...}		AB:1756_M...		
AnxGeni:0:I	{...}		AB:1756_M...		
AnxGeni:0:O	{...}		AB:1756_M...		
AnxGeni:15:C	{...}		AB:1756_M...		
AnxGeni:15:I	{...}		AB:1756_M...		
AnxGeni:15:O	{...}		AB:1756_M...		
GeniBlockFault_00_15	0	Decimal	INT	Geni Block Fault Table Bits/Nodes 0-15	AnxGeni:15:I.Data[0]
GeniBlockFault_16_31	0	Decimal	INT	Geni Block Fault Table Bits/Nodes 16-31	AnxGeni:15:I.Data[1]
GeniClrFaultAllEna	0	Decimal	BOOL	Toggle to Send 'Clear All Faults' Messages	
GeniClrFaultAllMsgCtl	{...}		MESSAGE	MSG Control to Send 'Clear All Circuit Faults' mess...	
GeniClrFaultAllNodeMask	3134	Decimal	DINT	Mask to specify Nodes to send Clear Faults messa...	
GeniClrFaultSngChan	8	Decimal	SINT	[0]=Node [1]=Channel	GeniClrFaultSngParm[1]
GeniClrFaultSngEna	0	Decimal	BOOL	Toggle to Send 'Clear Fault' Single Message	
GeniClrFaultSngMsgCtl	{...}		MESSAGE	MSG Control to Send 'Clear Circuit Fault' messages...	
GeniClrFaultSngNode	4	Decimal	SINT	[0]=Node [1]=Channel	GeniClrFaultSngParm[0]
GeniClrFaultSngParm	{...}	Decimal	SINT[2]	[0]=Node [1]=Channel	
GeniFaultChan	8	Decimal	SINT	Faulted Channel	
GeniFaultCode	{...}	Decimal	SINT[32,32]	Array to hold Collected GeniFault Codes [Node, Ch...	
GeniFaultCounter	{...}		COUNTER	Count Fault Records processes - for testing	
GeniFaultFnc	32	Decimal	SINT	Fault Report Datagram Function Code	
GeniFaultMsgBuf	{...}	Decimal	SINT[256]	Buffer to hold Geni Fault Records from AN-X	
GeniFaultMsgCtl	{...}		MESSAGE	MSG Control to read Geni Fault Records	
GeniFaultMsgTmr	{...}		TIMER	Time messages to limit traffic	
GeniFaultProcOns	1	Decimal	BOOL	Process Fault Record Just once	
GeniFaultRecordLen	0	Decimal	SINT	Fault Report Length	GeniFaultMsgBuf[0]
GeniFaultSrcNode	4	Decimal	SINT	Fault Record Source Node	GeniFaultMsgBuf[2]
GeniFaultSub	15	Decimal	SINT	Fault Report Datagram Sub Function	GeniFaultMsgBuf[3]
GeniFaultType	1	Decimal	SINT	Fault Report Type	


0
Toggle to Send Datagram with No Response
GeniMsgNoRspEna
MSG Control to Send Datagram with No Response
MSG
Message
Message Control GeniIMsgNoRspCtl ...
EN
DN
ER
1
Toggle to Send Datagram with Response
GeniMsgEna
MSG Control to Send Datagram with Response
MSG
Message
Message Control GeniIMsgCtl ...
EN
DN
ER
(End)