Technical Note

AN-X2-GENI Genius Fault Access   Rev 1.1

Purpose

This technical note shows how to access Genius Fault Records on the AN-X2-GENI module from a ControlLogix processor.

It also shows how to send "Clear Fault" messages to one or more nodes on the Genius network 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

Most Genius blocks collect fault and diagnostic information for their I/O. When a fault is detected the block sends a "Report Fault" Genius Datagram to the node scanning it (GeniComm pg 3-17).

The AN-X receives these "Report Fault" Datagrams and logs them as Fault Records. They can be viewed in the AN-X web interface.

```
43:29.171 405 :**Fault #4 : Disc/HSC Fault Chan#9  [08 02] 'No Load/Open Wire'
43:29.197 954 :**Fault #4 : Disc/HSC Fault Chan#10 [08 02] 'No Load/Open Wire'
```

In firmware rev 4.3.1 or later, the AN-X allows the Fault Record log to be read using an unscheduled CIP Generic message on Ethernet (MSG instruction on ControlLogix).

The faults on the Genius blocks can be cleared by sending them a "Clear Circuit Fault" or a "Clear All Circuit Faults" Datagram (GeniComm pg 3-28).

The AN-X web interface supports the "Clear All Circuit Faults" Datagram. It can be sent to one Node, or all nodes.

In firmware rev 4.3.1 or later, the AN-X allows the "Clear Circuit Fault" or "Clear All Circuit Faults" commands to be sent using an unscheduled CIP Generic message on Ethernet (MSG instruction on ControlLogix).

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 three commands that access Genius Device Faults. These commands are sent to **Instance 1**.

0x4a: Read Fault Record

The Fault Record is an exact copy of the Datagram received from the reporting block or node.

Ofs (SINT)	Name	Description
0	Len	Length of Datagram including Fnc, Src and Sub. 0 indicates Record buffer empty.
1	Fnc	Function Code (usually 0x20)
2	Src	Source Node
3	Sub	Sub-Function
4..255	Data	Data - Byte 0-nn in GeniComm

0x4e: Clear Circuit Fault

This command triggers the AN-X to send a "Clear Circuit Fault" Datagram (Function 0x20 Sub-function 0x12, GeniComm 3-28). This command has two SINT parameters (2 bytes).

Ofs (SINT)	Name	Description
0	DstNode	Destination Node (0-31)
1	Channel	Channel to clear (1 to N, 0 same as 1)

0x4f: Clear All Circuit Faults

This command triggers the AN-X to send up to 31 "Clear All Circuit Fault" Datagrams (Function 0x20 Sub-function 0x13, GeniComm 3-28). This command has one DINT parameter (4 bytes).

Ofs (DINT)	Name	Description
0	NodeMask	Bit Mask of Nodes to send Datagrams to. The Bit number corresponds to the Node number. Bit 0=Node 0, bit 1=Node 1 etc.

Accessing and Monitoring Fault Records

When the AN-X receives "Report Fault" Datagrams, it sets the bit corresponding to the source node in the "Block Fault Table".

The block fault table consists of two 16-bit words, one bit per serial bus address. Bit 0 of the first word corresponds to node (serial bus address) 0, bit 1 corresponds to node (serial bus address 1), and so on. The bits in the second word correspond to nodes 16 to 31.

By default, the AN-X maps the Block Fault Table into the first two words of Connection 15.

The ControlLogix sets the first 32 bits of any connection to 1 if the connection is not active. This corresponds to -1 in the first two 16 bit INTs of Input data from the AN-X.

With the default mapping then, the Block Fault Table mapped into Connection 15 goes to -1, -1 when the connection is not active.

The sample ladder logic checks for Connection 15 active (Data[0] or Data[1] not equal to -1), then for any Block Fault Table bits to be set (GeniBlockFault_00_15 or GeniBlockFault_16_31 non zero) before sending a CIP message to the GeniObj.

A timer is used to limit the rate that the CIP messages are sent. This prevents flooding of the CLX message queue.

Processing Genius Fault Records

The sample ladder logic processes the Fault Record read using a one-shot on the rising edge of the MSG DN bit.

The logic does the following:

- Checks for non-zero GeniFaultRecordLen
- Checks that the Datagram Function code is 0x20 (masking off uppermost bit).
- Checks that the Sub-Function is 0x0f.
- Extracts the Fault Type from the lower 3 bits (mask 0x07) of "byte 0". See GeniComm pg 3-19 to 3-21.
- Checks for Fault Type 1 (Discrete and High Speed Counter), Type 2 (Analog) or 4 (RTD and Thermocouple).
- Extracts the Channel Number into GeniFaultChan (bits 4-7 and bit 3 of "byte 0", GeniComm pg 3-19). Note that bit 3 of "byte 0" is used for discretes only, but it is 0 for analog and RTD so we can still use it as the uppermost bit for the 5 bit channel number.
- Using the GeniFaultSrcNode and GeniFaultChan, the Fault Record "byte 1" value is "OR'd" into the two dimensional array GeniFaultCode.

Note: as with other addressing between the CLX and Genius devices, CLX uses whole number (0-n) while Genius devices use natural numbers (1-n).

For example, if Node #4 channel #9 reported a "No Load" fault, we expect bit 3 of GeniFaultCode[4,8] to be set (see GeniComm pg 3-19 Fault Description byte 1).

Again, note that channel 9 corresponds to offset 8 in the array.

An end user would add logic to check for and process the faults.

HMIs can of course access the GeniFaultCode array.

Clearing Genius Device Faults

Device faults are cleared using GeniObj commands 0x4e and 0x4f.

See the sample ladder logic for details.

Error Responses

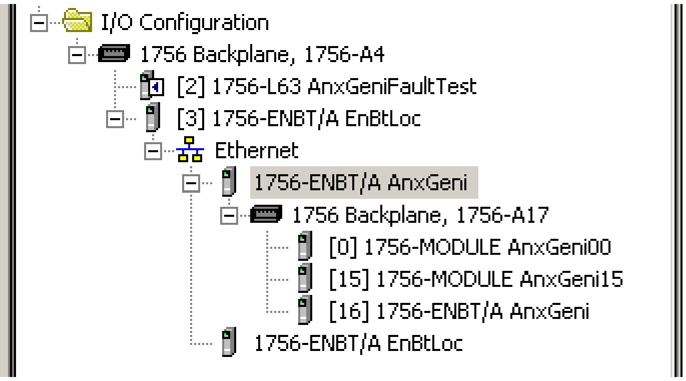
Gen Sts	Ext Sts	Description
0x08	N/A	Service Not Supported
0x14	N/A	Attribute Not Supported
0x13	N/A	Command Size Mismatch (2 bytes for 0x4e, 4 bytes for 0x4f)

Sample Ladder Logic

Sample ladder logic is included at the end of this document.

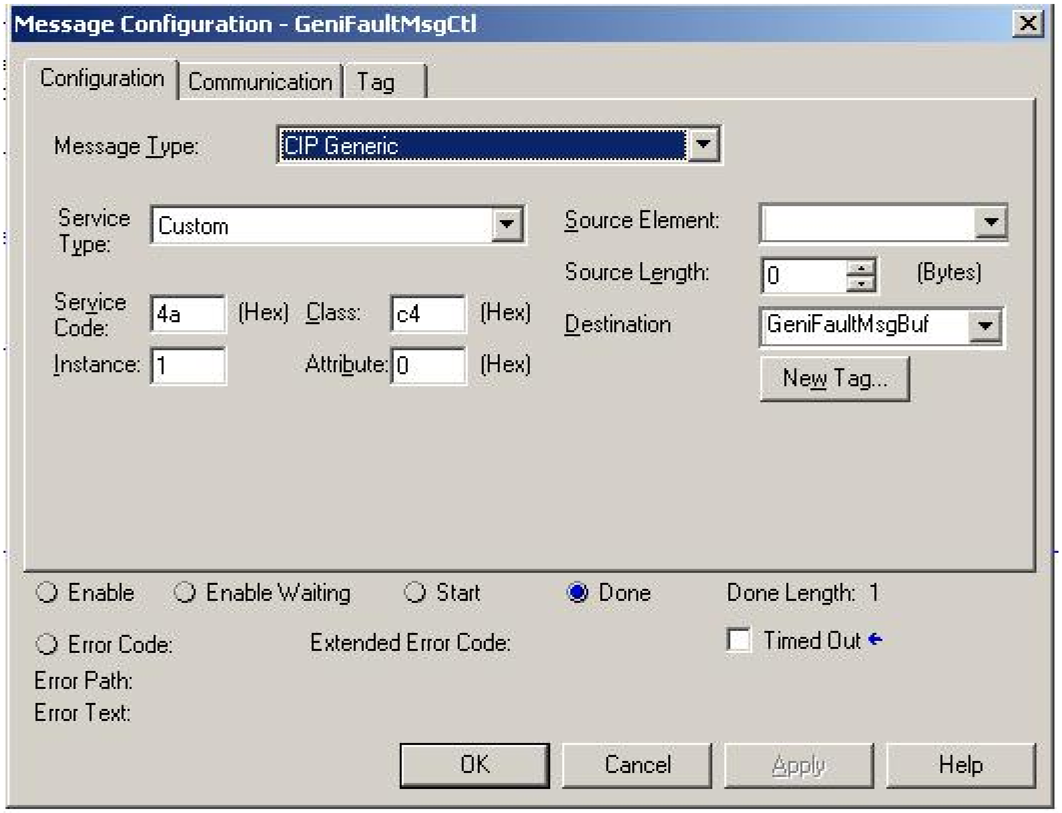
Below are some screen shots showing the MSG configurations and the Controller Tags.

I/O Configuration



```
I/O Configuration
  1756 Backplane, 1756-A4
    [2] 1756-L63 AnxGeniFaultTest
    [3] 1756-ENBT/A EnBtLoc
      Ethernet
        1756-ENBT/A AnxGeni
          1756 Backplane, 1756-A17
            [0] 1756-MODULE AnxGeni00
            [15] 1756-MODULE AnxGeni15
            [16] 1756-ENBT/A AnxGeni
        1756-ENBT/A EnBtLoc
```

"Read Fault Record" Message Configuration (Rung 1)



Message Type: CIP Generic

Service Type: Custom

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

Instance: 1 Attribute: 0 (Hex)

Source Element:

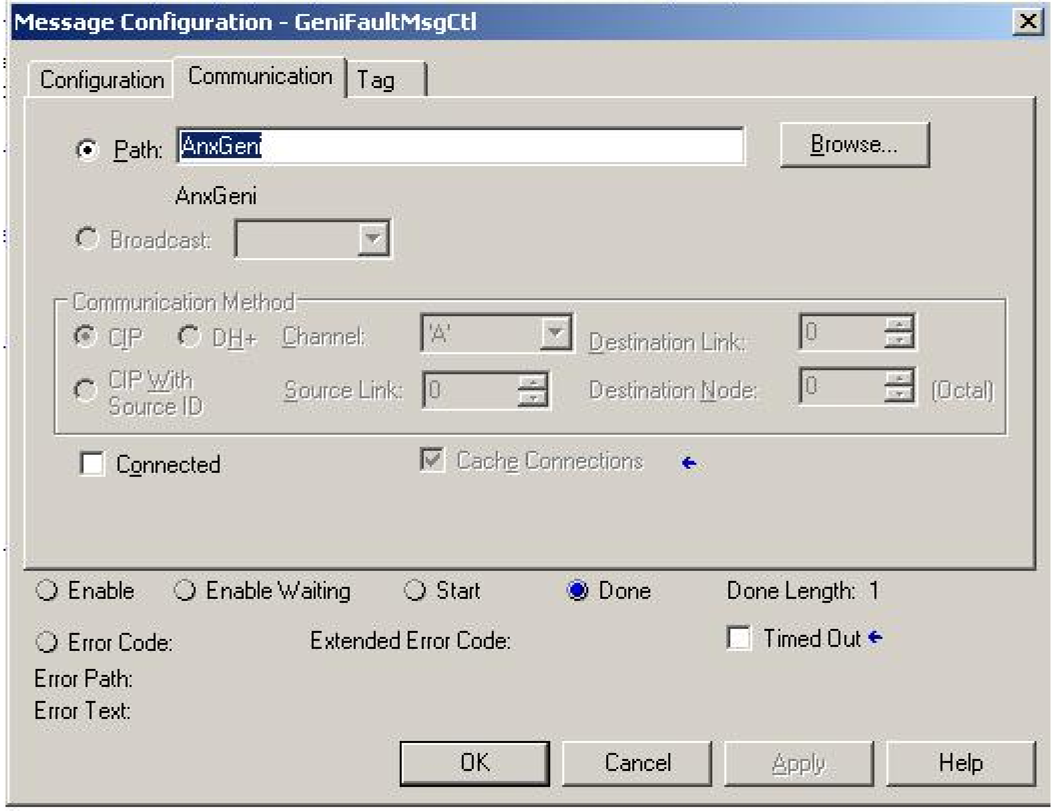
Source Length: 0 (Bytes)

Destination GeniFaultMsgBuf

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 1

○ Error Code: Extended Error Code: □ Timed Out

Error Path:
Error Text:



● Path: AnxGeni

AnxGeni

○ Broadcast:

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

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

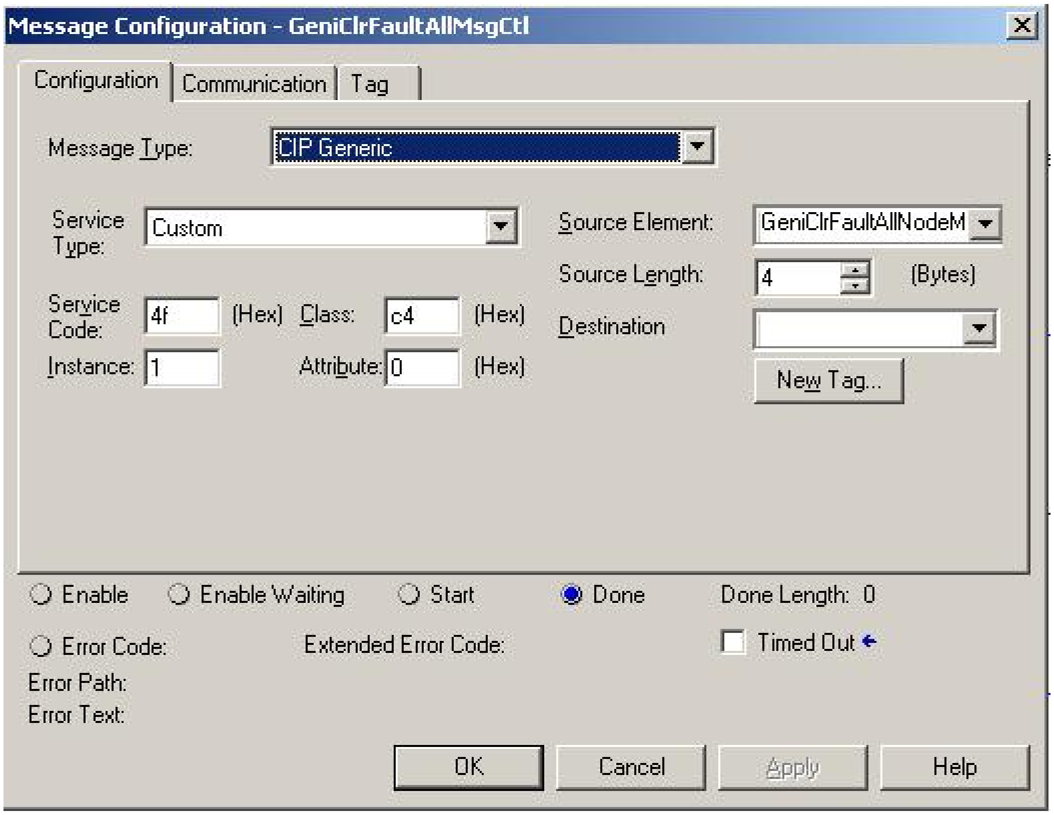
□ Connected ☑ Cache Connections

○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 1

○ Error Code: Extended Error Code: □ Timed Out

Error Path:
Error Text:

"Clear All Circuit Faults" Message Configuration (Rung 4)



Message Configuration - GeniClrFaultAllMsgCtl

Message Type: CIP Generic

Service Type: Custom

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

Instance: 1 Attribute: 0 (Hex)

Source Element: GeniClrFaultAllNodeM

Source Length: 4 (Bytes)

Destination:

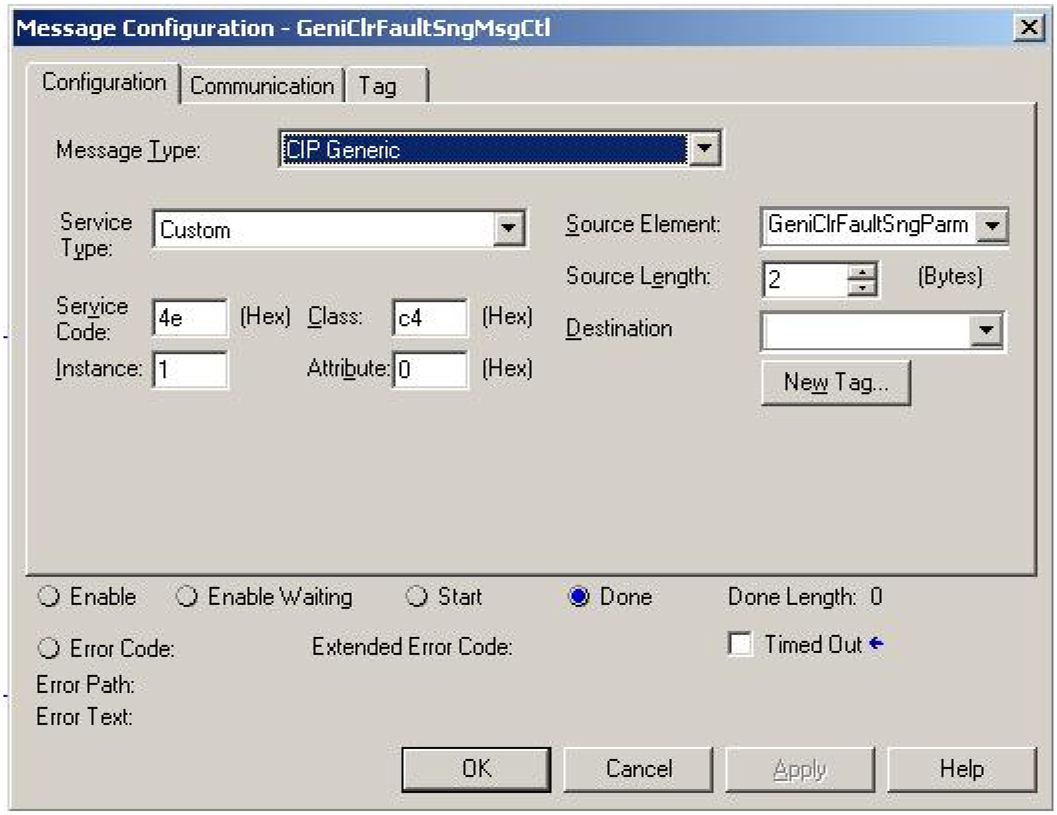
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

Error Path:

Error Text:

"Clear Circuit Fault" Message Configuration (Rung 5)



Message Configuration - GeniClrFaultSngMsgCtl

Message Type: CIP Generic

Service Type: Custom

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

Instance: 1 Attribute: 0 (Hex)

Source Element: GeniClrFaultSngParm

Source Length: 2 (Bytes)

Destination:

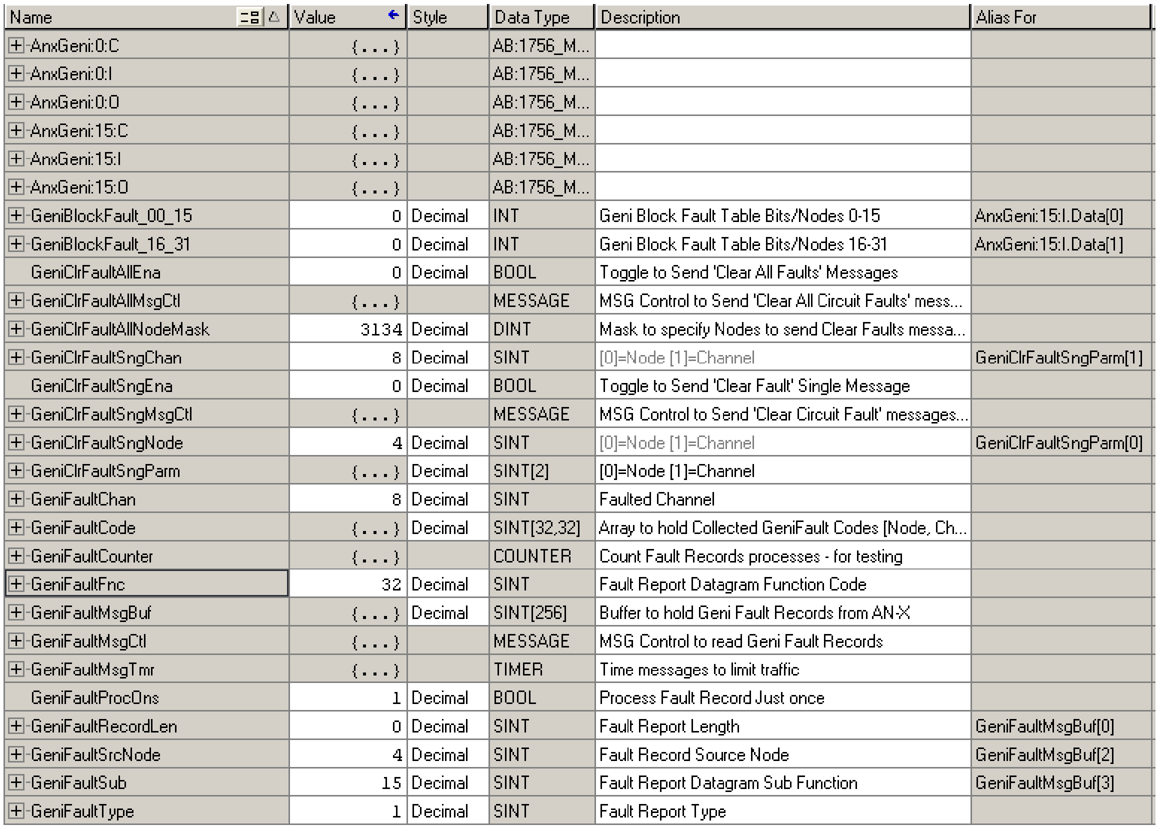
○ Enable ○ Enable Waiting ○ Start ● Done Done Length: 0

○ Error Code: Extended Error Code: □ Timed Out

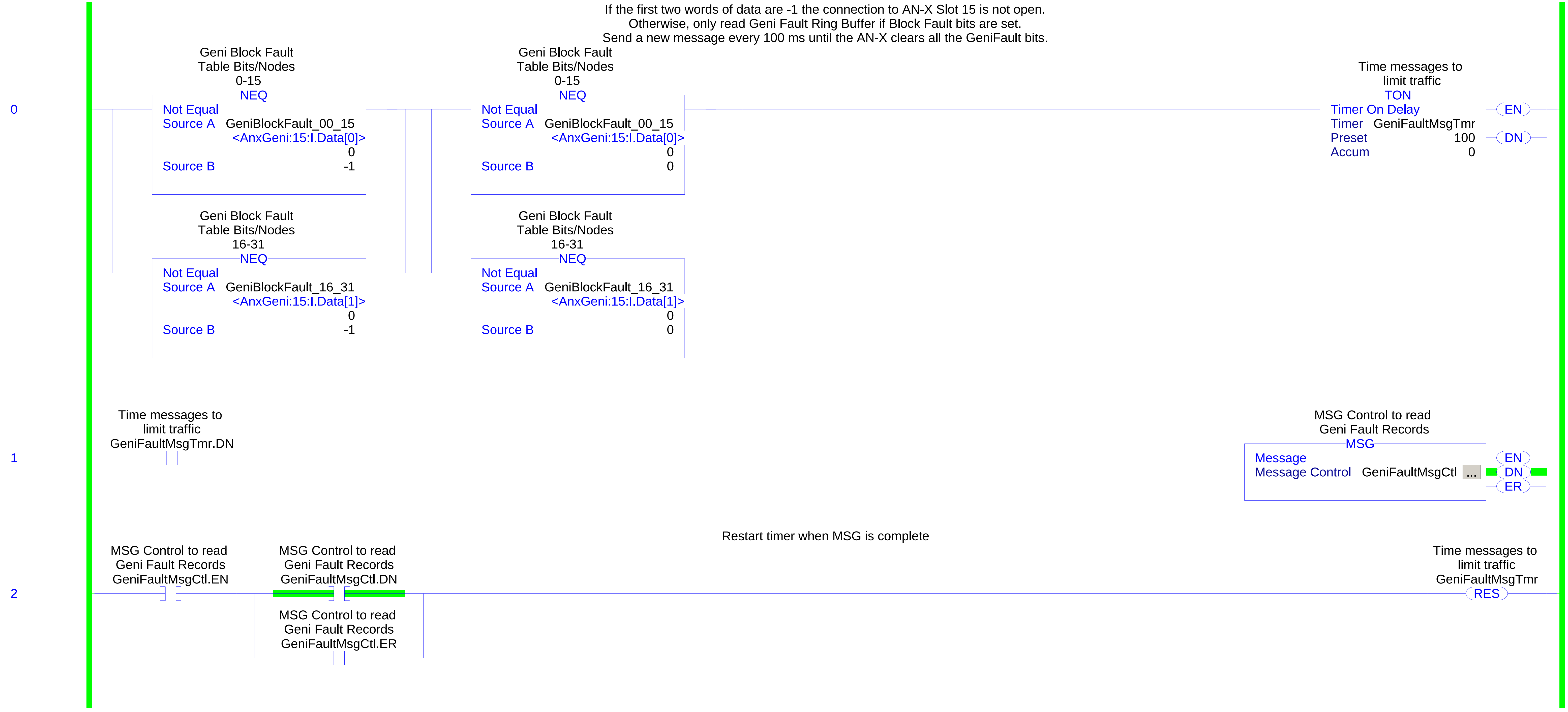
Error Path:

Error Text:

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

If the first two words of data are -1 the connection to AN-X Slot 15 is not open.
Otherwise, only read Geni Fault Ring Buffer if Block Fault bits are set.
Send a new message every 100 ms until the AN-X clears all the GeniFault bits.

Geni Block Fault
Table Bits/Nodes
0-15
NEQ
Not Equal
Source A GeniBlockFault_00_15
<AnxGeni:15:I.Data[0]>
0
Source B -1

Geni Block Fault
Table Bits/Nodes
16-31
NEQ
Not Equal
Source A GeniBlockFault_16_31
<AnxGeni:15:I.Data[1]>
0
Source B -1

Geni Block Fault
Table Bits/Nodes
0-15
NEQ
Not Equal
Source A GeniBlockFault_00_15
<AnxGeni:15:I.Data[0]>
0
Source B 0

Geni Block Fault
Table Bits/Nodes
16-31
NEQ
Not Equal
Source A GeniBlockFault_16_31
<AnxGeni:15:I.Data[1]>
0
Source B 0

Time messages to
limit traffic
TON
Timer On Delay
Timer GeniFaultMsgTmr
Preset 100
Accum 0
EN
DN

1

Time messages to
limit traffic
GeniFaultMsgTmr.DN

MSG Control to read
Geni Fault Records
MSG
Message
Message Control GeniFaultMsgCtl ...
EN
DN
ER

2

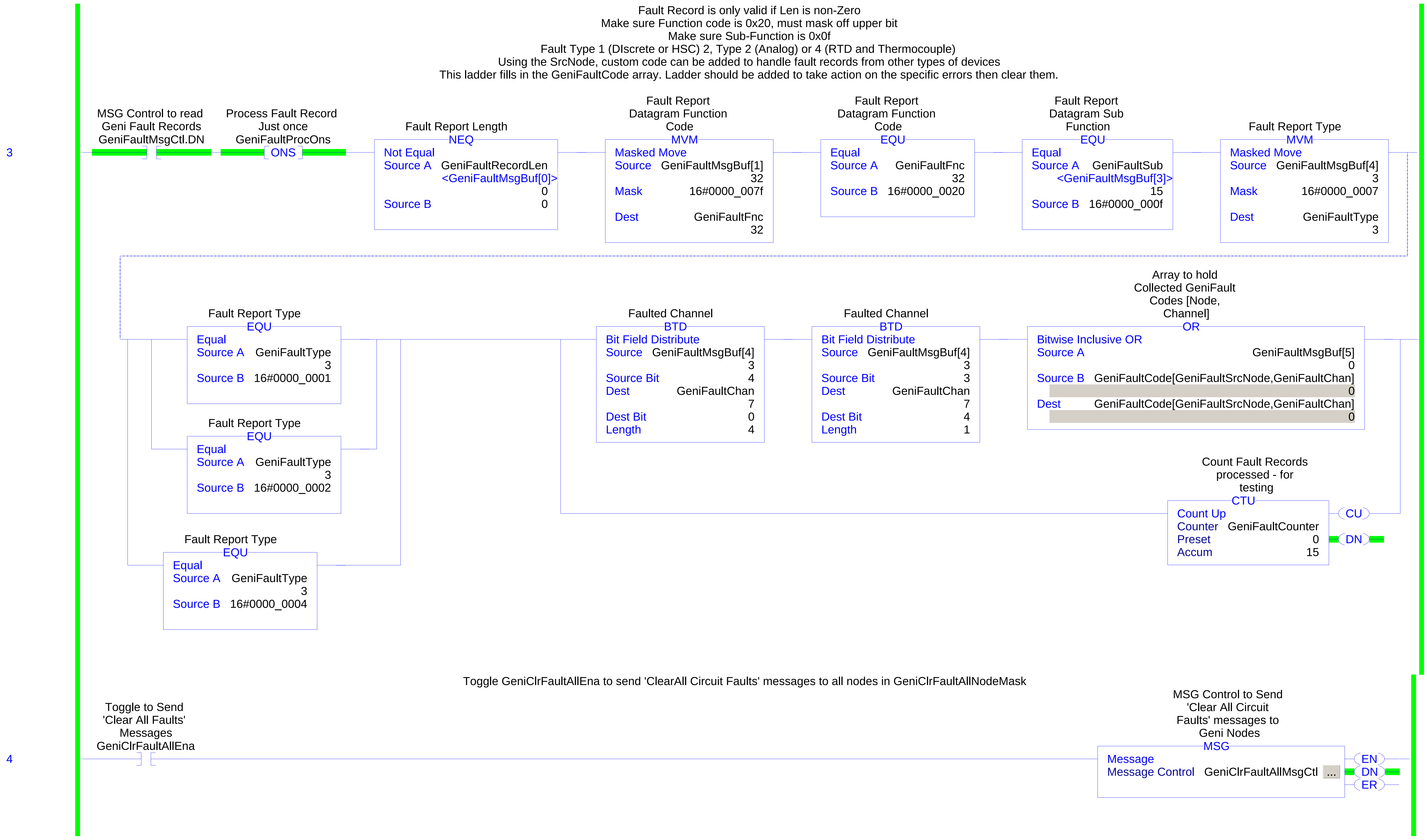
Restart timer when MSG is complete

MSG Control to read
Geni Fault Records
GeniFaultMsgCtl.EN

MSG Control to read
Geni Fault Records
GeniFaultMsgCtl.DN

MSG Control to read
Geni Fault Records
GeniFaultMsgCtl.ER

Time messages to
limit traffic
GeniFaultMsgTmr
RES



Fault Record is only valid if Len is non-Zero
Make sure Function code is 0x20, must mask off upper bit
Make sure Sub-Function is 0x0f
Fault Type 1 (DIscrete or HSC) 2, Type 2 (Analog) or 4 (RTD and Thermocouple)
Using the SrcNode, custom code can be added to handle fault records from other types of devices
This ladder fills in the GeniFaultCode array. Ladder should be added to take action on the specific errors then clear them.

3

MSG Control to read Geni Fault Records
GeniFaultMsgCtl.DN

Process Fault Record Just once
GeniFaultProcOns
ONS

Fault Report Length
NEQ
Not Equal
Source A GeniFaultRecordLen
<GeniFaultMsgBuf[0]>
0
Source B 0

Fault Report Datagram Function Code
MVM
Masked Move
Source GeniFaultMsgBuf[1]
32
Mask 16#0000_007f
Dest GeniFaultFnc
32

Fault Report Datagram Function Code
EQU
Equal
Source A GeniFaultFnc
32
Source B 16#0000_0020

Fault Report Datagram Sub Function
EQU
Equal
Source A GeniFaultSub
<GeniFaultMsgBuf[3]>
15
Source B 16#0000_000f

Fault Report Type
MVM
Masked Move
Source GeniFaultMsgBuf[4]
3
Mask 16#0000_0007
Dest GeniFaultType
3

Fault Report Type
EQU
Equal
Source A GeniFaultType
3
Source B 16#0000_0001

Fault Report Type
EQU
Equal
Source A GeniFaultType
3
Source B 16#0000_0002

Fault Report Type
EQU
Equal
Source A GeniFaultType
3
Source B 16#0000_0004

Faulted Channel
BTD
Bit Field Distribute
Source GeniFaultMsgBuf[4]
3
Source Bit 4
Dest GeniFaultChan
7
Dest Bit 0
Length 4

Faulted Channel
BTD
Bit Field Distribute
Source GeniFaultMsgBuf[4]
3
Source Bit 3
Dest GeniFaultChan
7
Dest Bit 4
Length 1

Array to hold Collected GeniFault Codes [Node, Channel]
OR
Bitwise Inclusive OR
Source A GeniFaultMsgBuf[5]
0
Source B GeniFaultCode[GeniFaultSrcNode,GeniFaultChan]
0
Dest GeniFaultCode[GeniFaultSrcNode,GeniFaultChan]
0

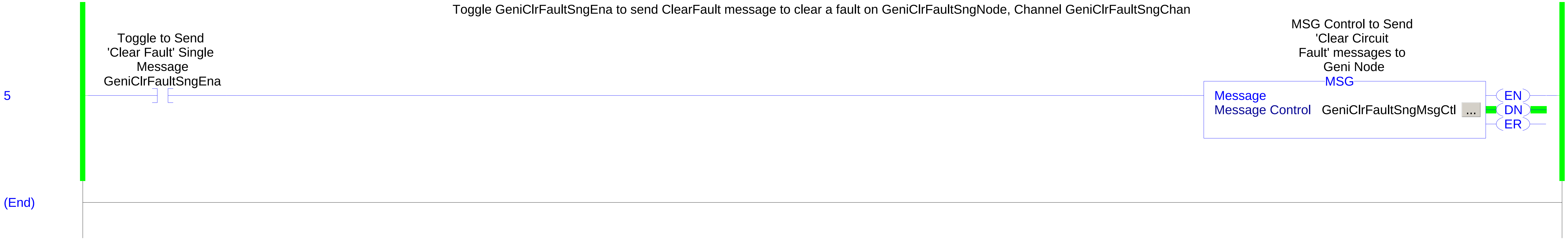
Count Fault Records processed - for testing
CTU
Count Up
Counter GeniFaultCounter
Preset 0
Accum 15
CU
DN

Toggle GeniClrFaultAllEna to send 'ClearAll Circuit Faults' messages to all nodes in GeniClrFaultAllNodeMask

4

Toggle to Send 'Clear All Faults' Messages
GeniClrFaultAllEna

MSG Control to Send 'Clear All Circuit Faults' messages to Geni Nodes
MSG
Message
Message Control GeniClrFaultAllMsgCtl ...
EN
DN
ER



Toggle GeniClrFaultSngEna to send ClearFault message to clear a fault on GeniClrFaultSngNode, Channel GeniClrFaultSngChan

5
Toggle to Send 'Clear Fault' Single Message
GeniClrFaultSngEna
MSG Control to Send 'Clear Circuit Fault' messages to Geni Node
MSG
Message
Message Control
GeniClrFaultSngMsgCtl
...
EN
DN
ER
(End)