



GIS GuardLogix Programming With Sample Code

**Tim Williams
Control Systems**

October 26, 2012

REVISION SUMMARY:

1. Date: 11/11/11
Revision: DRAFT 1
Changes: Rough draft for review and comments.
2. Date: 7/11/12
Revision: DRAFT 2
Changes: draft for FDR.
3. Date: October 26, 2012
Revision: Revision A
Changes: Initial formal release

Table of Contents

PREFACE	V
1. INTRODUCTION	1
1.1 PURPOSE	1
1.2 INTENDED AUDIENCE	1
2. DOCUMENTS	2
2.1 RELATED DOCUMENTS	2
2.2 REFERENCED DOCUMENTS	2
2.3 SOFTWARE	2
3. CONFIGURING THE NETWORK	3
3.1 INITIAL SETUP	3
3.2 SETTING UP PORTS	4
3.3 SETTING UP VLANS	4
4. CONFIGURING THE GUARDLOGIX CONTROLLER	7
4.1 FIRMWARE	7
4.2 SAFETY NETWORK NUMBER	7
4.3 SAFETY LOCKING	7
4.4 I/O MODULES	7
4.4.1 Module Definitions	7
5. CONFIGURING REMOTE I/O MODULES	8
5.1 FIRMWARE	8
6. IP ADDRESSING	9
6.1 VIRTUAL LANS	9
6.2 HOST ADDRESS	9
7. TAGS	11
7.1 TAG SCOPE	11
7.2 PRODUCED AND CONSUMED TAGS	11
7.3 DATA ACCESS CONTROL	11
7.3.1 External Access	11
7.3.2 Constant	11
8. GENERAL NAMING CONVENTIONS	12
8.1 PROGRAMS AND ROUTINES	12
8.2 TAG NAMING	12
8.2.1 General Tag Name Guidelines	12
8.2.2 Controller-Scoped Tag Names	12
8.2.3 Program-Scoped Tag Names	12
8.2.4 Aliases	13
8.2.5 Descriptions	13
8.3 I/O AND NETWORK MODULES	13
8.3.1 Description	13
8.3.2 Examples	15
8.4 USER DEFINED DATA TYPES (UDTs)	18
8.5 ADD-ON INSTRUCTIONS (AOIs)	18
9. LADDER LOGIC	19
10. CONVENTIONS FOR REVISION NUMBERING	20
10.1 REVISION NUMBERING EXAMPLES	20

11.	THE MAIN TASK.....	21
11.1	HEALTH AND STATUS	21
11.2	ALARMS	21
11.3	FAULT HANDLING	21
12.	THE SAFETY TASK.....	22
12.1	SAFETYPROGRAM.....	22
13.	EXAMPLE ROUTINES.....	23
13.1	MAIN ROUTINE.....	23
13.2	EMERGENCY STOP INPUT ROUTINE	24
13.3	EMERGENCY STOP ROUTINE.....	26
13.4	AXIS INTERLOCK	27
13.5	AXIS END-OF-TRAVEL ROUTINE	27
13.6	AXIS FINAL TRAVEL ROUTINE	29
13.7	AXIS ABSOLUTE ENCODER.....	31
13.8	AXIS ENABLE PENDANT	32
13.9	AXIS ENABLE OUTPUT ROUTINE	32
13.10	AXIS OVERSPEED	34
13.11	OVER TEMPERATURE	34
14.	USER DEFINED TYPES	35
14.1	UDT_SAFETYBOOL.....	35
15.	PASSWORDS	36

PREFACE

It is desirable to layout general guidelines for programming of various components of the Global Interlock System to produce a consistent and unified approach that will aid the end users in better understanding the system. Development time should also be reduced by fostering the use of re-usable code.

For the most part, these guidelines reflect Rockwell Automation's own guidelines for modular programming and sample code provided by Rockwell.

1. INTRODUCTION

1.1 PURPOSE

This document will build a foundation for consistency in programming the various components of the Global Interlock System (GIS). Nothing in the document is necessarily an absolute requirement or specification. Consult the appropriate project document, specification, or interface controller document for actual system requirements.

This document is based on standard practices that are found in Rockwell Automation's documentation for their products, most notably Integrated Architecture: Foundations of Modular Programming.

This document also gathers configuration details into one centralized location.

This document should be a collaborative effort of all vendors, design teams, and the ATST project. It is not the purpose of the document to restrict viable solutions. Comments and discussion of this document are highly encouraged.

1.2 INTENDED AUDIENCE

This document is intended primarily to be used by the designers and programmers of the individual local interlock controllers (LICs).

2. DOCUMENTS

2.1 RELATED DOCUMENTS

- “Global Interlock System Design Definition”, ATST Project Document SPEC-0112.
- “Global Interlock System Specification”, ATST Project Document SPEC-0046.

2.2 REFERENCED DOCUMENTS

- Rockwell Automation, “GuardLogix Controller Systems Safety Reference Manual”, Publication 1756-RM093.
- Rockwell Automation, “GuardLogix Controllers”, Publication 1756-UM020.
- Rockwell Automation, “Integrated Architecture: Foundations of Modular Programming”, Publication IA-RM001.
- Rockwell Automation, “Logix5000 Controllers Common Procedures”, Publication 1756-PM001.
- Rockwell Automation, “Logix5000 Controllers Ladder Diagram”, Publication 1756-PM008.
- Rockwell Automation, “Logix5000 Controllers Tasks, Programs, and Routines”, Publication 1756-PM005.
- Rockwell Automation, “58814 - Smartport assignment guidelines for Stratix 8000”
- Rockwell Automation, “65491 - Resilient Ethernet Protocol Ring Using Stratix 8000/8300”
- Rockwell Automation, “65566 - Inter-VLAN routing using Stratix 8300 and Stratix 8000 switches”

2.3 SOFTWARE

The following software will be required for GIS Development

- RSLogix 5000, version 20.01 or higher
- RSLinx Classic, version 2.57.00 or higher

The following software is recommended in addition to the required software.

- Cisco Network Assistant, version 5.6 (3) or higher
- WireShark, version 1.6.5 or higher

3. CONFIGURING THE NETWORK

3.1 INITIAL SETUP

The first step in getting the system up and running is configuring the network switch(es). Follow the procedure given in 1783-UM003 to “initialize the Switch with Express Setup.”

Configure as follows:

Network Settings	
Management Interface (VLAN ID)	Default -1
IP Assignment Mode	Static
IP Address	See Table 1 below
Subnet Mask	255.255.255.0
Default Gateway	10.4.0.200
Password	
Confirm Password	
CIP VLAN settings	
CIP VLAN	Default 1
IP Address	Blank
Subnet Mask	255.255.255.0
Optional Settings	
Host Name	See Table 1 Switch IP Address Assignments Table 1 below
System Date	
System Time	

Figure 1 Express Setup

Table 1 Switch IP Address Assignments

System	Address	Hostname
Global Interlock Controller	10.4.0.200	GIC_Switch
Telescope Mount Drive Assembly	10.4.0.201	Tel_Switch
Coudé Rotator System	10.4.0.202	Rot_Switch
Optical Support System	10.4.0.203	OSS_Switch
Instruments System	10.4.0.204	Inst_Switch
Enclosure	10.4.0.205	Enc_Switch
Facility Equipment	10.4.0.206	Fac_Switch
Facility Thermal System	10.4.0.207	FTS_Switch
Development LIC1	10.4.0.208	LIC1_Switch
Development LIC2	10.4.0.209	LIC2_Switch

3.2 SETTING UP PORTS

The Stratix 8300 has pre-configured settings (“Smartports”) that optimize the switch port for the type of device that is connected. See Rockwell knowledge base article 58814 - Smartport assignment guidelines for Stratix 8000.

Device	Smartport Role
A single 1734-AENTR module	Automation Device
A single 1791ES-IB8XOBV4 or 1791ES-IB16 module	Automation Device
Daisy-chained 1734-AENTRs (linear device network)	none
1756-EN2TR, 1756-EN2F	Automation Device
HMI terminal	Automation Device
A single PC	Desktop for Automation
1783-RMS10T Stratix 8300 switch	Switch for Automation
1783-ETAP	None



3.3 SETTING UP VLANS

Each subsystem will exist in its own virtual LAN (VLAN). This will require that each switch be configured for each VLAN. In addition, there is a 'Setup' VLAN that will facilitate connection to devices at the default IP addresses to enable configuration. Table 2 VLAN Assignments shows the VLANs that should be created on each switch.



Figure 2 VLAN Setup Screen

Table 2 VLAN Assignments

System	VLAN Name	VLAN ID
Global Interlock Controller	VLAN0100	100
Telescope Mount Drive Assembly	VLAN0101	101
Coudé Rotator System	VLAN0102	102
Optical Support System	VLAN0103	103
Instruments System	VLAN0104	104
Enclosure	VLAN0105	105
Facility Equipment	VLAN0106	106

Facility Thermal System	VLAN0107	107
Development LIC1	VLAN0108	108
Development LIC2	VLAN0109	109
Setup	VLAN0192	192

Table 3 Subnet Address Assignments

System	Address Space	VLAN
Global Interlock Controller	10.4.0.0/24	100
Telescope Mount Drive Assembly	10.4.1.0/24	101
Coudé Rotator System	10.4.2.0/24	102
Optical Support System	10.4.3.0/24	103
Instruments System	10.4.4.0/24	104
Enclosure	10.4.5.0/24	105
Facility Equipment	10.4.6.0/24	106
Facility Thermal System	10.4.7.0/24	107
Development LIC1	10.4.8.0/24	108
Development LIC2	10.4.9.0/24	109
Setup	192.168.0.0/16	192

4. CONFIGURING THE GUARDLOGIX CONTROLLER

4.1 FIRMWARE

All controller will need to have version 20 of the appropriate firmware installed. The controller will ship with minimal firmware to allow initial configuration and installation of a fully functional version of the firmware.

4.2 SAFETY NETWORK NUMBER

The Safety Network Number (SNN) is a unique number that is generated for each network segment. It is used by the Safety CIP protocol to ensure that the communicating device is indeed the correct device, not just a similar one.

For the purposes of GIS, the automatic, time-based SNN should be sufficient. This should ensure that no two devices share a unique SNN. However, there is a remote but non-zero chance that two units could be configured at different locations by different developers at the exact same moment. Prior to commissioning the entire system, these numbers will need to be verified for their uniqueness.

4.3 SAFETY LOCKING

Safety-locking the controller helps protect safety control components from modification. This feature requires two separate passwords, one for lock and one for unlock.

See section 15 for password requirements.

Typically, safety-locking will only be required once development is completed on a particular revision and it is ready for verification and validation.

4.4 I/O MODULES

4.4.1 Module Definitions

I/O modules should be configured for providing combined status.

Electronic Keying should be set for 'Exact Match.'

Requested Packet Interval (RPI) should be set for 20 milliseconds.

'Major Fault on Controller if Connection Fails While in Run Mode' should not be set.

Input configuration should be set for single. All inputs will use dual channel input instructions for safety.

Test outputs should be configure as required by how the module is wired to the field devices. Typically, this will be set for pulse test.

Output configuration should be set for single point operation type. Typically, this will be set for Safety Pulse test point mode.

Since the controller and all remote I/O devices are part of the same network the safety network number should be the same as the safety network number of the controller.

5. CONFIGURING REMOTE I/O MODULES

5.1 FIRMWARE

All remote I/O modules will need to have version 20 firmware installed.

6. IP ADDRESSING

The GIS is assigned the 10.4.0.0/16 subnet. Each subsystem of the GIS will receive its own /24 network.

6.1 VIRTUAL LANS

Each of the subnets will be its own virtual LAN.

Table 4 Subnet Address Assignments

System	Address Space	VLAN
Global Interlock Controller	10.4.0.0/24	100
Telescope Mount Drive Assembly	10.4.1.0/24	101
Coudé Rotator System	10.4.2.0/24	102
Optical Support System	10.4.3.0/24	103
Instruments System	10.4.4.0/24	104
Enclosure	10.4.5.0/24	105
Facility Equipment	10.4.6.0/24	106
Facility Thermal System	10.4.7.0/24	107
Development LIC1	10.4.8.0/24	108
Development LIC2	10.4.8.0/24	109
Setup	192.168.1.0/24	192

6.2 HOST ADDRESS

For uniformity the following number scheme is suggested. Replace LIC1 with the appropriate abbreviation from Table 6 LIC Abbreviations. Replace *x* with the third octet from Table 4.

Table 5 Host Address Assignments

IP Address	Hostname	Description
10.4.x.0/24		Local Interlock Controller subnet
10.4.x.1	LIC_Gateway	LIC Stratix 8300 switch Gateway
10.4.x.10	LIC_R00_S02_ENET	first Ethernet adapter
10.4.x.11	LIC_R00_S03_ENET	second Ethernet adapter (if installed)
10.4.x.20		Second ControlLogix rack (if installed)
10.4.x.80	GIS_PVP	Reserved for HMI
10.4.x.90	GIS_Dev	Reserved for Development Station
10.4.x.91	GIS_Maint	Reserved for Maintenance Station
10.4.x.101	LIC_R01_S00_ENET	Remote I/O adapter (Rack #01)
10.4.x.102	LIC_R02_S00_ENET	Remote I/O adapter (Rack#02)
10.4.x.103	LIC_R03_S00_ENET	Remote I/O adapter (Rack #03)
...
10.4.x.199	LIC_R99_S00_ENET	Remote I/O adapter (Rack #99)
10.4.x.201	LIC_Switch201	Embedded Switch #01

IP Address	Hostname	Description
10.4.x.202	LIC_Switch202	Embedded Switch #02
10.4.x.203	LIC_Switch203	Embedded Switch #03
...
10.4.x.254	LIC_Switch254	Embedded Switch #54

7. TAGS

7.1 TAG SCOPE

All produced and consumed tags are controller-scoped tags. Controller-scoped tags represent information that must be passed between the GIC and the LIC. Program-scoped tags represent information that is only required at the LIC. Examples of controller-scoped tags would be emergency stop status signals.

7.2 PRODUCED AND CONSUMED TAGS

The entire Global Interlock System will produce and consume a large number of tags. Because of the limited resources available to handle these connections it may be necessary to use user-defined types to aggregate the tags into larger structures.

Tags that will be consumed by the individual LICs from the GIC include emergency stop, fire alarm, and seismic alarm.

7.3 DATA ACCESS CONTROL

Starting with revision 18 of the RSLogix software, two new tag attributes are available: External Access and Constant.

7.3.1 External Access

External Access attribute defines how an external application can access a tag. Options you can configure for the External Access attribute include:

- Read/Write
- Read Only
- None

By default, all tags external access attribute should be set to None. No tags in the Safety Task should be set to Read/Write.

Program-scoped tags should typically be set to Read Only.

7.3.2 Constant

The Constant attribute is used to protect the tag from being changed via the controller programming application or logic in the controller.

8. GENERAL NAMING CONVENTIONS

8.1 PROGRAMS AND ROUTINES

- Names should be meaningful to maintenance personnel.
- Keep names short by using abbreviations and acronyms.
- Spaces are not allowed, instead of using underscores (“_”) use mixed case.
- Use standard industry abbreviations or abbreviations listed in this document.
- When using abbreviations be consistent, clear, and unambiguous.

8.2 TAG NAMING

RSLogix enforces the following rules:

- A name may not exceed 40 characters.
- A name can contain only upper-case and lower-case letters, numbers and underscores.
- A name cannot begin with a number.
- A name cannot have adjacent underscores or end with an underscore.
- Underscores are significant.

Case is not significant (lower-case letters match upper-case letters), and names are displayed with the case entered when first created. The use of mixed case, sometimes referred to as camel case is desirable to increase legibility.

8.2.1 General Tag Name Guidelines

Care should be taken when choosing names to avoid ambiguity. For example the tags *DoorOpen* and *OpenDoor*, each could alternately refer to a door being ajar or a command to open a door. Therefore, such tags should be avoided in favor of tags such as *DoorOpened* and *CmdOpenDoor*.

8.2.2 Controller-Scoped Tag Names

Controller-scoped tags will take the format of *major_minor_component_signal*.

major indicates the major subsystem that the device is located in. This is the controller that “owns” the tag. This is the LIC abbreviation found in Table 6 LIC Abbreviations.

minor indicates the subsystem that the device is associated with, such as *Az* or *El* for azimuth and elevation drive subsystems.

component indicates the physical device which the tag is associated with. This could be an individual drive in a subsystem, such as *Drive1*.

signal is the particular status or information that the tag indicates, such as *InputOK* or *DoorOpened*

It is often desirable for the data type to be included in the name. Because data types often contain several pieces of information not just a BOOL value.

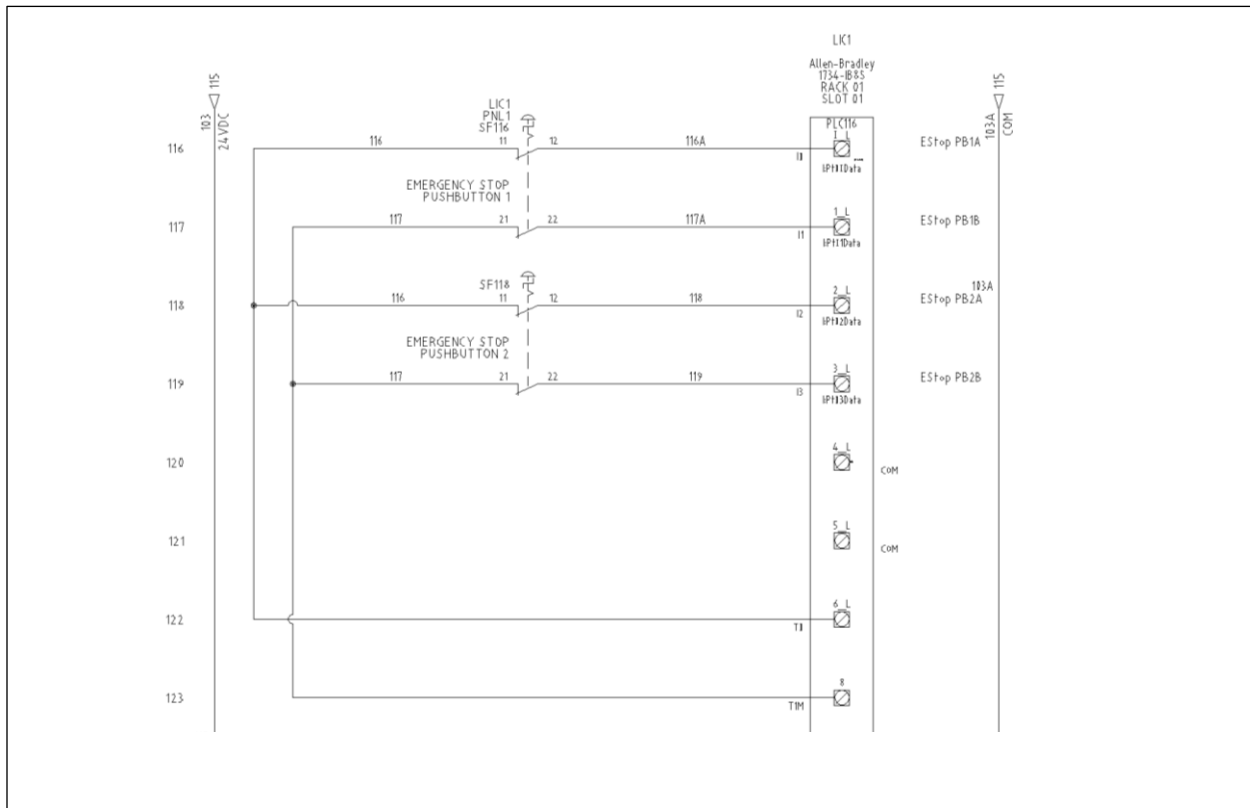
8.2.3 Program-Scoped Tag Names

Program-scoped tags, because they are used only within a single LIC have no need of the elaborate structure given in Controller-Scoped Tag Names above.

8.2.4 Aliases

Aliases allow individual tags to be referenced by various nomenclatures. In addition to the base tag, these can include functional names, and even references to schematic diagrams. Tags can even be double aliased.

For example: The first contact block on the emergency stop switch -SF116 is connected to input 0 on the POINT I/O module in slot 01 of rack 01.



LIC1_R01:2:I:PtData01 can be aliased as `_SF116_1` which can be further aliased as `EStopPB1A`. This way a technician referencing the ladder logic can know the actual I/O point, the connected component that the I/O is referencing, and also the logical name.

8.2.5 Descriptions

Descriptions should avoid simply repeating the tag name, but rather more fully describe what the tag represents in the logic. While tag names were chosen to be as clear and concise as possible, clear descriptions will help remove any ambiguity for a technician who may be troubleshooting a problem.

In the *DoorOpened* example from above, it could clearly indicate which door and that this is an input, “Access Door to Telescope Level is not fully closed and locked.”

8.3 I/O AND NETWORK MODULES

8.3.1 Description

Each module in a RSLogix project requires a unique name. By default, I/O and network modules should be named as follows:

LIC_Rnn_Snn_function

Where *LIC* represents the ATST system and LIC to which the component belongs. There are seven LICs in the GIS as listed in Table 6 below.

Rnn is the chassis or remote adapter number. R00 is the local rack, R01 and R02 would be remote adapters #1 and #2.

Snn is the slot number. S00 is slot or module #0 (far left), S01 and S02 would be slot #1 and #2 or module #1 and #2.

function is an abbreviation for the type of module as listed in Table 8 (taken from “Integrated Architecture: Foundations of Modular Programming”).

Strictly adhering to this scheme is not always desirable because it can make for some convoluted and potentially confusing naming. For example, the second input module on a Point I/O chassis would be LIC2_R01_S00_ENET:2.I. In which case, it would be better to not use the slot number and function. This would become LIC2_R01:2.I.

For this reason it is suggested that ‘slot zero,’ the controller or communications adapter, in a chassis not use slot number or function, but rather the logical location or communications adapter name, such as LIC2_G LX would be the GuardLogix controller in LIC2, while LIC1_R01 would be the first remote I/O chassis in the LIC1 system. The rack designation, R01, might also be replaced with a more appropriate identifier, such as

Table 6 LIC Abbreviations

System	Abbreviation
Global Interlock Controller	GIC
Telescope Mount Drive Assembly	Tel
Coudé Rotator System	Rot
Optical Support System	OSS
Instruments System	Inst
Enclosure	Enc
Facility Equipment	Fac
Facility Thermal System	FTS

Table 7 System Abbreviations

System	Abbreviation
Global Interlock Controller	GIC
Telescope Mount Assembly	TMA
M1 Assembly	M1
Top End Optical Assembly	TEOA
Feed Optics	FO
Wavefront Correction	WFC
Observatory Control System	OCS
Telescope Control System	TCS
Enclosure Control System	ECS
Facility Equipment	Fac
Facility Thermal System	FTS

Table 8 Function Abbreviations

Function	Abbreviation	Example Module
Analog Input	AI	
Analog Output	AO	
Discrete Input	DI	1734-IB8S
Discrete Output	DO	1734-OB8S
Analog Input/Output combo	AIO	
Discrete Input/Output combo	DIO	1791ES-IB8XOBV4
Analog/Discrete Input/Output combo	ADIO	
Remote I/O	RIO	
Serial data I/O	SIO	
Motion I/O	MIO	
DeviceNet	DNET	
ControlNet	CNET	
EtherNet/IP	ENET	1734-AENTR, 1756-EN2TR
Profibus	PFB	
High Speed Counter	HSC	
Programmable Limit Switch	PLS	
Sequence Of Events	SOE	
GuardLogix Processor	GLX	1756-L62S
ControlLogix Processor	CLX	1756-L55

8.3.2 Examples

Global Interlock Controller rack contains two Ethernet cards, one (in slot #2) for communication with the GIS and the other (in slot #3) for communication with the OCS, they would be called GIC_R00_S02_ENET and GIC_R00_S03_ENET, respectively.

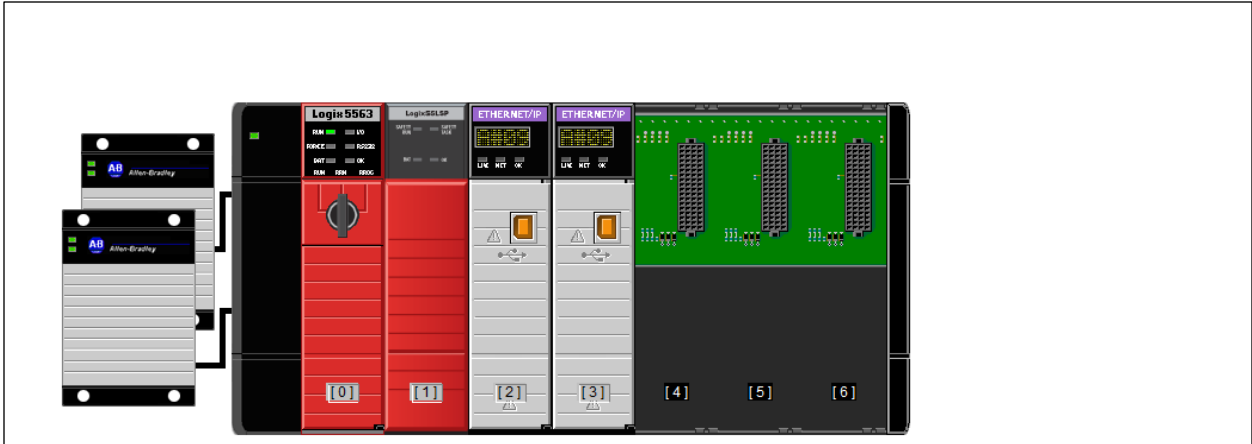


Figure 3 Global Interlock Controller

Module Name	Description
GIC_GLX	1756-L63S Controller
GIC_R00_S02_ENET	1756-EN2TR Ethernet adapter in GIC rack slot 2
GIC_R00_S03_ENET	1756-EN2TR Ethernet adapter in GIC rack slot 3

Note that the controller does not specify a rack or slot position, regardless of where it is located as controller have a slightly different naming convention.

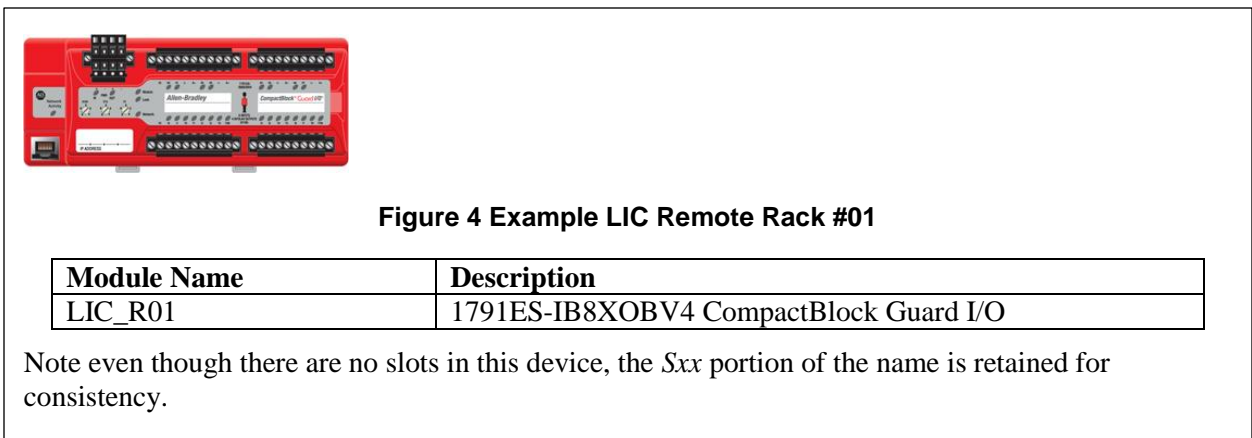
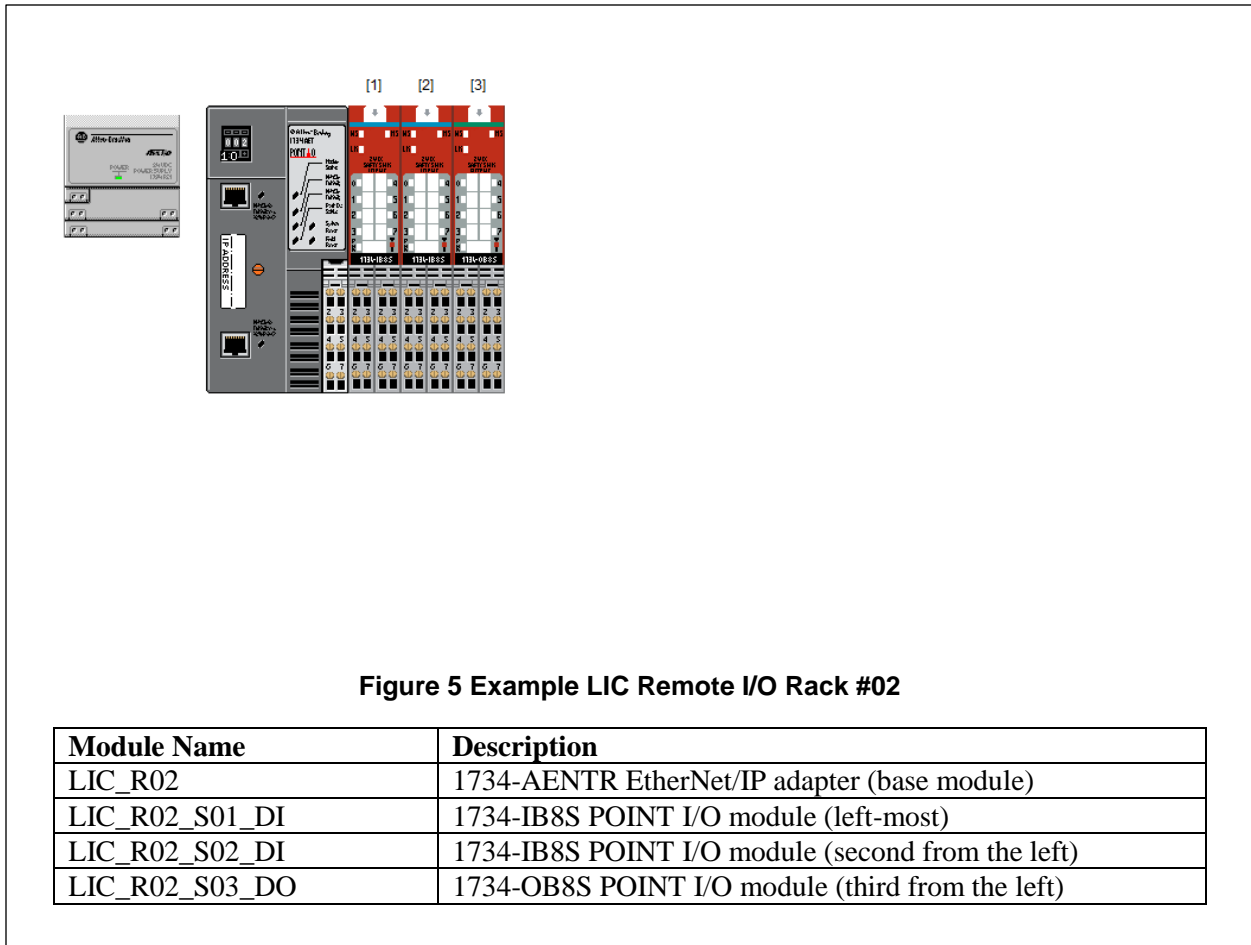
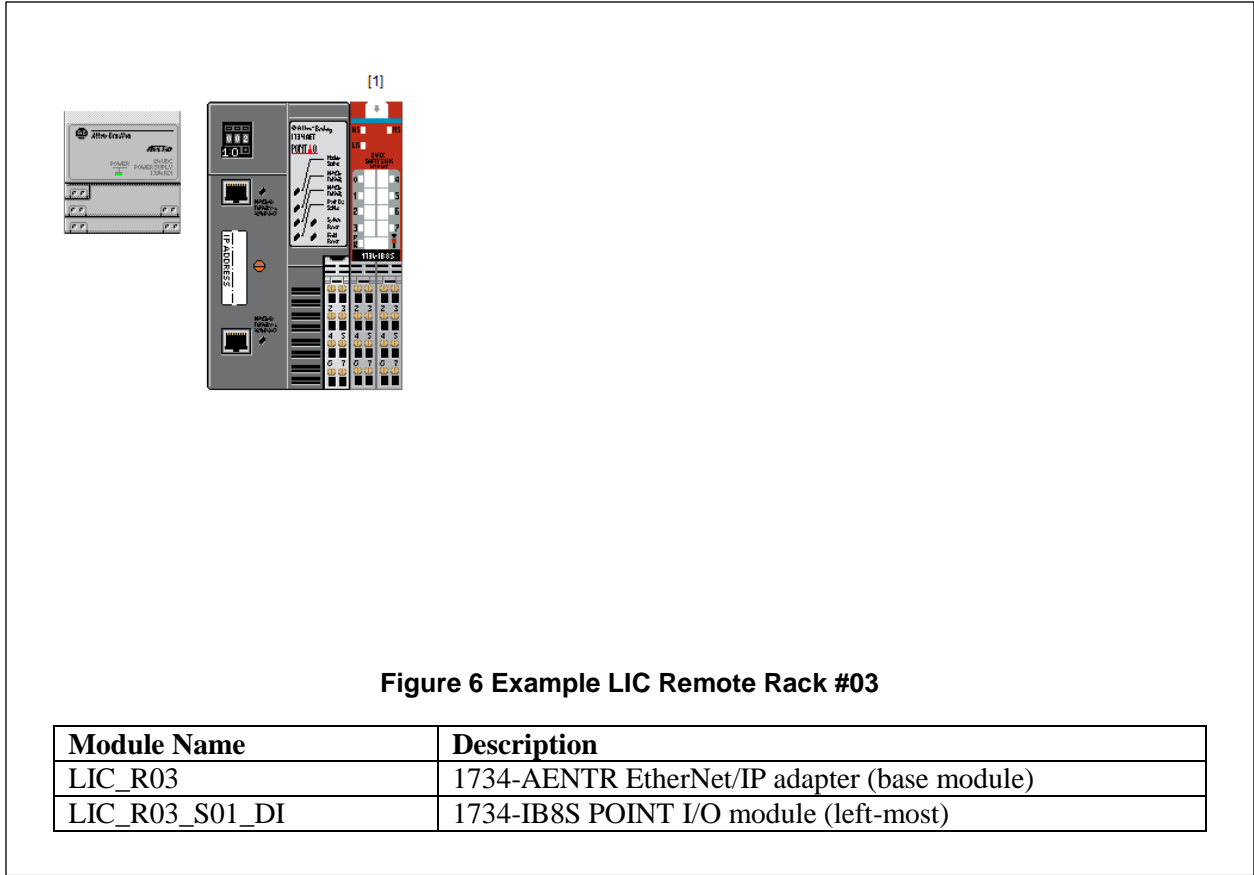


Figure 4 Example LIC Remote Rack #01

Module Name	Description
LIC_R01	1791ES-IB8XOBV4 CompactBlock Guard I/O

Note even though there are no slots in this device, the Sxx portion of the name is retained for consistency.





8.4 USER DEFINED DATA TYPES (UDTs)

All user-defined data types will begin with “UDT_.”

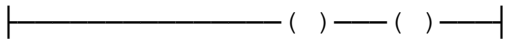
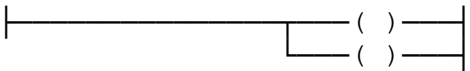

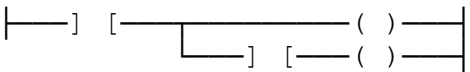
8.5 ADD-ON INSTRUCTIONS (AOIs)

All add-on instructions will begin with “AOI_.”

9. LADDER LOGIC

All programming for the Global Interlock system will be Ladder Logic.

Although output instructions can be placed in sequence on a single rung (serial), it is preferred that they are placed in branches (parallel) to assist technicians viewing the ladder logic code. The same applies for mixing input and output instructions on the same line. It is preferred to place the output instruction at the right-most side and create parallel branches as needed. These arrangements are commonly found in hardware relay logic.

Possible Option	Preferred
	
	

It is preferred that the instruction most likely to be FALSE be placed on the left and the instruction least likely to be FALSE be placed on the right. Typically instructions are executed faster when the rung condition is FALSE.

10. CONVENTIONS FOR REVISION NUMBERING

Revision number will apply to all software components as follows:

Program_Name M.n-TT.PP

M, a major revision number, which changes when a major set of components is released as a set.

n, a minor revision number, which changes when a when the interface to the component changes.

TT a “tweak” revision number, which changes when any change at all is made

and *PP* any system specific revision number.

PP is unlikely to change for the top level programs as there aren’t any system specific revisions (unless a second ATST is built). Rather the *PP* suffix is useful when producing and sharing routines between subsystems, which require a change from the standard routine that is specific to the subsystem. *PP* will normally be omitted

10.1 REVISION NUMBERING EXAMPLES

Program Name	
GIC_GLX 1.0-00.00	First major release of the GIC code
GIC_GLX 1.0-01.00	First revision, which doesn’t change its interface
GIC_GLX 1.1-00.00	next revision which affects its interface with other systems

11. THE MAIN TASK

The main task does not run on Safety Partner. Safety-related tasks must not be run in the main task.

The main task will handle general controller faults; system health and status updates; and alarms to the rest of the GIS.

11.1 HEALTH AND STATUS

The controller will monitor itself for general health and produce an alarm if certain thresholds are reached. These should not be safety-related but may indicate a need for service that does not require immediate attention.

One such example is the program backup battery. Another example is redundant power supplies. While a failure of even the second power supply will not result in an unsafe condition this information needs to signal a minor alarm to the system, so that maintenance personnel can address the issue.

11.2 ALARMS

While the interlocking of the safety system is handled in the Safety Task, alarms for display at the HMI can be handled in the standard task at a much more leisurely update rate. These alarms will provide specific indications of devices which have been activated or are in a faulted condition.

The alarms are to alert the operator to what interlocks have been asserted and the corrective action that must be taken to reset the interlock. Additionally information in the alarms will assist in troubleshooting in the event of faulted conditions.

11.3 FAULT HANDLING

Specific routines will handle controller faults. Generally, these will be to generate an alarm and provide status about the controller fault.

There may be some controller faults that are considered so minor that the fault-handling routine will clear them so that normal operation may continue after notifying the operator.

12. THE SAFETY TASK

GuardLogix supports only one Safety Task, called “SafetyTask.” There can be multiple safety programs composed of multiple safety routines. There is a limit of 32 programs in the safety task.

There should be one safety program per *independent* subsystem. For example, the Enclosure LIC handles both the Carousel and the Coudé Rotator; these are both independent subsystems controlled by the same LIC. On the other hand, the Telescope Mount LIC controls both Az/El motion of the telescope. These are not truly independent.

You cannot schedule standard programs or execute standard routines within the Safety Task.

12.1 SAFETYPROGRAM

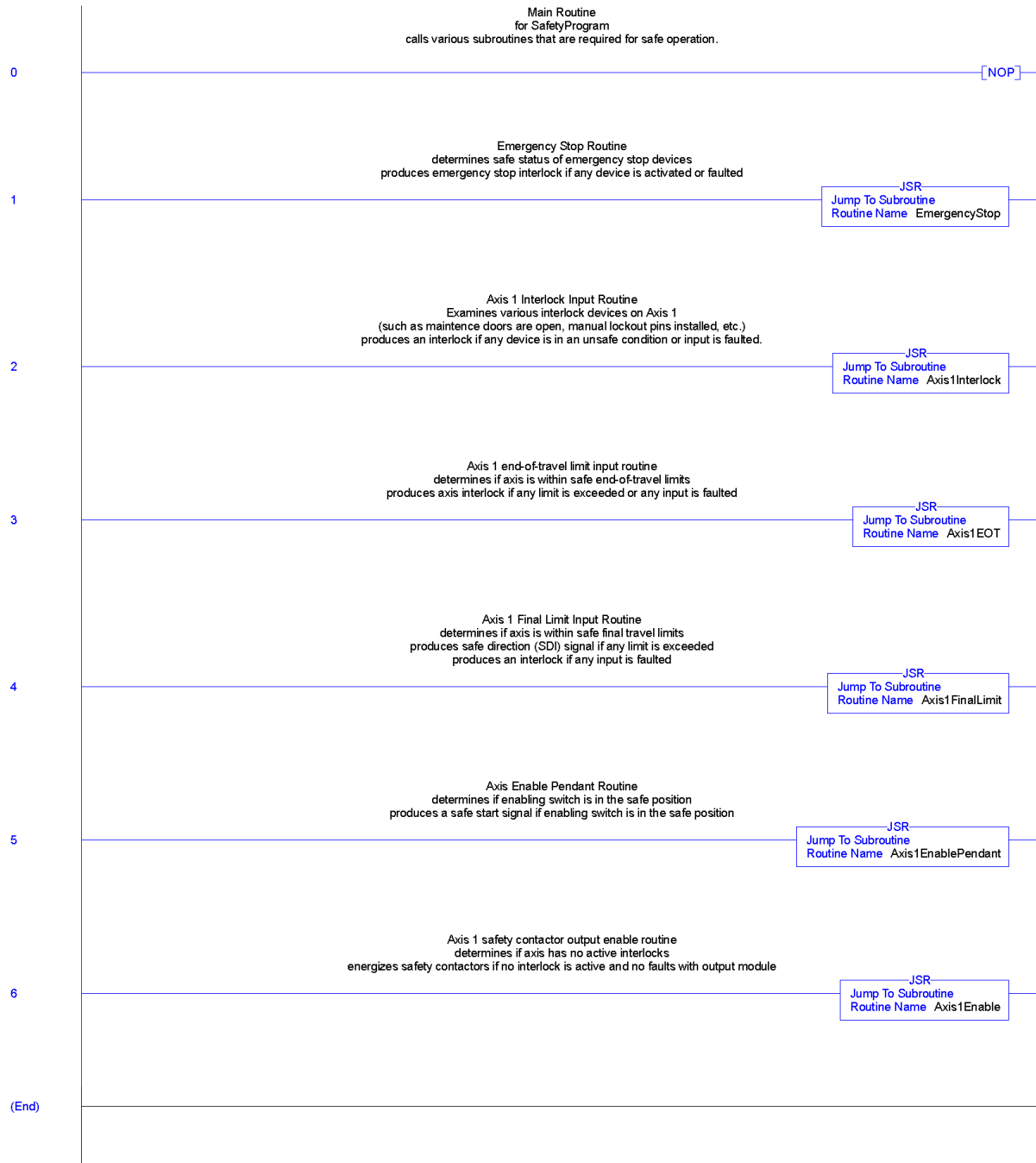
The Safety Program consists of various routines. For each safety function that the LIC performs, there will typically be two single safety routines, one input routine and one output routine. It will be common for several input routines to share output routines. For example, the Emergency Stop input routine will likely be influence every output routine; while output routines will often be influence by a limit (such as and end-of-travel) as well as Emergency Stop.

For some complex functions, such as for emergency stop monitoring, may be broken up into several routines as there may be a large number of emergency stop switches, with each routine handling a group of related switches.

13. EXAMPLE ROUTINES

13.1 MAIN ROUTINE

The main routine of the safety program will consist almost entirely of calls to the various input and output routines.

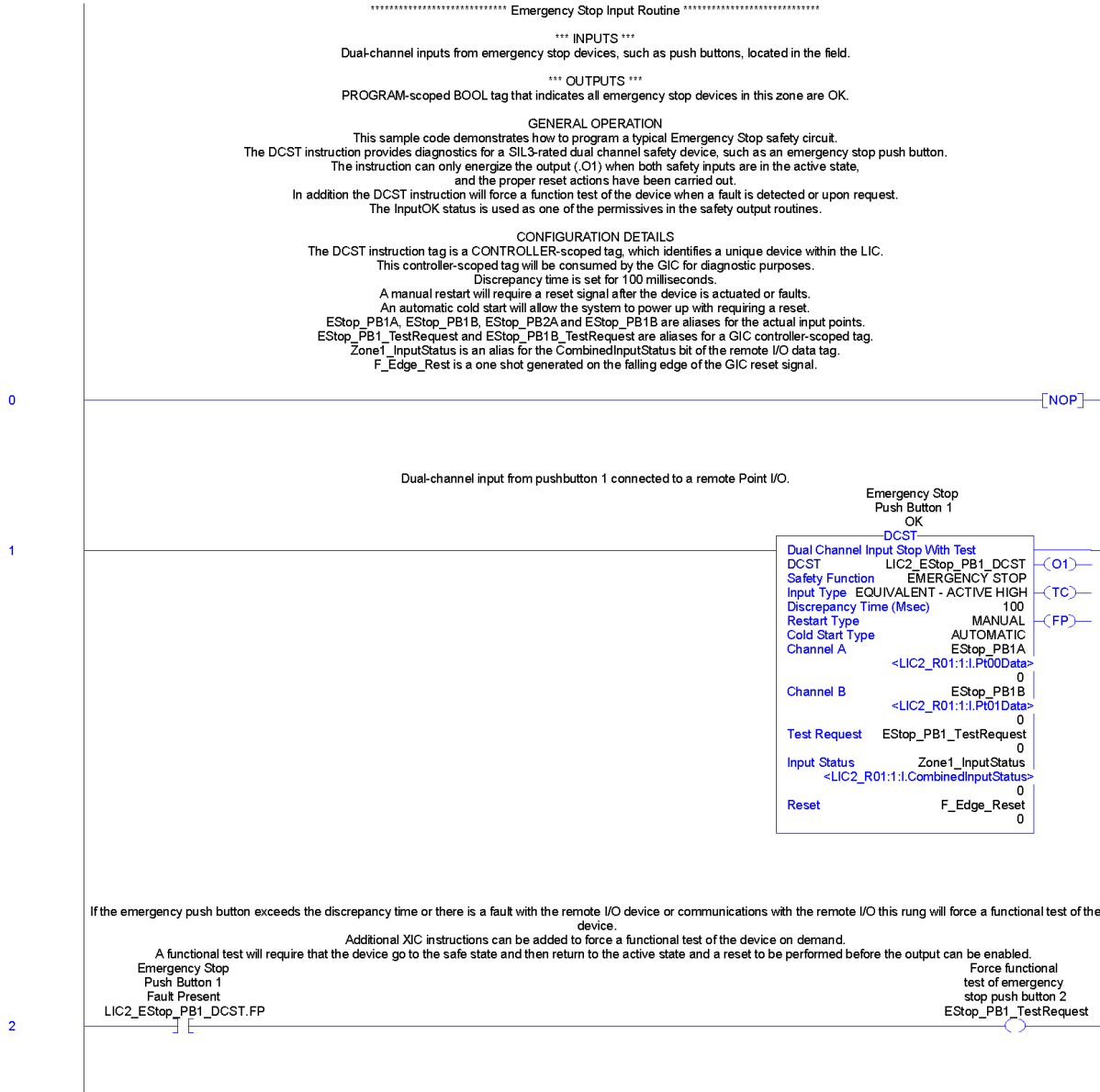


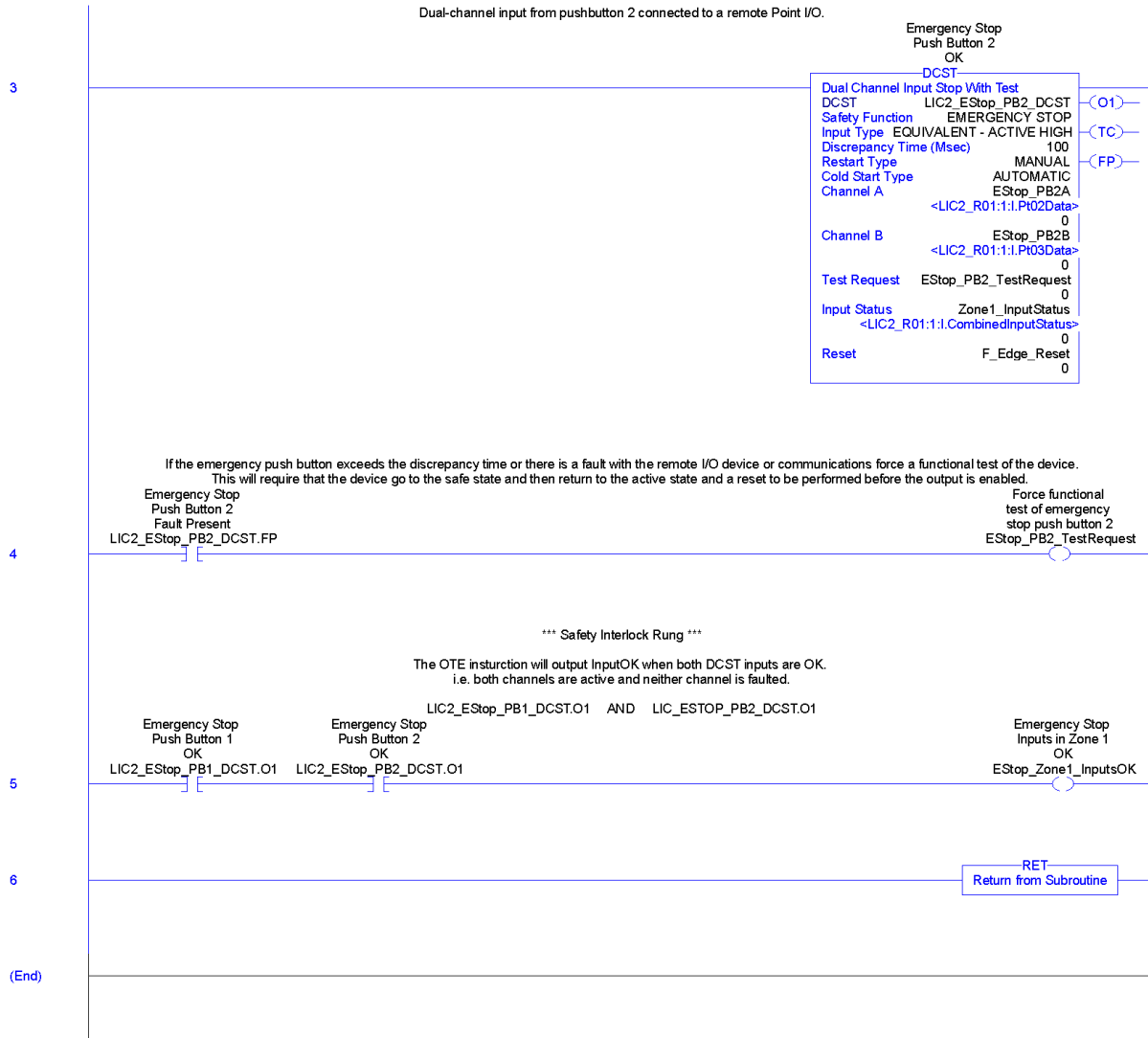
This routine can be imported from MainRoutine_RTN_1.0-00.L5X.

13.2 EMERGENCY STOP INPUT ROUTINE

This routine monitors input from switches wired to remote I/O points and produces a status signal to indicate when no emergency stop switches have been activated and there are no faults with input switches or communications from the remote I/O.

There may be several of these routines in a given LIC's safety program. These routines will cover separate 'zones' which could be any logical subdivision of the various switches. A 'zone' could be all the switches attached to one remote I/O block or could be all the switches located on one level of the facility. The purpose of this division is simply to keep the routine size at a manageable size.



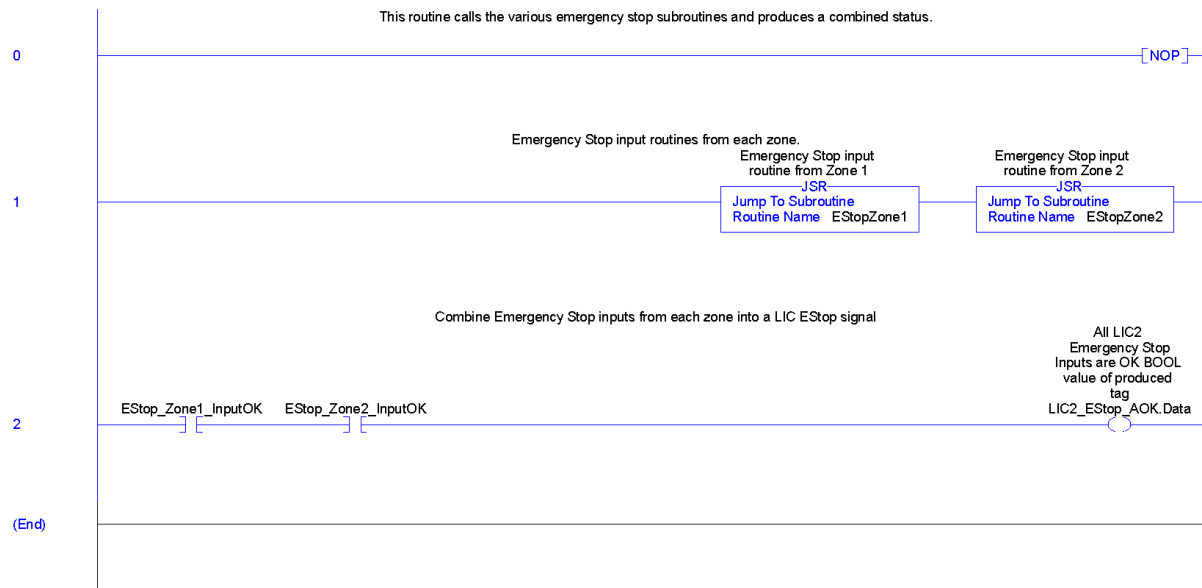


This routine can be imported from EStopInput_RTN_1.0-00.L5X.

13.3 EMERGENCY STOP ROUTINE

This routine calls the emergency stop input routines (see 13.2) and produces a status signal to indicate that all emergency stop zone indicate that no emergency stop switches have been activated and there are no faults with input switches or communications from the remote I/O.

If there is only one emergency stop input routine then the JSR instructions in rung 1 can be placed in the MainRoutine and the XIC and OTE instructions in rung 2 can be placed in the emergency stop input routine.



This routine can be imported from EmergencyStop_RTN_1.0-00.L5X.

13.4 AXIS INTERLOCK

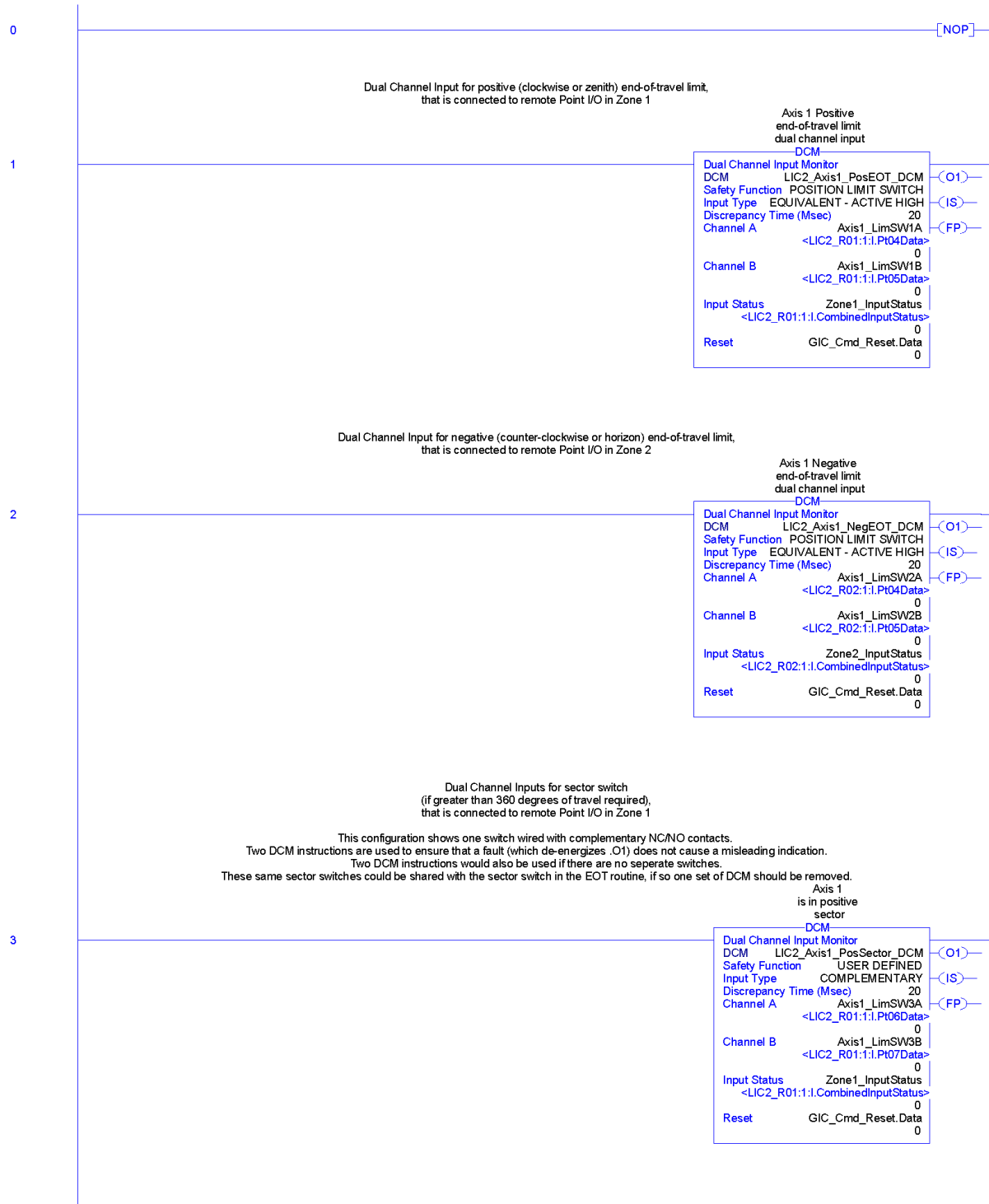
The routine handles the various interlocks that inhibit contrary motion. Just as there may be multiple routines for handling of emergency stop, there may be multiple routines for handling the various interlocks.

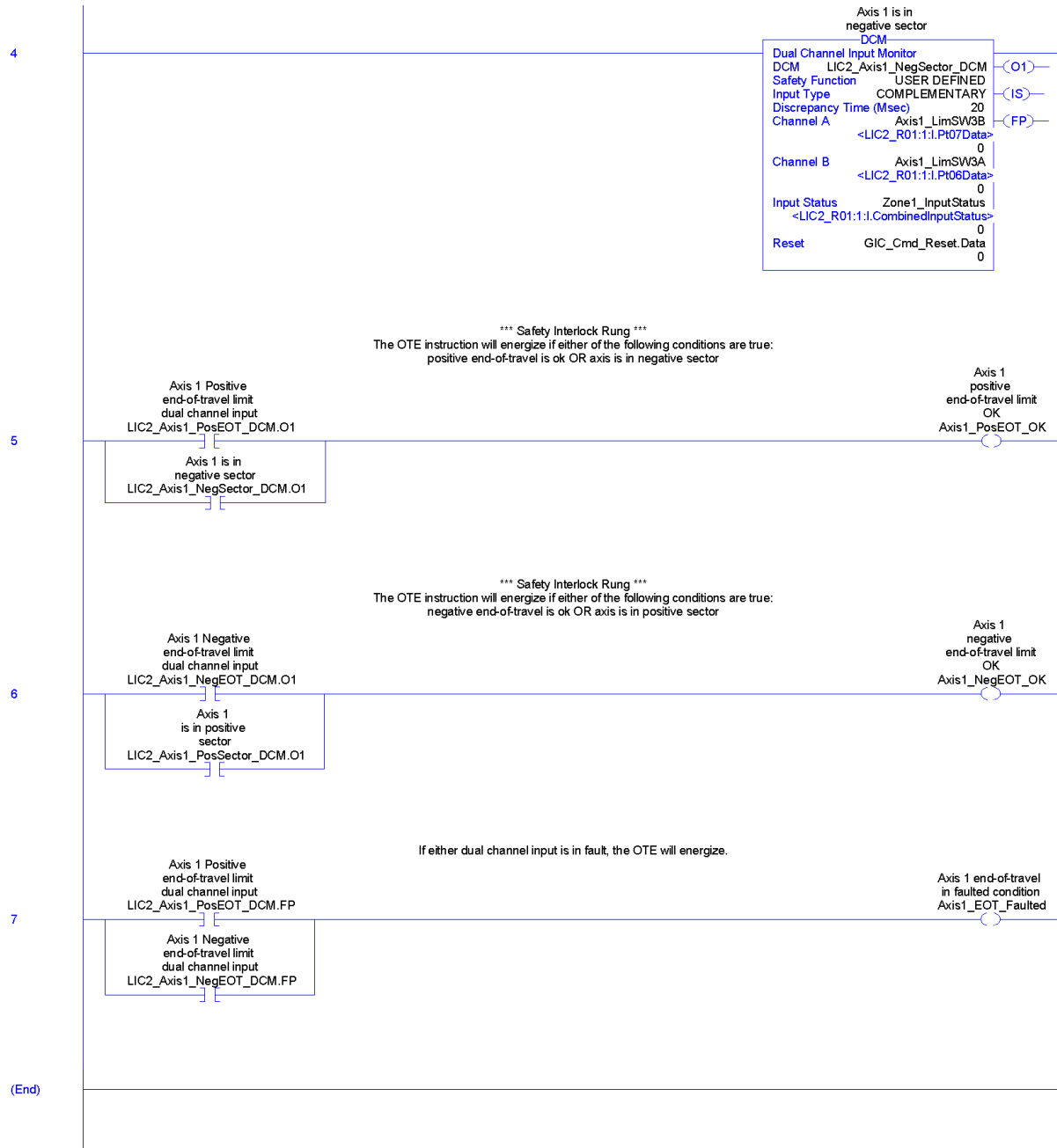
13.5 AXIS END-OF-TRAVEL ROUTINE

This routine handles the typical end-of-travel limit switches that are installed on various axes of motion. This routine monitors input from limit switches that are wire to remote I/O points and produces a signal to indicate if the axis has exceeded its end-of-travel limits are there are no faults with the limit switches or communications from the remote I/O.

In the event of end-of-travel limit being detected the Axis will produce a Safe Direction signal. On a fault this routine will instead stop motion.

There is only one end-of-travel routine for each axis. This routine is designed to monitor both end-of-travel limit switches plus and additional 'sector' switch to allow for rotary axes with more than 360° of rotation. An example of a sector switch would be switch the indicated the position of an end stop.



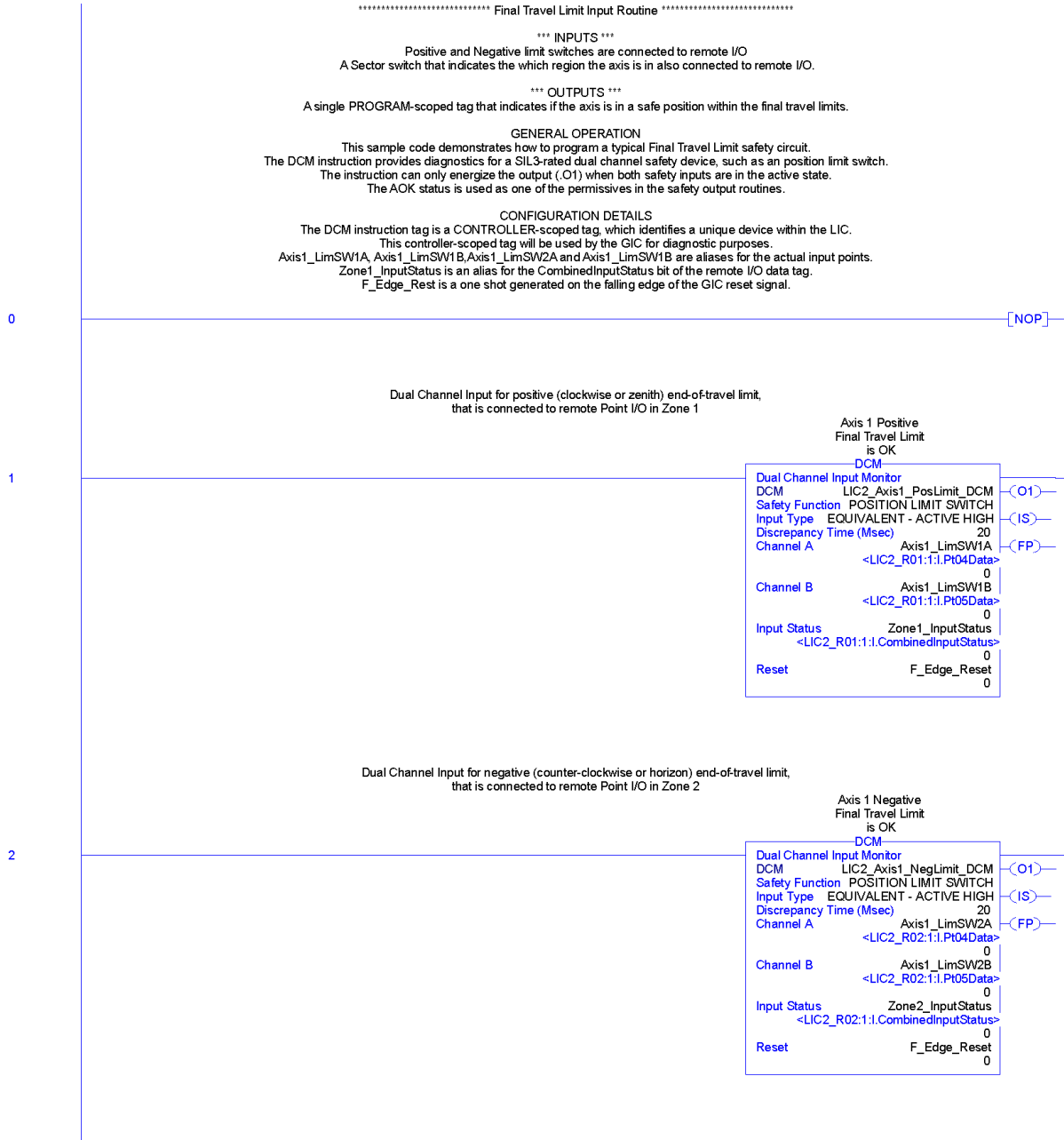


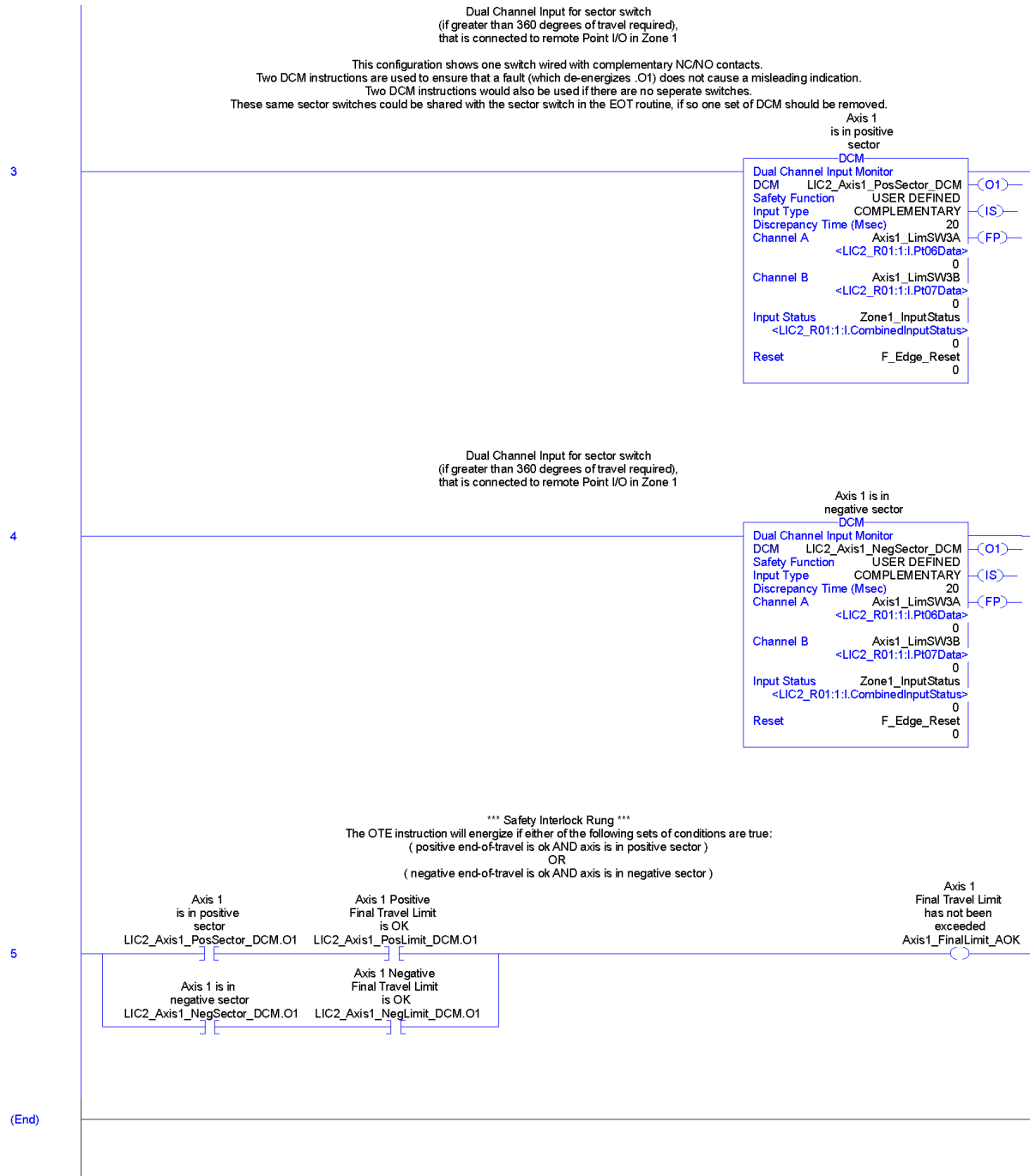
This routine can be imported from AxisEOTLimit_RTN_1.0-00.L5X.

13.6 AXIS FINAL TRAVEL ROUTINE

This routine handles the typical final travel limit switches that are installed on various axes of motion. This routine monitors input from limit switches that are wire to remote I/O points and produces a signal to indicate when the axis is between its final travel limits there are no faults with the limit switches or communications from the remote I/O.

There is only one final travel limit routine for each axis. This routine is designed to monitor both end-of-travel limit switches plus an additional 'sector' switch to allow for rotary axes with more than 360° of rotation. An example of a sector switch would be switch the indicated the position of an end stop.





This routine can be imported from AxisFinalLimit_RTN_1.0-00.L5X.

13.7 AXIS ABSOLUTE ENCODER

Instead of using discrete limit switches, the end-of-travel and final limit functions could be based on a safety-rated absolute encoder.

This routine can be imported from AxisAbsoluteEncoder_RTN_1.00-00.L5X.

13.8 AXIS ENABLE PENDANT

This routine handles the use of an enabling switch to allow Safe Limited Speed motion of an axis.

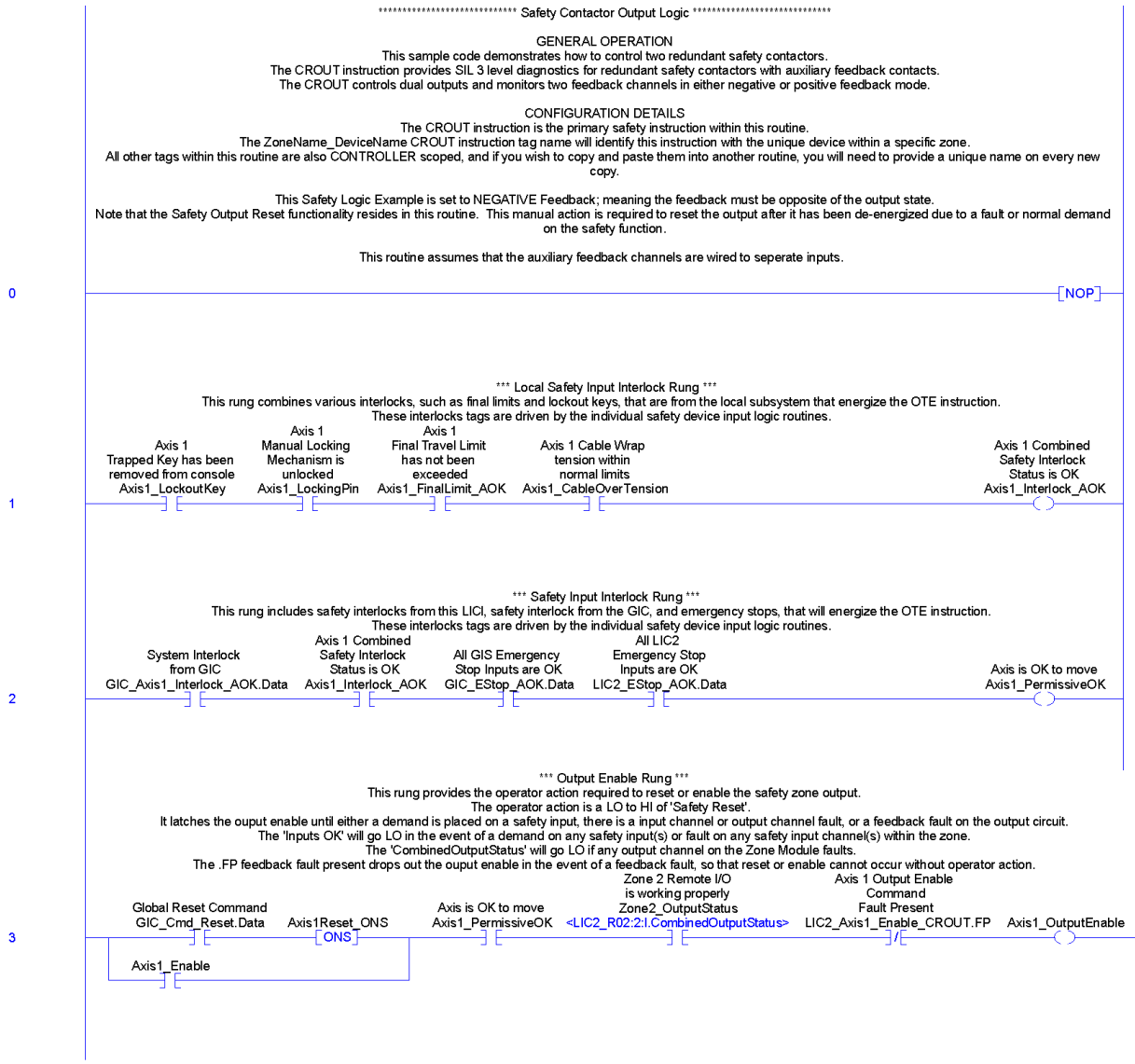
13.9 AXIS ENABLE OUTPUT ROUTINE

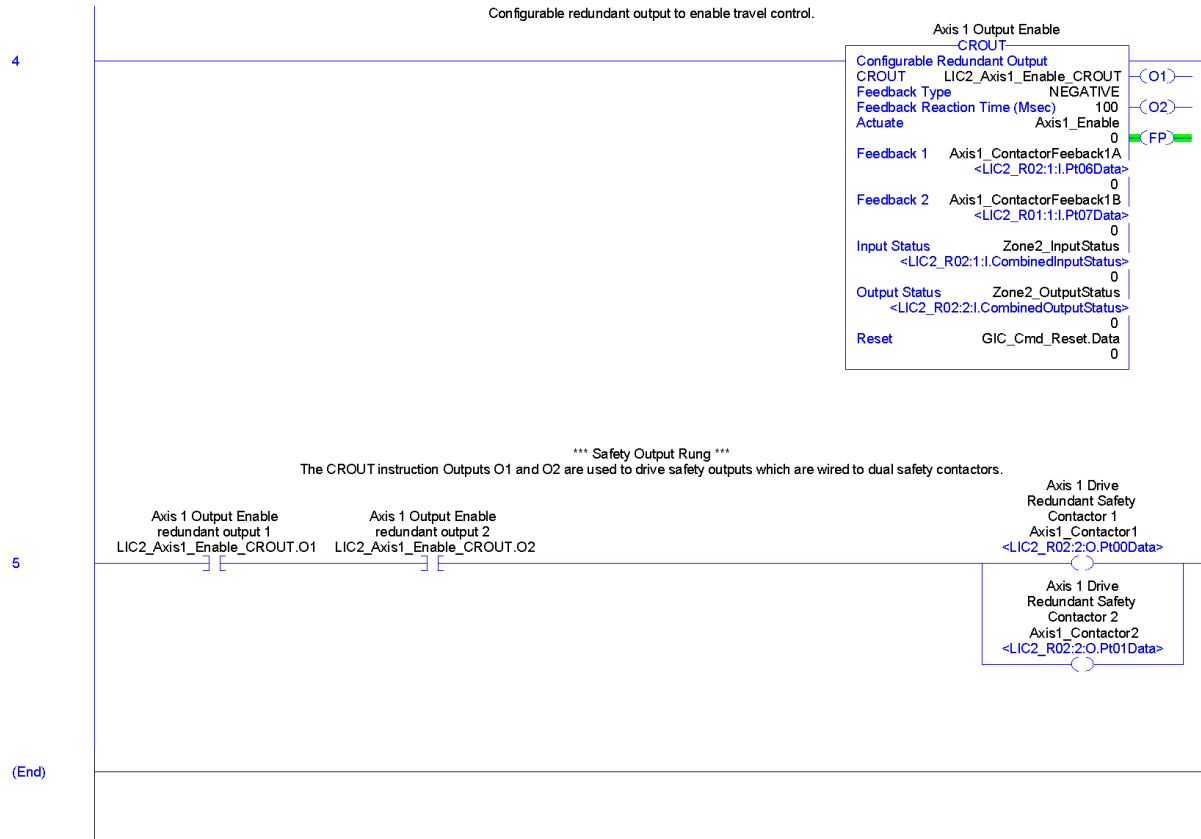
This routine monitors the conditions that are required to enable the output. This routine is designed to energize dual safety contactors to provide power to the axis' actuators.

The interlock rungs contain the conditions that must be met in order to enable axis motion. In the example routine, this includes axis within final travel limits, no system interlock from the GIC, no local axis interlock, no global emergency stop, and no local emergency stop.

Additional interlock rungs can be added to avoid extremely long rungs. This will help in legibility by dividing up the various interlocks into manageable sections.

The safety output instruction includes self-monitoring which will disable the output and prevent re-enabling the output in the event of a fault or loss of communications.





This routine can be imported from `AxisEnable_RTN_1.0-00.L5X`.

13.10 AXIS OVERSPEED

This routine monitors the speed of an axis and determines if the axis has exceeded the Safe Maximum Speed or Safe Limited Speed.

The Logix platform does not currently have a built-in overspeed monitoring function. An external module, such as the GuardMaster MSR57P Speed Monitoring Safety Relay, will need to be utilized. See Rockwell Automation Publication SAFETY-AT025, Using the MSR57 in a Safety Architecture to Monitor Machine Motor Speed for details regarding this implementation.

This routine can be imported from `AxisOverspeed_RTN_1.0-00.L5X`.

13.11 OVER TEMPERATURE

This routine monitors two analog values. Based on the range and tolerance, it compares the two values for agreement. It also has a discrepancy time during which the two values are allowed to diverge. It outputs an OK signal indicating the two values are in agreement and outputs a faulted signal which indicates the two values have exceeded the allowable difference in values for longer than the discrepancy time.

This is a SIL 2 routine.

14. USER DEFINED TYPES

14.1 UDT_SAFETYBOOL

For produced/consumed safety tags it is necessary to create a user-defined type. UDT_SafetyBOOL consists of CONNECTION_STATUS and a BOOL.

Name:

Description:

Safety binary bit includes CONNECTION_STATUS

Members: Data Type Size: 8 byte(s)

	Name	Data Type	Style	Description	External Access
<input type="checkbox"/>	Connection_Status	CONNECTION_STATUS			Read/Write
<input type="checkbox"/>	RunMode	BOOL	Decimal		Read/Write
<input type="checkbox"/>	ConnectionFaulted	BOOL	Decimal		Read/Write
<input type="checkbox"/>	Data	BOOL	Decimal		Read/Write
<input type="checkbox"/>					

This data type can be imported from UDT_SafetyBOOL_UDT_1.0-00.L5X.

15. PASSWORDS

15.1 SUGGESTED BEST PRACTICES

Default passwords are never to be used.

The same password is not to be used for multiple purposes; such as using the same password for both the lock and unlock function in a GuardLogix controller, or using the same password for all the EtherNet routers. In addition to making a more secure system this also helps prevent inadvertent configuration/programming changes.

Passwords should meet the following complexity requirements:

- should not contain all or any part of the user account name
- should be at least six characters long
- should contain characters from three of the following four categories:
 - unaccented uppercase characters (A to Z)
 - unaccented lowercase characters (a to z)
 - numerals (0 to 9)
 - non-alphanumeric characters (!, @, #, %)

APPENDIX A WORKSTATION SETUP

A-1 ISOLATED NETWORK

The easiest, safest, and most secure method is to completely isolate the Safety Network by disconnecting the development workstation from the facility LAN and Internet. There will be no duplication of IP addresses or unwanted traffic entering or leaving the Safety Network. This of course limits functionality of the development workstation and will require the implantation of “SneakerNet” i.e. using portable flash drives to carrying information between the networks. This method ensures that IP address will not be duplicated on the facility LAN.

A-2 CONFIGURING A SECOND NETWORK CARD

When developing applications, it is often desirable to be able to access the both the normal facility LAN and the Safety Network. I’ve used a method that utilizes a second network interface card (NIC). This ensures that no one can directly connect to the Safety Network (other than the development workstation).

This method cannot be used if development station needs to access the facility LAN which uses the 10.4.0.0/16 network.

Install a second NIC into the development workstation. If this is not possible, such as when using a laptop, a simple USB to Ethernet adapter will suffice (I have used a TrendNet TU2-ET100 with success).

The NIC should be configured as follows:

IP address	See To ensure the second NIC properly routes traffic it may be necessary to add static routing to the PC. In Windows 7, this is done from an elevated command prompt. <pre>route -p add 10.4.0.0 mask 255.255.255.0 10.4.0.200</pre> <pre>route -p add 192.168.0.0 mask 255.255.0.0 10.4.0.200</pre> Table 9 below
Subnet mask	255.255.255.0
Default gateway	10.4.0.200

To ensure the second NIC properly routes traffic it may be necessary to add static routing to the PC. In Windows 7, this is done from an elevated command prompt.

```
route -p add 10.4.0.0 mask 255.255.255.0 10.4.0.200
```

```
route -p add 192.168.0.0 mask 255.255.0.0 10.4.0.200
```

Table 9 PC IP Address Assignments

System	Address	Hostname
ATST GIS Development	10.4.0.90	GIS_Dev
ATST GIS Maintenance	10.4.0.91	GIS_Maint
IMT Telescope Mount Drive Assembly Development	10.4.0.92	Tel_Dev
IMT Coudé Rotator System Development	10.4.0.93	Rot_Dev
Optical Support System Development	10.4.0.94	OSS_Dev
Instruments System Development	10.4.0.95	Inst_Dev
Enclosure Development	10.4.0.96	Enc_Dev
Facility Equipment Development	10.4.0.97	Fac_Dev
Facility Thermal System Development	10.4.0.98	FTS_Dev

If additional addresses are needed they can be easily assigned, but they need to be coordinated with ATST to avoid duplication of addresses.