2024

PLC Programming Framework (Ladder Programming)

(N+P_PLC)V1.0

PHILLIP KRAGULAC

Contents

Licensing agreement

Introduction

The following document is a framework for programmers and machine asset owners. This is not an exhaustive list as to how machines are to be programmed but a over arching guide to align programmers' programming architectures with owners' expectations. There may be variations between this framework and equipment programming as a result due to specific equipment requirements.

This guide has been developed to provide a common standard for programming as well provide asset owners a better understanding of how their equipment works. It provides asset owners with a standard of programming which can be implemented across their machinery. By standardising machine programming across multiple machines – it should assist with reducing equipment downtime during breakdowns and reduce the cost for system upgrading.

IMPORTANT: This guide is not to be used in place of a certified safety system. This is only for the purpose of functional control of an industrial machine. Safety systems require accurate risk assessments and comprehensive control measures – which this guide does not cover or include. It is also important that this guide is not used for control systems that are exposed to hazardous environments.

Benefits of Open Sourcing

This document is based on the Open-Source licencing. This allows for greater transparency between all stake-holders, promotes greater cross compatibility, and allows asset owners to utilise various service providers without incurring additional costs associated with reverse engineering programming. Open Sourcing promotes community involvement and embraces collaboration between various groups which in turn contributes more positive outcomes.

Contributors

Definitions

Add On Instructions (AOIs)

A program (or function block) which is reused within the code multiple times.

Assembly

A set of processes (Systems) which when working together modifies a raw product and transforms that product into a different form.

Inputs / Outputs (IOs)

These are the inputs and outputs typically connected to the interfacing slots on the PLC. Although it is not just restricted to these and can include those inputs and outputs connected to remote racks and also those devices connected via a more advanced communication medium (such as ethernet / Profibus etc).

Original Equipment Manufacturer

This is referring to the equipment manufacturer that design and manufactures the component being discussed.

Programmable Logic Controllers (PLC)

This is referring to the computer that is designed to run a custom program (as discussed in this framework) for the industrial Assembly.

Product

A material which when supplied to the subject machine (Assembly) changes from one form to another form as intended by the original design of the machine.

System

A set of components (also known as Sub-Systems) which when working together influence the outcome of a product being processed through a piece of machinery (also known as the Assembly).

Sub System

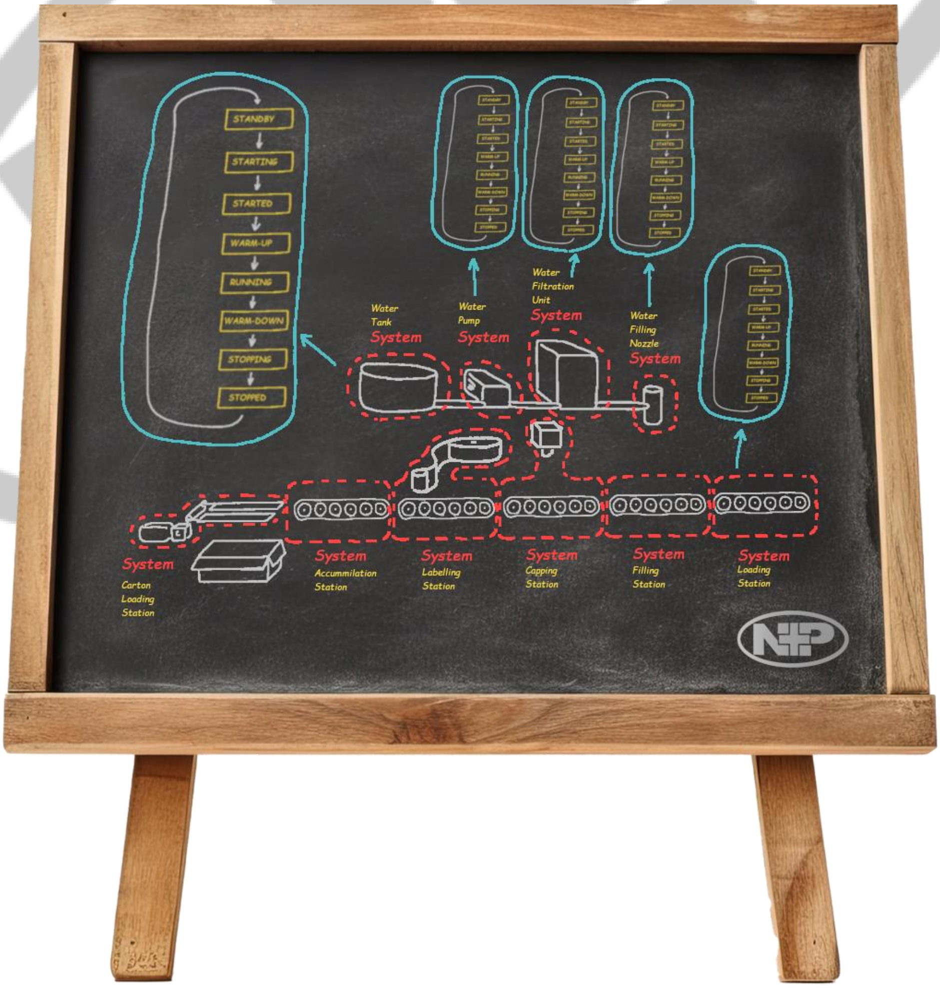
A physical component which either provides an input or an output to a control system.

Equipment Overview

Machines are present in almost every premise these days. Some examples of machinery include; Fabrication Equipment (such as saws, drill presses, sheet metal presses), Bottling Equipment, Spraying Systems, Welding Systems, Crushing Units, Mixing Systems, and so forth. All these systems typically have a control system installed. Some of these control systems are managed by basic electrical circuits while others are controlled by Programmable Logic Controllers (PLCs). Regardless of the type of control system used, they all employ the following principles. That is, the different levels of control with how it manipulates the materials (also known as products) in which it was designed for.

How It Works

This framework can be used across various types of systems. It's designed in a way that it can be scaled, it is flexible for future expansion and upgrading, assists with system integration, and should assist with breakdown recovery times. What this guide does not do, is teach how to program a PLC. It is assumed that a suitable level of programming knowledge is held by the programmer.



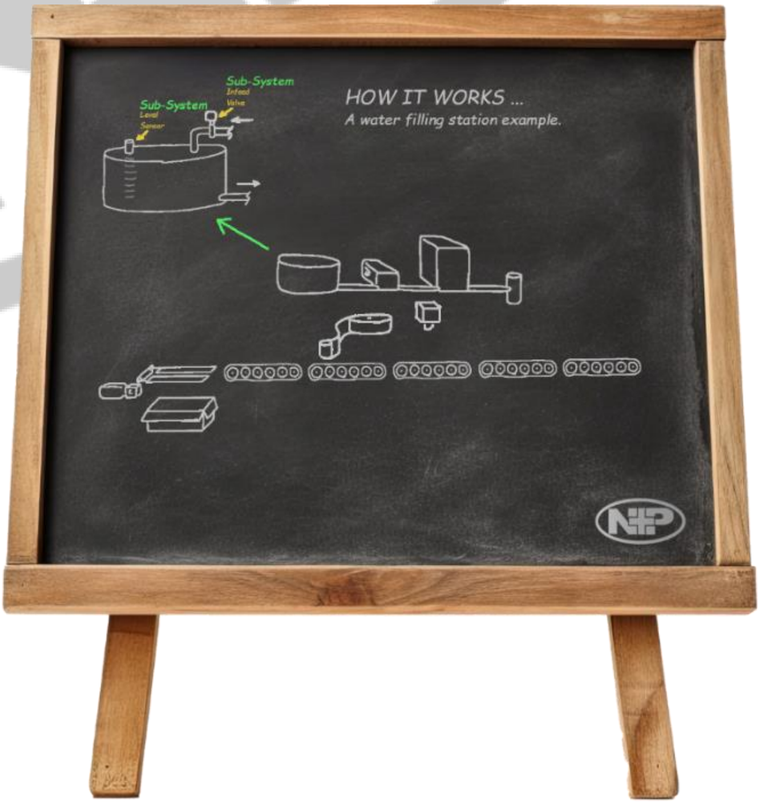
What are Sub Systems?

The best starting point in understanding how machinery works, is looking at the basic components. A machine is made up of various sensors, buttons, actuators and valves. Each of these components (for example a sensor), in the confines of this document, are classified as a Sub System. These units can be as basic as a dry contact switch or as complicated as a VSD having its own internal CPU.

These components can be connected to a more advanced communication medium such as ethernet, however employing such technologies will increase system costs as well as reduce 'fix' options during breakdowns. There are also risks associated with differences in communication protocol versions which can result in cost increases during breakdowns when equipment has become obsolete. When keeping communication to basic forms like digital IOs and analogue IOs, this risk can be significantly reduced.

Within this document we will use the example of a bottling assembly. An example of a Sub System within this scenario, is that of a product liquid level sensor (input) which is monitoring the tanks product liquid level which is ultimately going to be used for filling bottles on the production line. Another example is that of the valve (output) which controls the flow of the fluid being supplied to that said tank. Both components are independent from one another and there only link is through the control System that either listens to their signals or tells them when to operate.

An important point to take from this is that the communications between the component should be reduced to a small number of Input / Outputs (IOs). Ideally, the number of connections needs to be kept to a minimum between these units. The reason being that it reduces the workload required for finding non 'like for like' parts during breakdown and provide greater options for what can be later used if there is a desire to change the unit from the Original Equipment Manufacturer (OEM).



What are Systems?

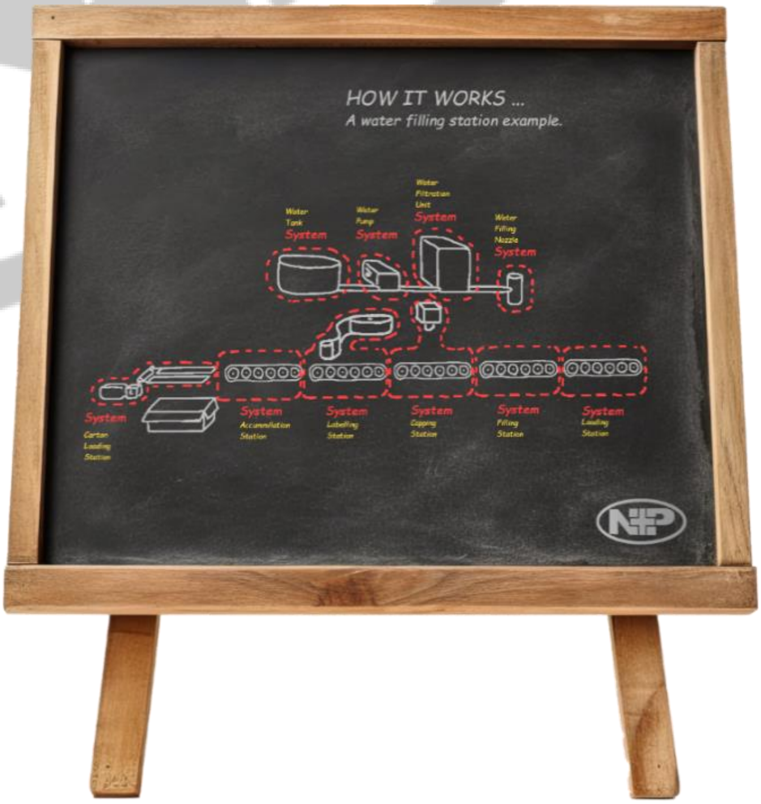
The next level up from Sub Systems are Systems. System is the term used to describe a relationship between various components (also known as Sub System). This is typically a single point of transformation of a product being used in the manufacturing process. It usually only involves a single step. It usually involves monitoring and controlling various Sub Systems.

An example of a System, relating to the bottling Assembly scenario, is that of the product liquid storage tank. The tank level needs to be constantly monitored and filled when nearing the empty point. The purpose of the tank is to act as a buffer between the slow product liquid supply and a high-volume discharge during the bottling filling phase. The connection of the liquid level sensor with the control valve that re-fills the tank, is what we refer to as a System. It is the system of filling the tank. Note that it does not include the draining, that would be another System.

Another example of a system could be the detection of a bottle being present within the filling location and the valve filling the liquid into the bottle. Note that the System is kept to a minimum number of interactions between Sub-Systems. This means that the System which is responsible for filling the bottles does not include moving the bottle in and out of the filling location.

Another example of a System could be the maintenance of pressure within the product liquid fill line. The Sub-Systems utilised within this System are the product liquid fill line pressure sensor and the VSD for the motor connected to the product liquid fill line's pump. The interaction between these sub-systems (i.e. when the pressure drops below a certain amount - the pump starting) is what is known as the System. The System is maintaining the set pressure of the product liquid fill line.

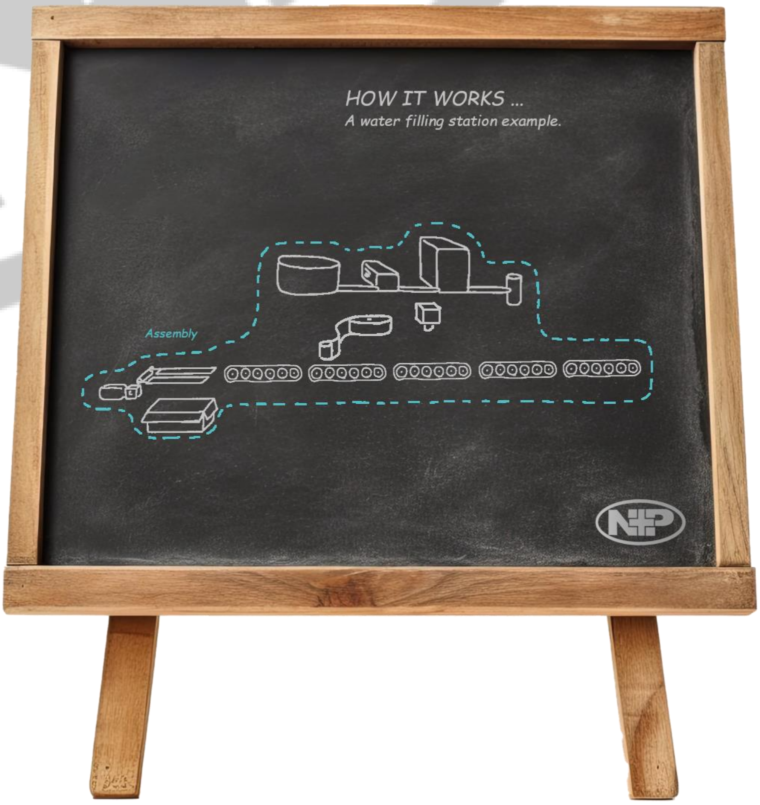
Ideally, it should be possible to list all the Systems utilised within a PLC as well as having each System residing in its own independent program module. This allows for faster fault finding as well as faster and more cost-effective upgrades.


HOW IT WORKS ...
A water filling station example.
Water Tank System
Water Pump System
Water Filtration Unit System
Water Filling Nozzle System
System Carton Loading Station
System Accumulation Station
System Labelling Station
System Capping Station
System Filling Station
System Loading Station

What are Assemblies?

Finally, Assemblies are the collection of Systems which ultimately transfer a product from its raw form to a more finished product.

An example of an Assembly is the filling, labelling, and capping of water bottles. The Assembly has various Systems (tank storage level management, bottle filling, bottle capping, conveyor movement, bottle labelling etc).

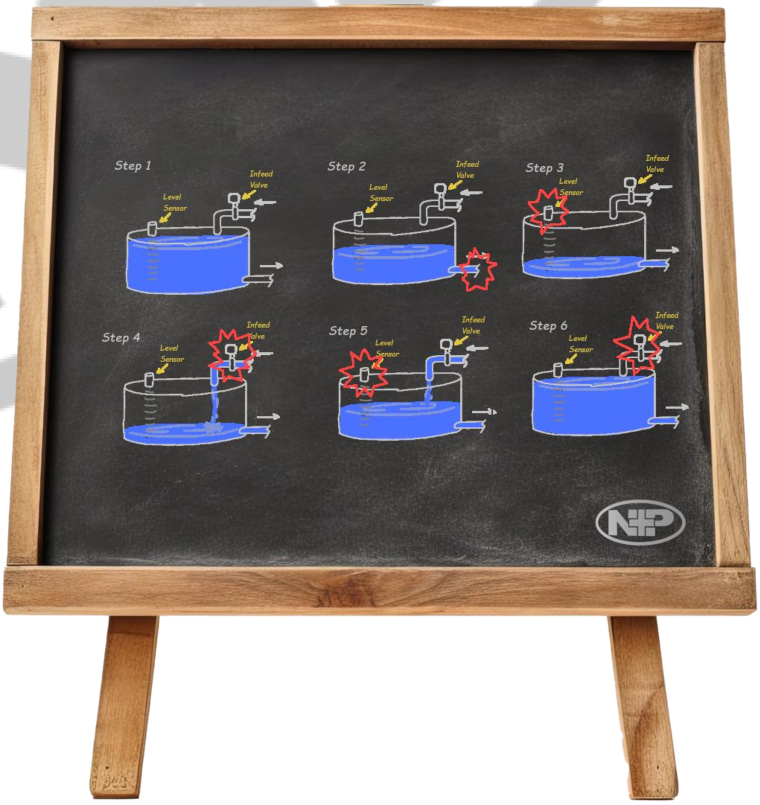


What are Sequences?

Sequences are essentially a storyline or pathway of how a machine is to operate. They are utilised in both System level programming as well as Assembly level programming[1]. It includes both the management of the outputs (controlling valves and VSDs) as well as monitoring of inputs (buttons and sensor readings). It is the logical progression of events during a machine's operation.

An example of a sequence is the product tank liquid level monitoring System. Initially, when the tank is full, the System (filling the tank) is in the **Standby Stage**. Once the water level drops below a set low level point, the Sequence then goes into a **Filling (Run) Stage.** It then subsequently triggers the inlet valve to open. Once the product liquid level climbs to the high point, the Sequence goes into a **Stop Stage** which then triggers the inlet valve to close. Once the Sequence detects that the valve has closed, and the level is no longer increasing then the Sequence switches to the **Standby Stage** again. This process is then repeated as often as required.

Sequence programming is typically included within the System and Assembly programming modules.



[1] Not to be confused with the low-level programming that is transferred to the PLC. For the purposes of this framework Assembly Programming is that which involves the interconnection programming that links the system programming.

What Are Stages?

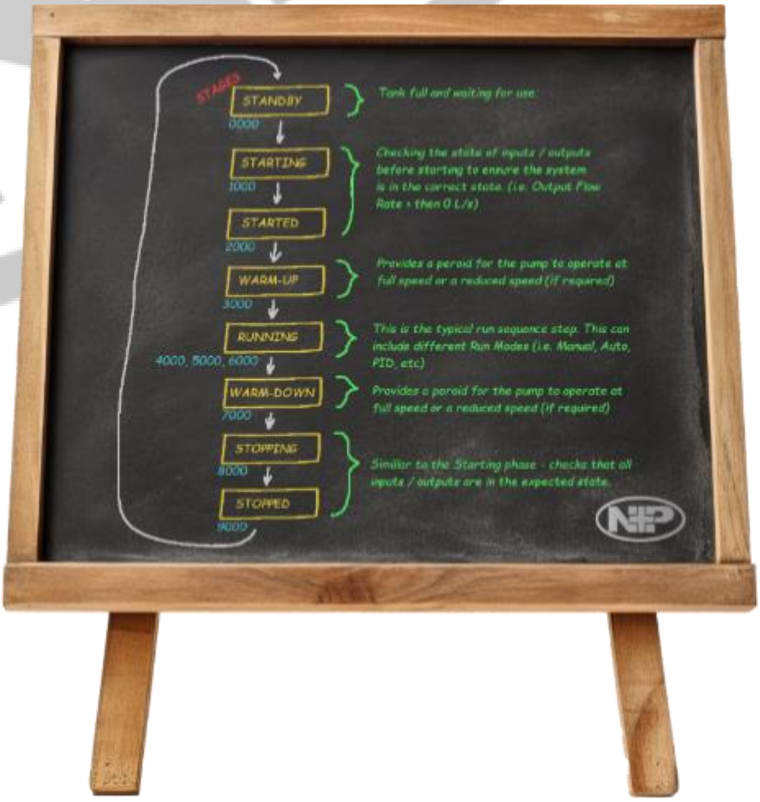
Stages are the sub-set of a Sequence. They are the steps in which a machine takes as opposed to the entire storyline (which would be referring to the Sequence). Depending on what conditions are present at the time, the machine will jump to the required Stage as pre-determined within the Sequence programming. For example; if an operator presses the Run button while the machine is in the Standby stage, the machine will progress to the Starting stage.

As part of this framework there are 10 fundamental stages;

1. Standby
2. Starting
3. Started
4. Warm-up
5. Running (Initial)
6. Running (Manual)
7. Running (Auto)
8. Warm-Down
9. Stopping
10. Stopped

Depending on the system requirements, there may be more present, however as a fundamental all of these stages should exist in every System. This is to ensure that in the chance that the System needs to be upgraded, and or improved, these items are not having to be added and integrated at a later stage into a working system. Making major modifications into a working system can increase the risk of equipment damage and malfunction during programming.

This framework allows for greater flexibility when adding stages (which is a common request during the life cycle of equipment). If implemented across multiple machines it allows for greater understanding by programmers and technicians during breakdown. It also allows for greater visibility of System status across all levels within the machine. It also lowers the risk of incorrect programming. This method of programming also allows for safer cut-overs when programming on live equipment.


STAGES
STANDBY
0000
Tank full and waiting for use.
STARTING
1000
STARTED
2000
Checking the state of inputs / outputs before starting to ensure the system is in the correct state. (i.e. Output Flow Rate > then 0 L/s)
WARM-UP
3000
Provides a peroid for the pump to operate at full speed or a reduced speed (if required)
RUNNING
4000, 5000, 6000
This is the typical run sequence step. This can include different Run Modes (i.e. Manual, Auto, PID, etc)
WARM-DOWN
7000
Provides a peroid for the pump to operate at full speed or a reduced speed (if required)
STOPPING
8000
STOPPED
9000
Similar to the Starting phase - checks that all inputs / outputs are in the expected state.
NHP

Essentially Sequencing operates on an indexing variable. The corresponding values for each of the stages are detailed below.

*Standby (Index: **0000** → 0999)*

Within this Stage the System is not operating and is waiting to be initiated. This can either be manually triggered via operator intervention or via an AUTO Run Switch / Latch. Once a Sequence has finished its Stopped Stage, it should return back to this Stage. During this Stage, no output should be operating.

*Starting (Index: **1000** → 1999)*

This Stage typically involves the System conducting pre-start checks before running. These can include making sure consumable levels are correct, and or doors are closed etc. This is different from fault interrupters which occur during the system Run Stages. These checks are usually those System readings which are present while the system is operating.

An example of this would be the System checking the pressure on the product liquid fill line. As the system would not be running, the expected pressure reading should be near zero. If the System detects a high-pressure reading, then the System could conclude that there is a fault which maintenance crews need to investigate.

This Stage can also include other functions like opening supply lines to the System.

*Started (Index: **2000** → 2999)*

This Stage is essentially the point in which the above checks have passed, and the System can then progress to the next Stage. This Stage can also include checks that the actions taken in the previous stage have successfully executed (for example checking those pressures in the supply line (downstream of the valves are within an expected range).

*Warm-up (Index: **3000** → 3999)*

This Stage is typically used for warming equipment up (if required). It can be used for charging lines as well as for testing liquid lines at lower pressures prior to engaging full pressure (which would be present during normal operation). Thereby reducing the amount of spillage in the event of pipe rupture. Having the system start at a lower pressure may also reduce the effects of hydraulic shock. When using this stage, a timer will be needed, which switches over to the Running stage (4000) after a pre-determine amount of time. If this Stage is not required, then program the system to immediately jump to the next Stage (4000).

*Running [Initial] (Index: **4000** → 4999)*

This Stage is the initial Stage of Running. Within this Stage the System typically determines which Running mode is required. It can do this through checking user inputs and or equipment states. An example of this is if the user has selected AUTO on the mode switch on the control board. As part of this Stage, the System would check the state of that switch and then change the Stage to Running (Auto) (6000).

Typically, when a Stage is switched out (for example being switch out of AUTO to Manual mode by and operator), the Sequence will then be switched back to this Stage (4000) which will then determine that the next stage will be Run Manual mode (5000).

*Running [Manual] (Index: **5000** → 5999)*

In this Stage the System / Assembly is being manually ran. System setpoints are typically defined by the operator / maintenance staff. When the System / Assembly detects a change in Run Mode (for example Manual → Auto) then the Stage will be changed to Running (Initial) (4000).

*Running [Auto] (Index: **6000** → 6999)*

This Stage is the typical stage that's going to be employed most of the time. For complex Systems which may have several steps, additional Stages can be added (for example 6001, 6002, 6003 and so forth).

An example of this could be a product liquid fill line pressurisation system which operates at two pressures (Standby and Fill). When the System isn't filling bottles with liquid, the Stage is set to Running Auto Standby (6000). When the System detects that it is about to fill a bottle (via monitoring another System's status which is responsible for filling product into the bottles) then it sets its own Sequence Stage to Running Auto Fill (6001). When the System responsible for Bottle Filling changes to completed, the Pressure System sets its own Sequence back to the Running Auto Standby (6000) stage.

*Warm-Down (Index: **7000** → 7999)*

Please refer to Stage Warm-up (3000) as this is the reverse function of that Stage.

*Stopping (Index: **8000** → 8999)*

Please refer to Stage Starting (1000) as this is the reverse function of that Stage.

*Stopped (Index: **9000** → 9499)*

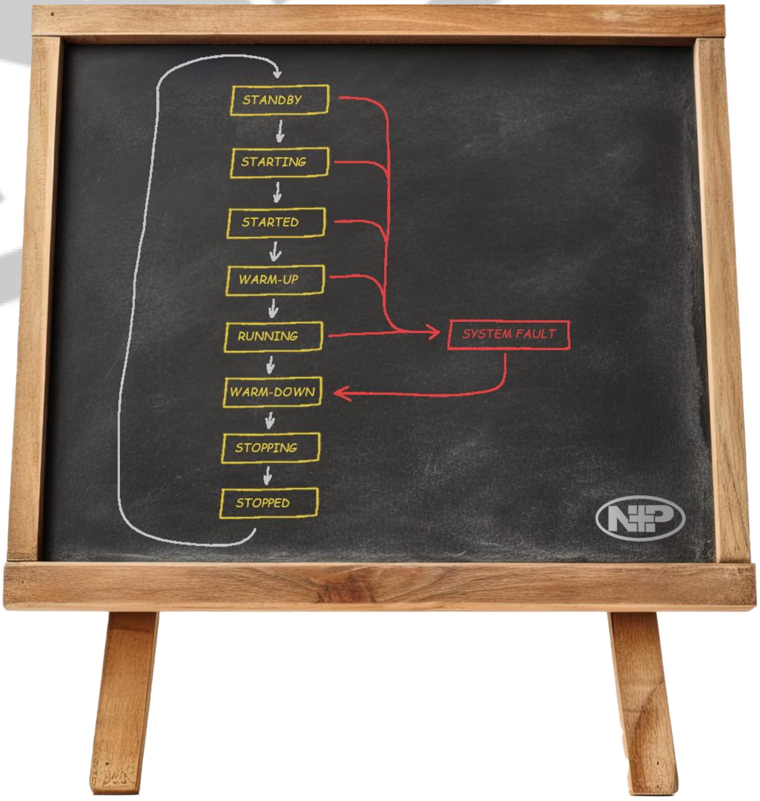
Please refer to Stage Started (2000) as this is the reverse function of that Stage.

*Operator Intention (Index: **9500** → 9899)*

…

*System Fault (Index: **9900** → 9998)*

This Stage is for setting the system into a state which is acceptable under any fault conditions. The purpose for employing a Fault Stage is to ensure that when a fault does occur, the System is then deliberately interrupted, and requires a deliberate action by the operator and or maintenance to acknowledge the fact that an error has occurred.

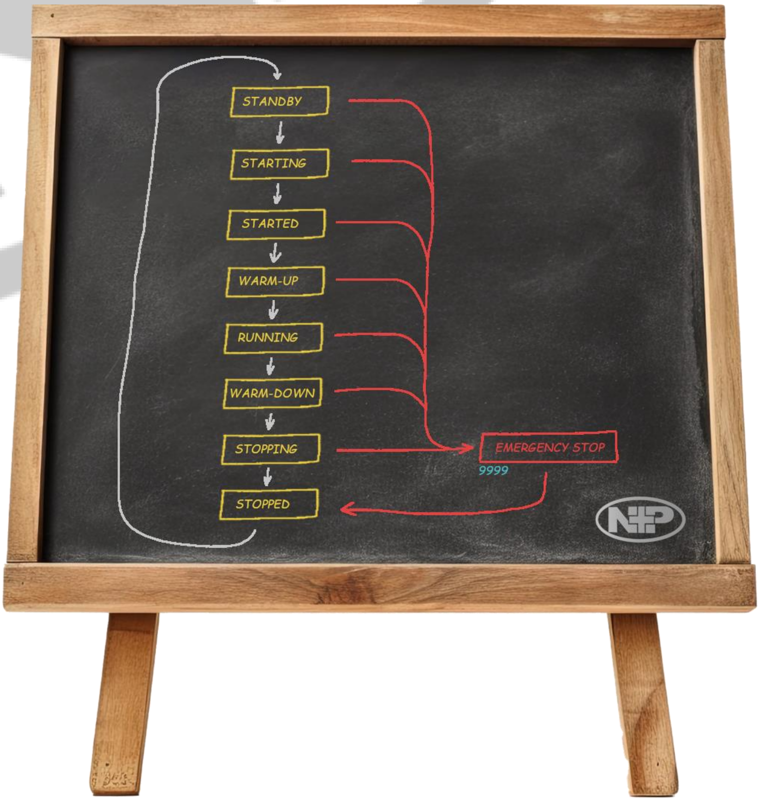


*Emergency Stop (Index: **9999**)*

IMPORTANT: This Stage is not a replacement for having a certified safety system. This is a supplementary and informative function to allow the system to return to a state ready for resetting.

The emergency stop Stage is purely for stopping the control system outputs and ensuring that they ready for resetting. This stopping function is purely supplementary to the emergency control system and is only there if the safety system fails, the control system isn't forcing the outputs to trigger. It is also, used for setting System / Assembly into a state ready for it to be return into service.

In the same manor that the Fault Stage works, the system requires acknowledgement from the operator and or maintenance staff that the system has been placed into emergency mode. Within this section, any emergency indications which are added, must not be latching in nature.



Programming Layouts

Modification Guide

It is recommended, that any modifications made to any of the programming should be clearly recorded and commented within the project. This allows for programmers to quickly revert back to known working code as well as contact any programmers regarding any questions they may have regarding the code that has been added. It also allows asset owners to contact the Service Providers if any future changes are required.

Commenting

Rung Changes

```
[YYYYMMDD] [Initial].[Last Name] [Company] [Description of Change]
```

Inputs Section

The following section is for mapping inputs into working variables. This assists with system upgrades, by reducing the number of points which need to be modified when a PLC is changed. It also helps when upgrading a system by allowing the programmer to easily switch the output from existing code to the new code. If inputs are not programmed in a way which allows for greater flexibility, it can present serious issues for programmers conducting live changes to existing and running systems.

It is also recommended that the inputs which are directly associated with that programming (module), be retained within the same module, unless it is used in others. At which point the input should be programmed into a common mapping module. The reason behind this is, if a System is going to be worked on, all the associated inputs are within the same logical area. This helps with any work being conducted post commissioning by programmers, who are unfamiliar with the system.

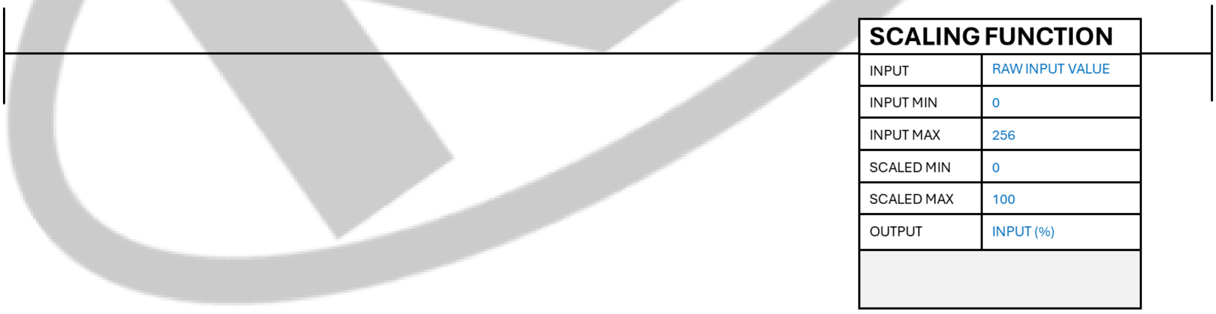
Programming

==== INPUTS ====

SECTION DESCRIPTION: The following section is for programming all inputs specific for the control of this program module. Any inputs which are used in multiple modules will be listed within the common input mapping module.

SCALING INPUTS

RUNG DESCRIPTION: The following rung is for mapping RAW Analog inputs into SCALED Working variables.

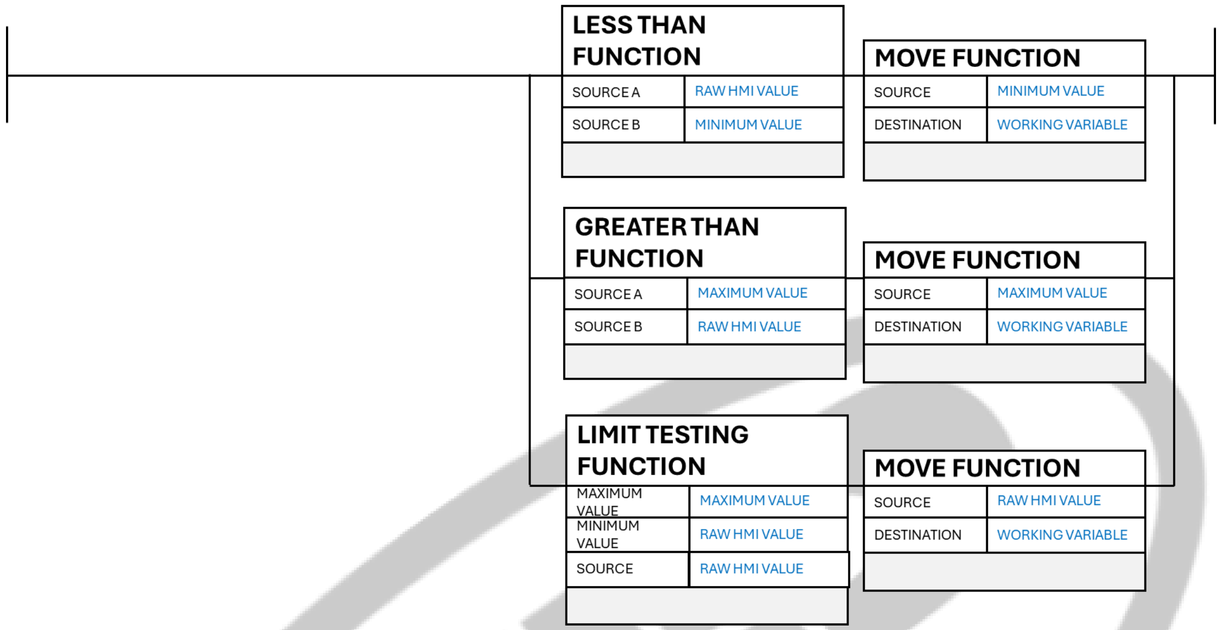


SCALING FUNCTION

INPUT	RAW INPUT VALUE
INPUT MIN	0
INPUT MAX	256
SCALED MIN	0
SCALED MAX	100
OUTPUT	INPUT (%)

HMI INPUT FILTERING

RUNG DESCRIPTION: The following rung is for mapping HMI user defined inputs into FILTERED Working variables.



LESS THAN FUNCTION

SOURCE A	RAW HMI VALUE
SOURCE B	MINIMUM VALUE

MOVE FUNCTION

SOURCE	MINIMUM VALUE
DESTINATION	WORKING VARIABLE

GREATER THAN FUNCTION

SOURCE A	MAXIMUM VALUE
SOURCE B	RAW HMI VALUE

MOVE FUNCTION

SOURCE	MAXIMUM VALUE
DESTINATION	WORKING VARIABLE

LIMIT TESTING FUNCTION

MAXIMUM VALUE	MAXIMUM VALUE
MINIMUM VALUE	RAW HMI VALUE
SOURCE	RAW HMI VALUE

MOVE FUNCTION

SOURCE	RAW HMI VALUE
DESTINATION	WORKING VARIABLE

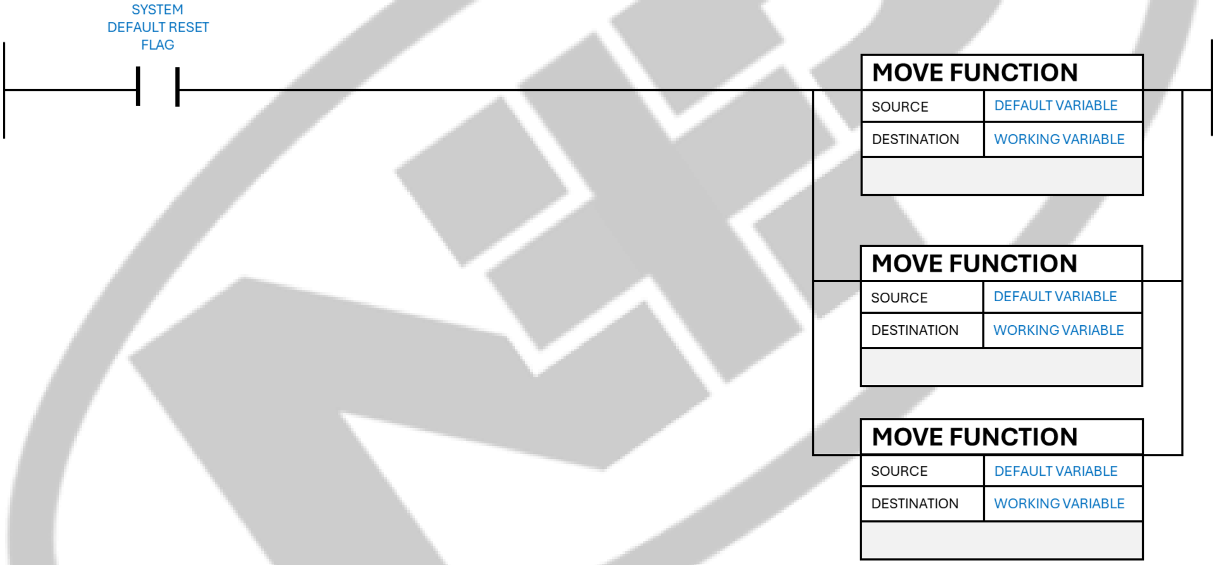
Default Parameter Restoration Section

The following section is for listing all critical default user defined variables. The purpose behind this code is to provide a working benchmark for programmers, if there are any doubts regarding the suitability of settings. Typically, once the System / Assembly has been commissioned, the settings, which have been proven to work, are loaded into this section. It is also possible to have the ‘System Default Reset Flag’ linked to an HMI which allows operators to switch back to a known working state, allowing them to self-recover if they have made an error entering a wrong value into the HMI.

Programming

==== DEFAULT SETTINGS ====

SECTION DESCRIPTION: The following section is for resetting user defined variables back to known working values. It is highly recommended that these values are updated at the completion of final commissioning.



General Operation Section

This section is primarily for any function which exists outside of the sequence. In other words, any function which is required to operate regardless of the Sequencing Stage.

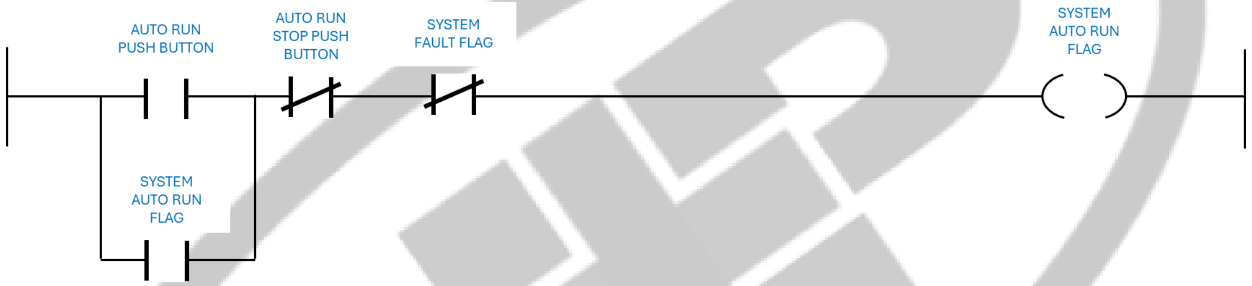
Programming

==== GENERAL OPERATIONS ====

SECTION DESCRIPTION: The following section is for any operation which is required to be process regardless of the Sequence Stage.

Auto Run Function

RUNG DESCRIPTION: The following rung is for providing a latching Run Auto function.



Fault Section

Like the 'General Operation Section' this section is dedicated for all fault monitoring. This section of programming will be processed on each cycle. Any conditions, which should result in a positively switched variable, is then picked up in the Sequence Stages.

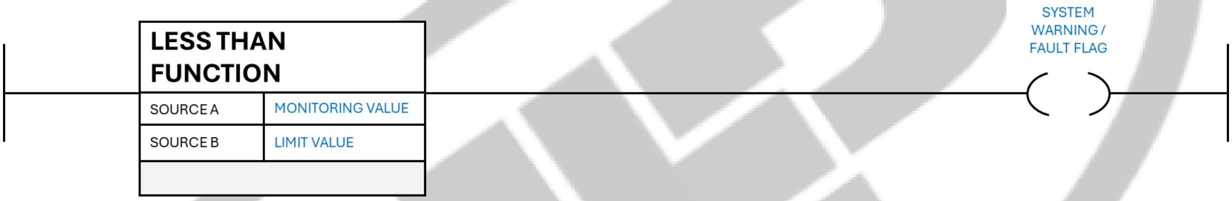
Programming

==== FAULT MANAGEMENT ====

SECTION DESCRIPTION: The following section is for monitoring fault condition. Any faults detected are linked to a flag variable which is then programmed into the Sequence Stage.

Low Level Monitoring (with Nuisance Protection)

RUNG DESCRIPTION: The following rung is for detecting a low value reading.



Range Monitoring (with Nuisance Protection)

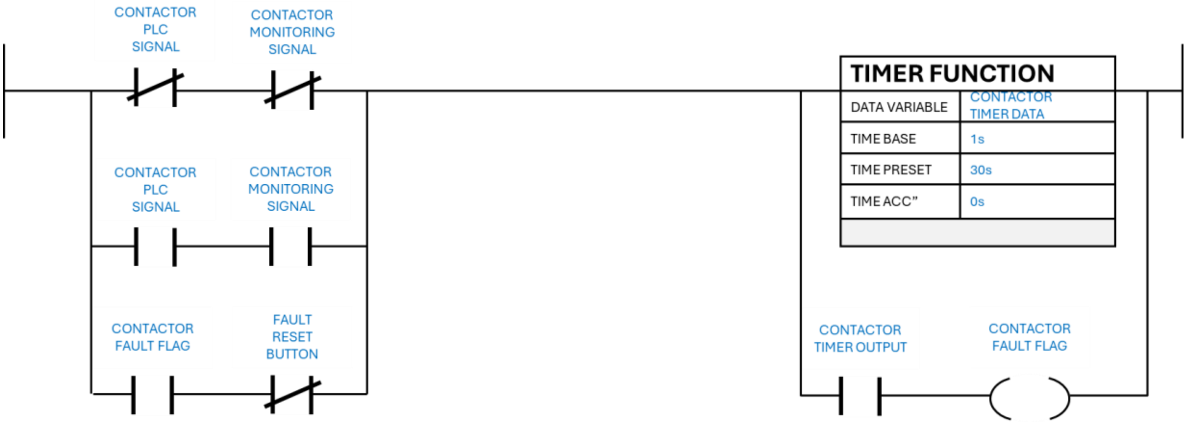
To be added…

High Level Monitoring (with Nuisance Protection)

To be added…

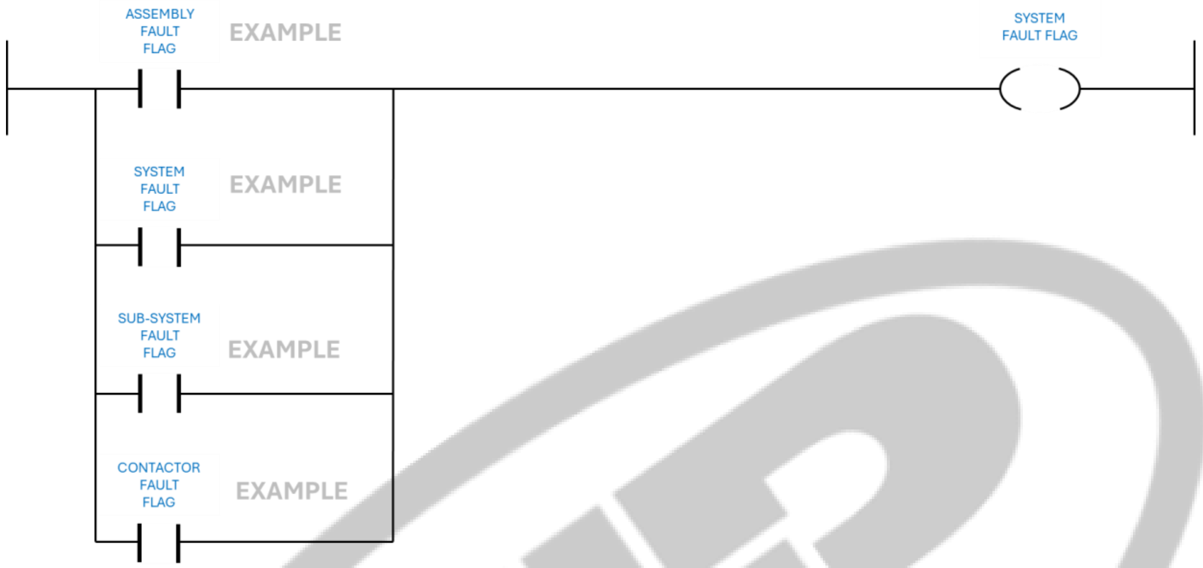
Contactor Monitoring (with Nuisance Protection)

RUNG DESCRIPTION: The following rung is for detecting the condition of contactor. The Timer function is included as a debounce filter.



Fault Signalling

RUNG DESCRIPTION: The following rung is for grouping multiple flags to a single common flag. Typically used for HMI displays.



PID Section

To Be Added…

Sequencing Section

As discussed previously, the sequencing section is a step-by-step process. As the system progresses through its operational phases, the PLC jumps from one Sequence Stage to another. This allows for greater flexibility with future modifications and allows for greater control when working on a live system.

Programming

For the following section the following shading represents:

🟦 **Stage Support Programming**
Any programming which is not always used and is only included to support the functionality of something within that stage. For example having a Timer which then automatically triggers a jump to the next stage (possibly in a Warm-up / Warm-down function)

🟨 **Stage Transition Programming**
Any programming which is used for transitioning to another stage within that sequence.

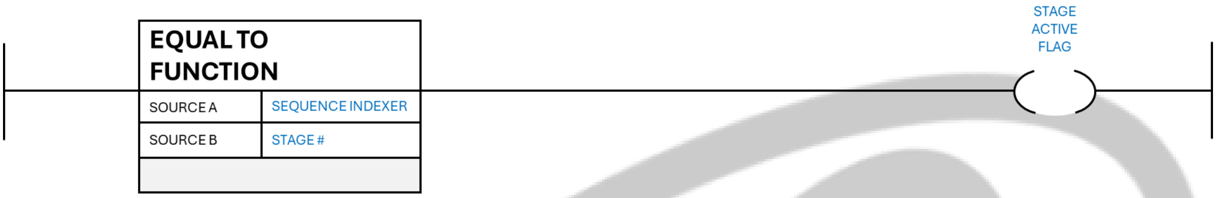
🟩 **Stage Control Programming**
Any programming which is used to control the output. Typically, there will be a MASTER variable which is linked to the output. The alternative settings (i.e. manual settings, auto settings, PID settings) which be switched in and out of operation (linked to the MASTER) when required.

STANDBY (Index: 0000)

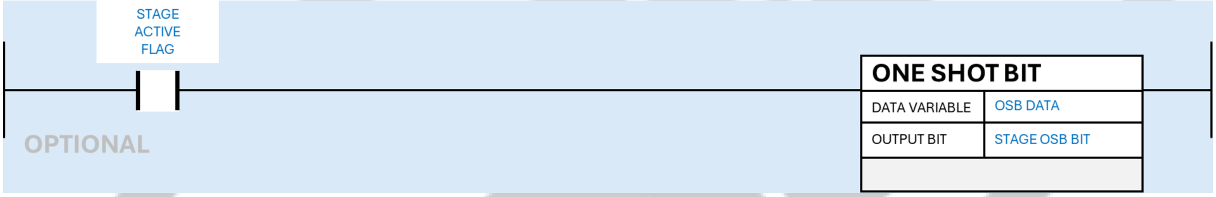
==== SEQUENCING [STANDBY] ====

SECTION DESCRIPTION: The following section for having the System placed in an ‘idle’ type state. Typically, there should be Outputs engaged during this state.

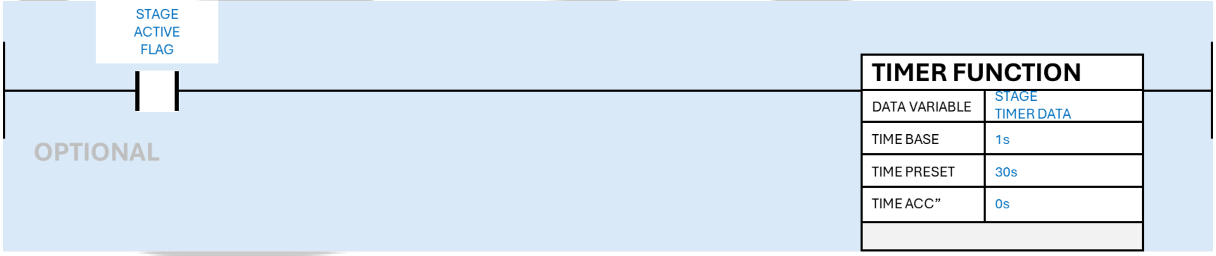
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



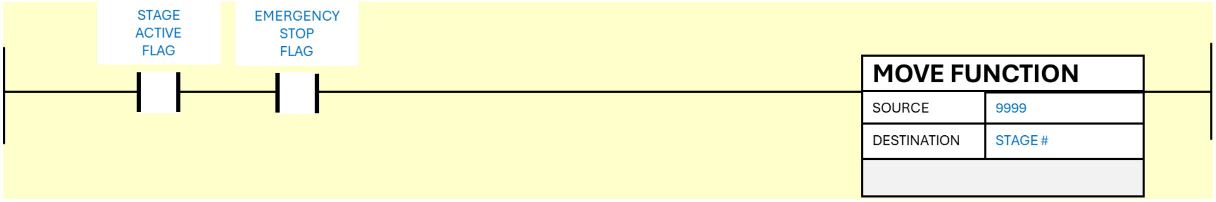
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



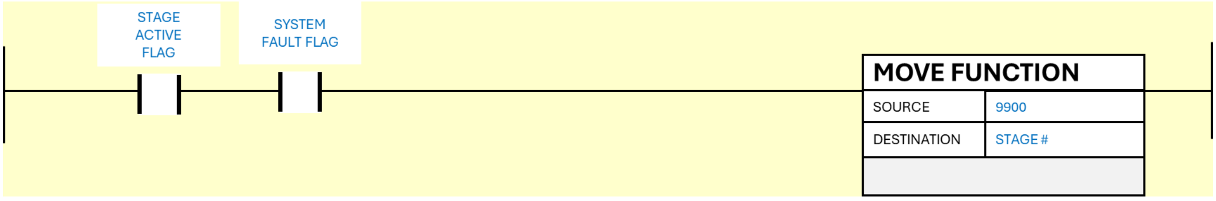
RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.



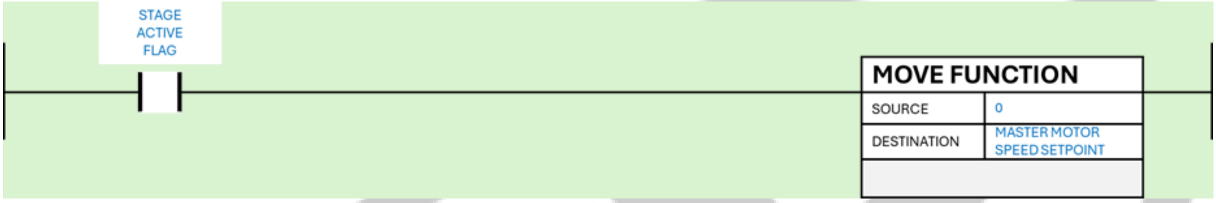
RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



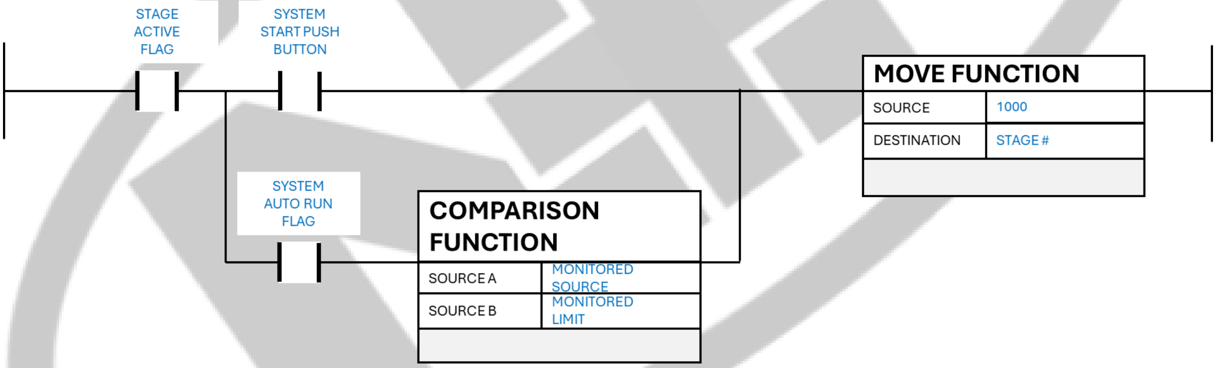
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



RUNG DESCRIPTION: The following rung is for setting the speed and or state of an output.



RUNG DESCRIPTION: The following rung is for starting the System Sequence.

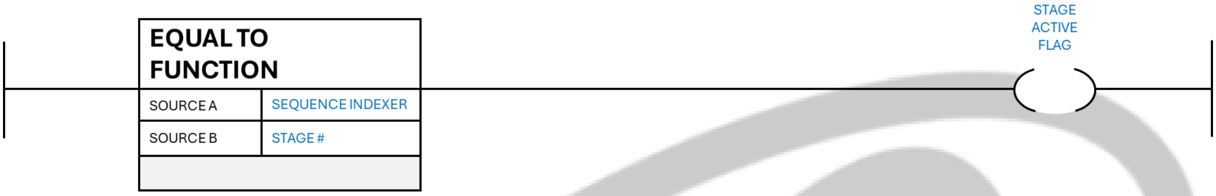


STARTING (Index: 1000)

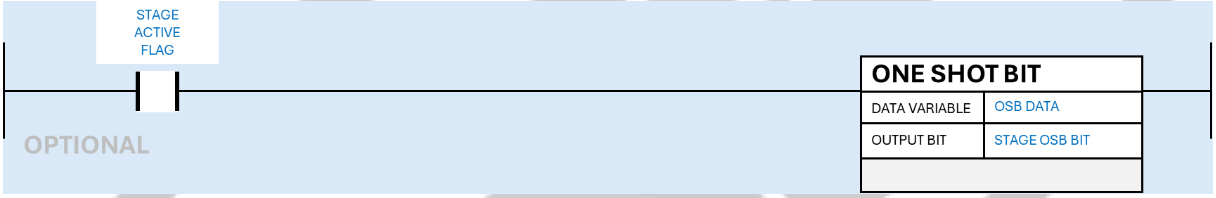
==== SEQUENCING [STARTING] ====

SECTION DESCRIPTION: The following section is typically for completing System / Assembly Pre-Start Checks and or energizing supplies.

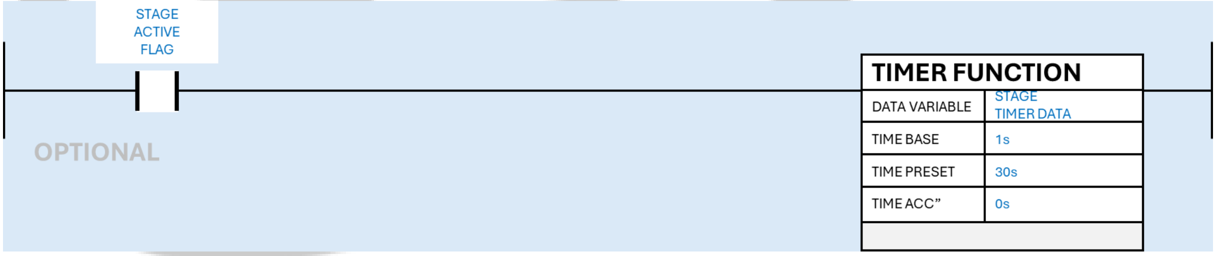
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.

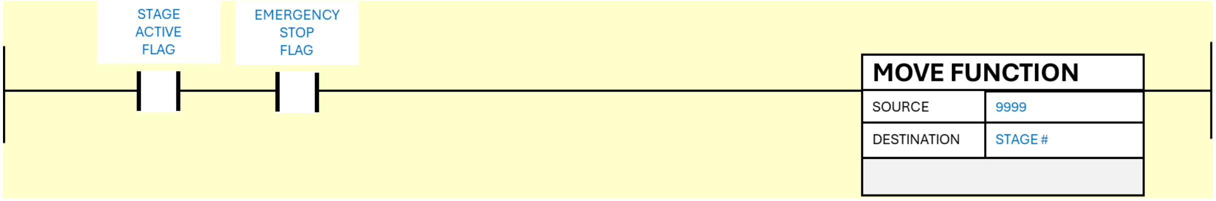


RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.

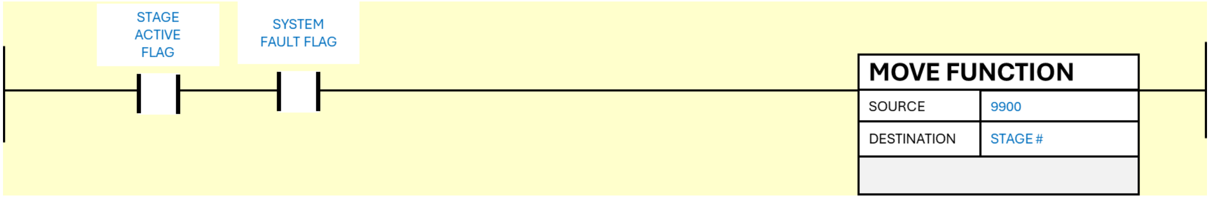


RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.

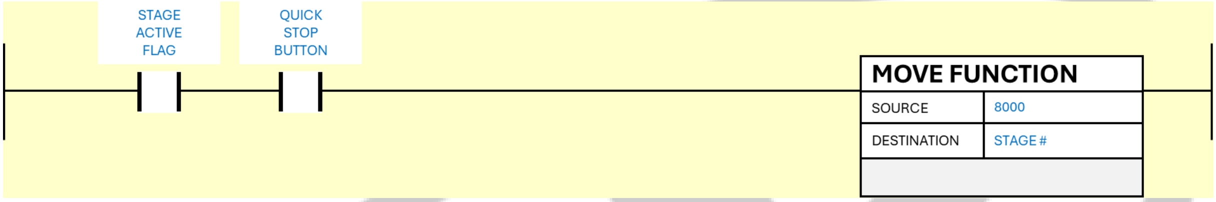
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



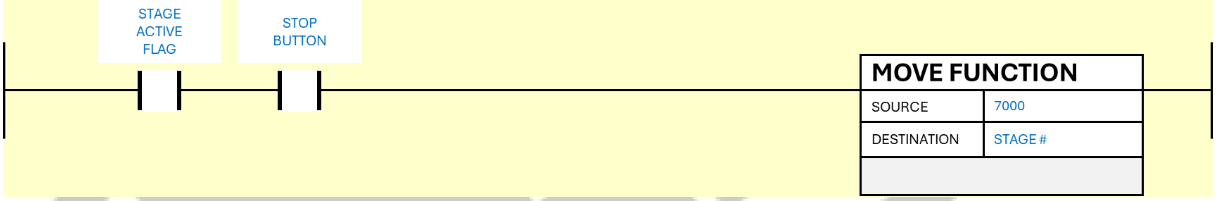
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



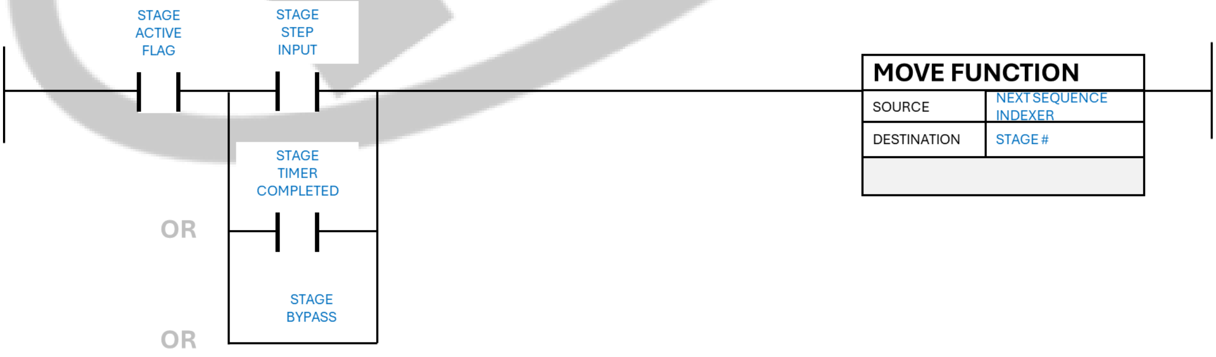
RUNG DESCRIPTION: The following rung is for progressing to the Quick Stop Stage.



RUNG DESCRIPTION: The following rung is for progressing to the Normal Stop Stage.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

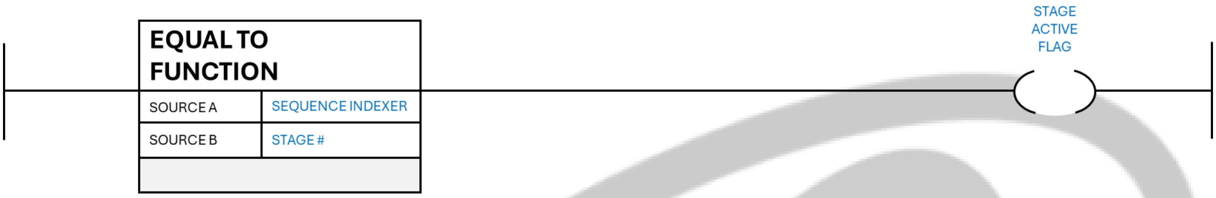


STARTED (Index: 2000)

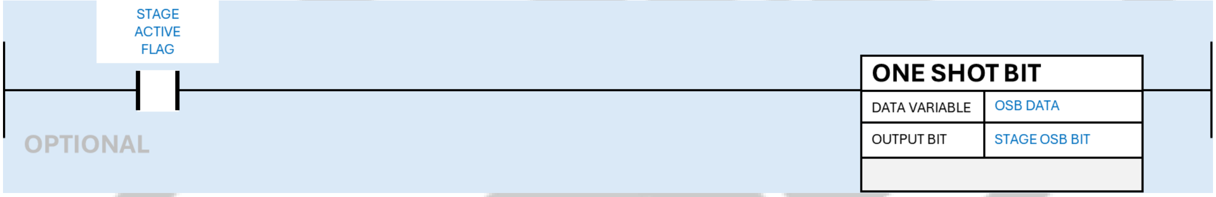
==== SEQUENCING [STARTED] ====

SECTION DESCRIPTION: The following section is typically for completing System / Assembly Post-Start Checks.

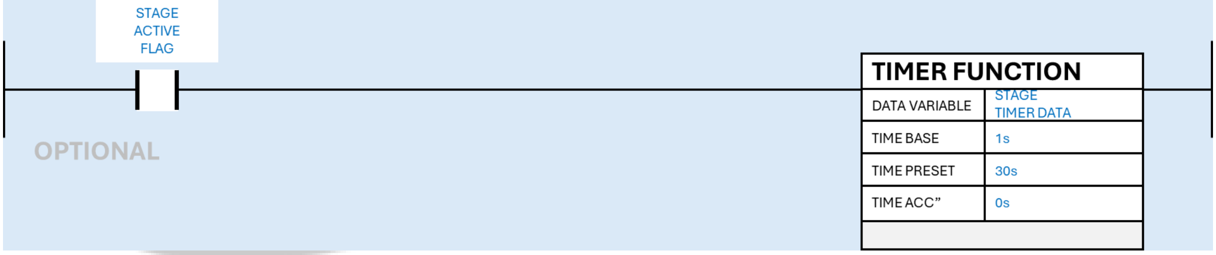
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



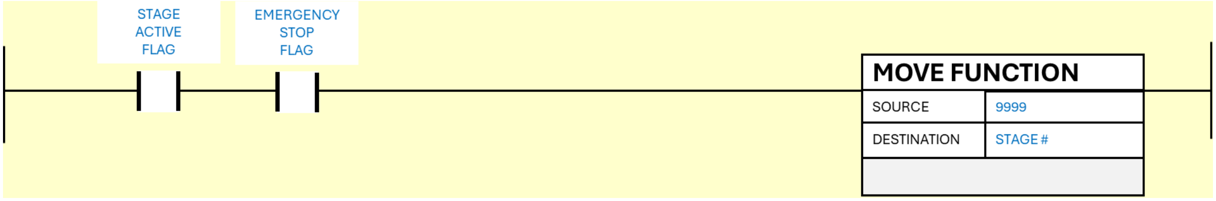
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



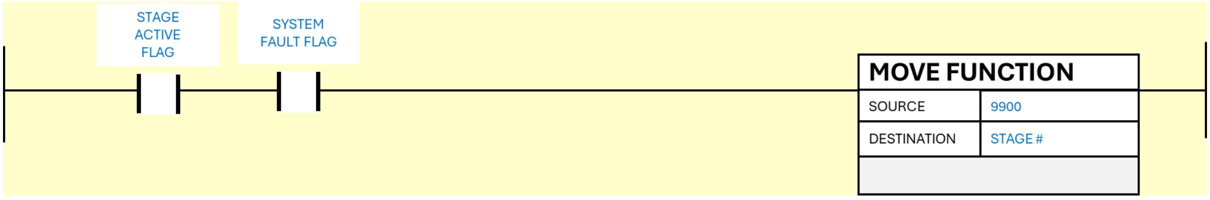
RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.



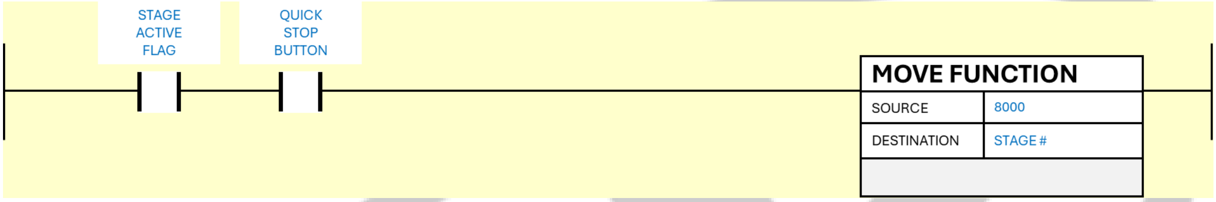
RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



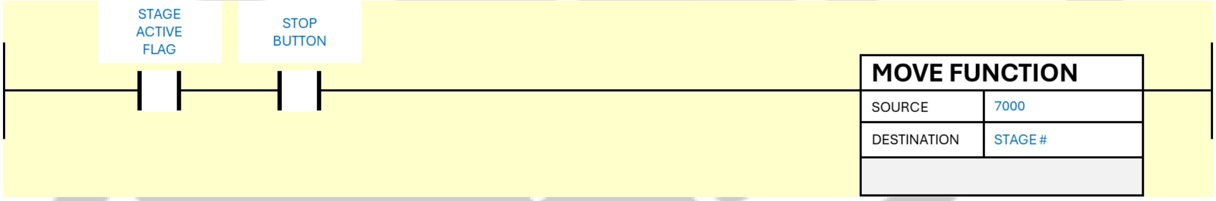
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



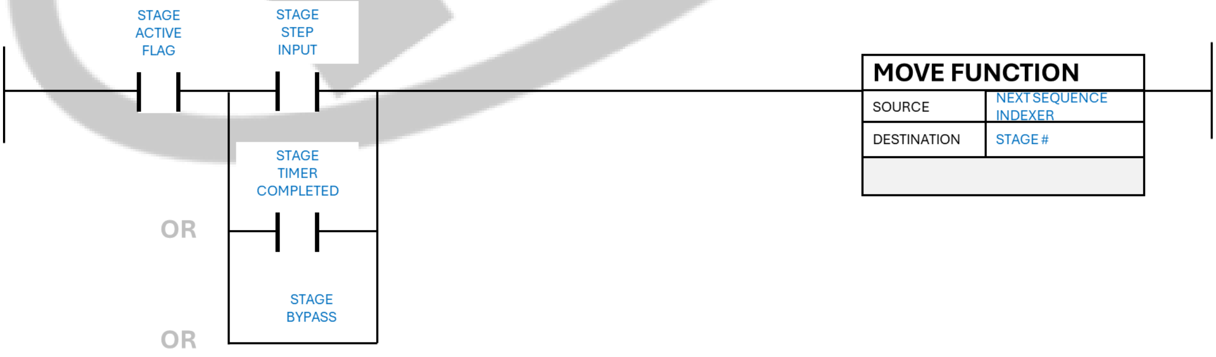
RUNG DESCRIPTION: The following rung is for progressing to the Quick Stop Stage.



RUNG DESCRIPTION: The following rung is for progressing to the Normal Stop Stage.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

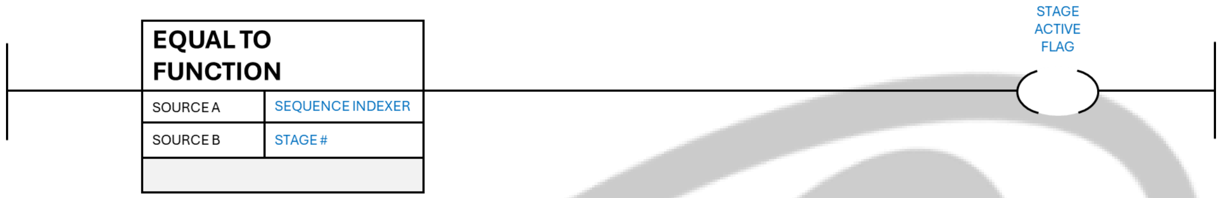


WARM-UP (Index: 3000)

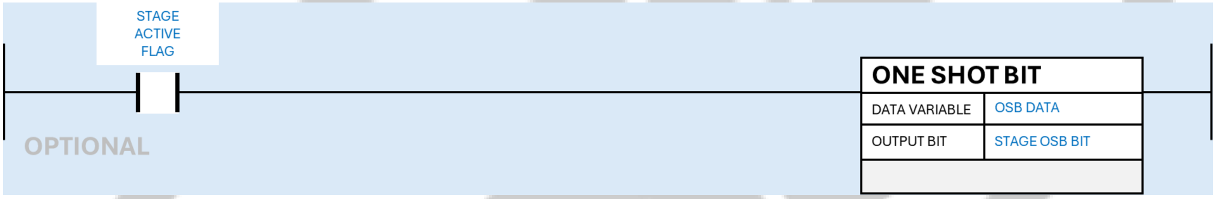
==== SEQUENCING [WARM-UP] ====

SECTION DESCRIPTION: The following section is for Warming-Up Equipment and or Charging Fluid Lines.

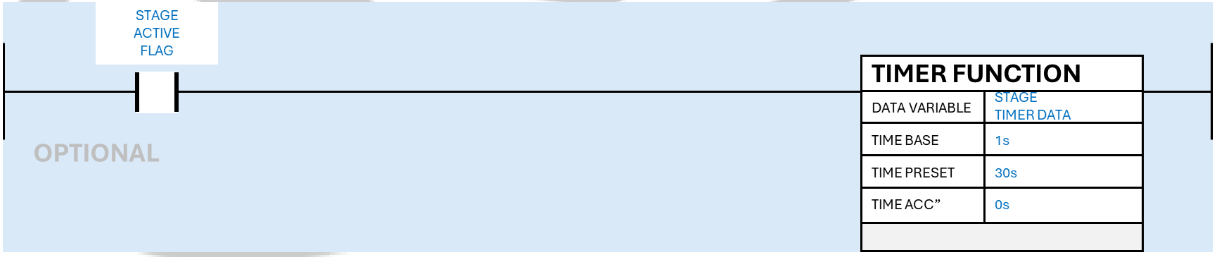
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.

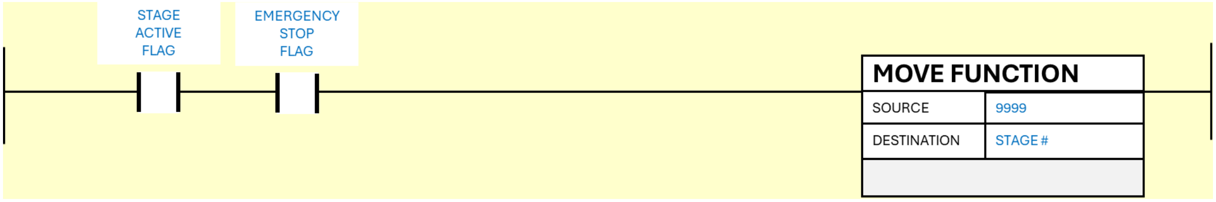


RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.

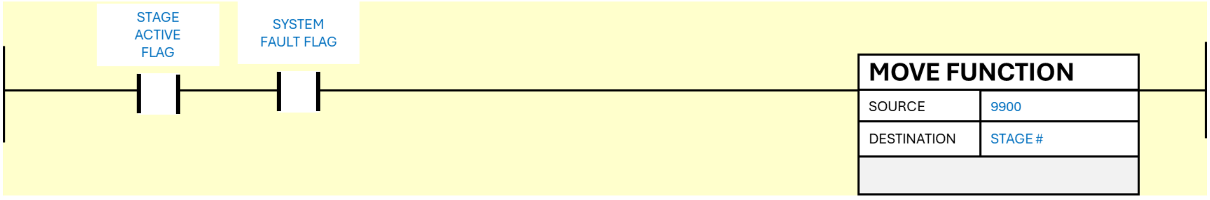


RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.

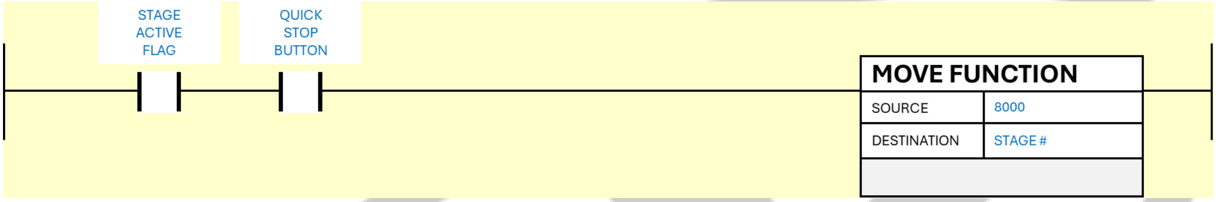
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



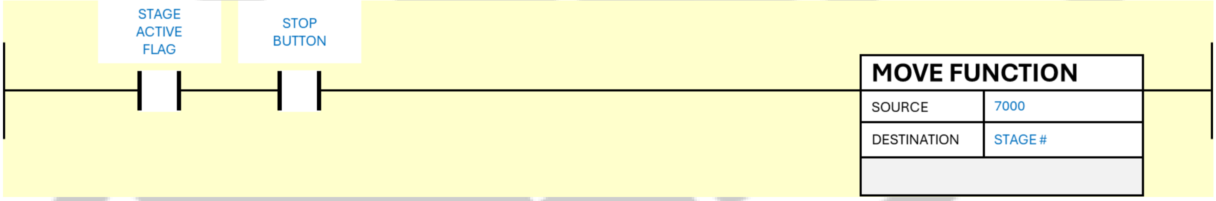
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



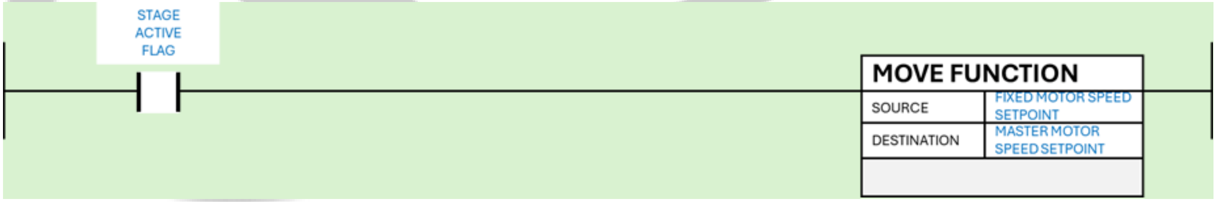
RUNG DESCRIPTION: The following rung is for progressing to the Quick Stop Stage.



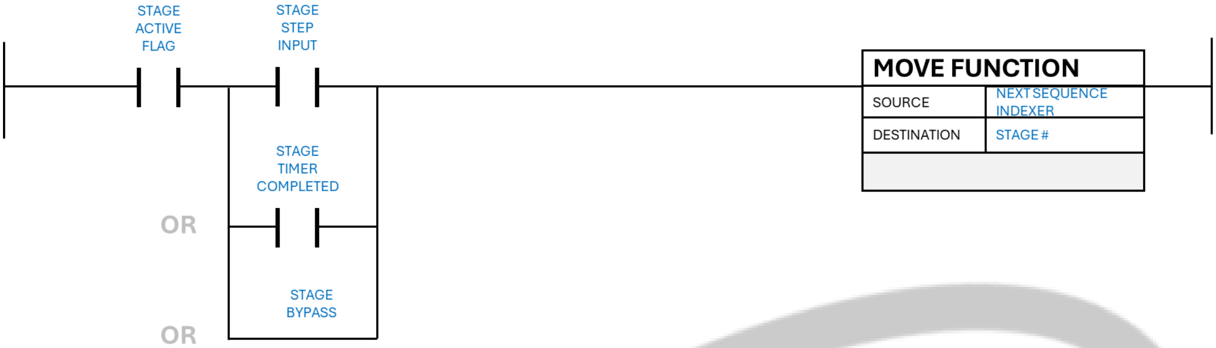
RUNG DESCRIPTION: The following rung is for progressing to the Normal Stop Stage.



RUNG DESCRIPTION: The following rung is for setting the speed and or state of an output.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

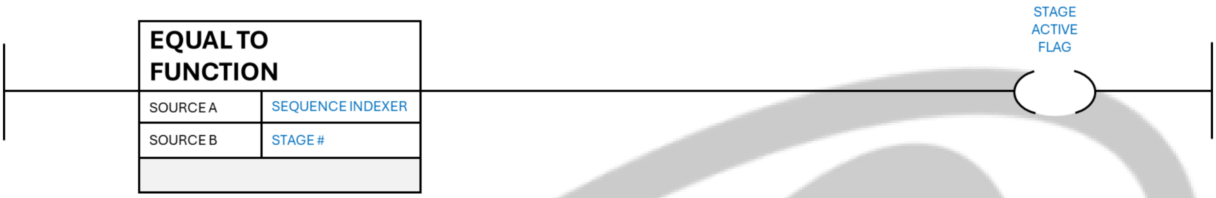


RUNNING ▶ INITIAL (Index: 4000)

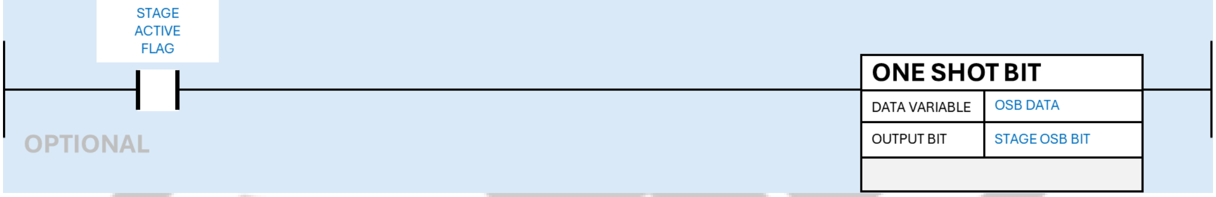
==== SEQUENCING [RUNNING - INITIAL] ====

SECTION DESCRIPTION: The following section is for checking which Mode the Sequence needs to be switch to. This typically falls within Manual and Auto (although not limited).

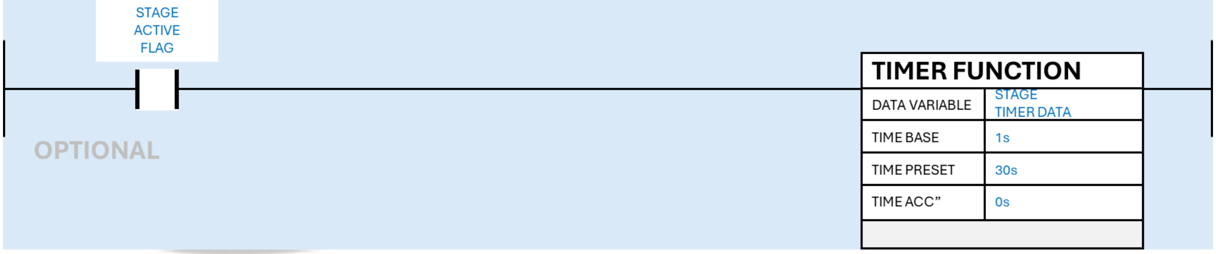
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



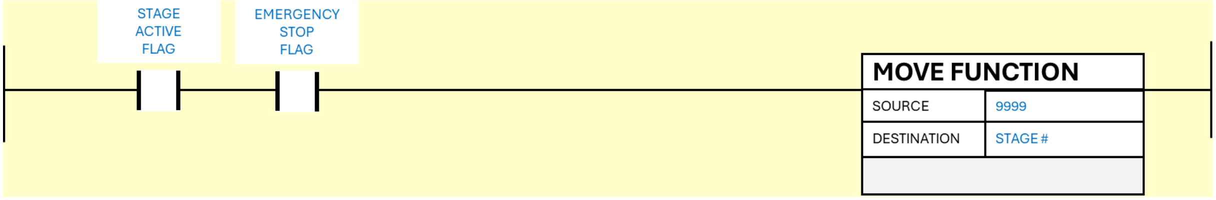
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



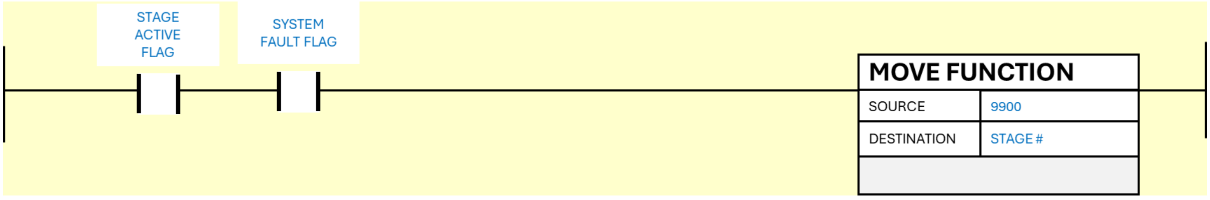
RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.



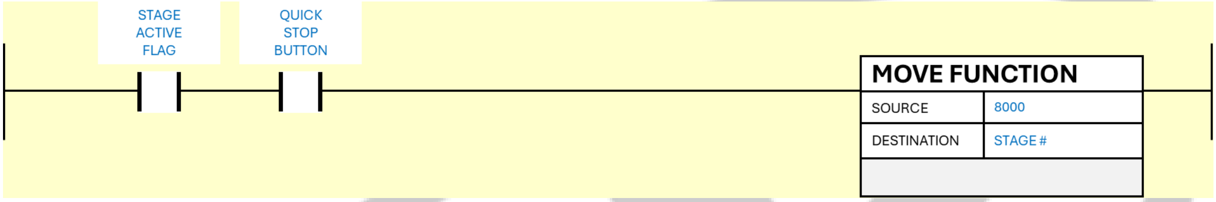
RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



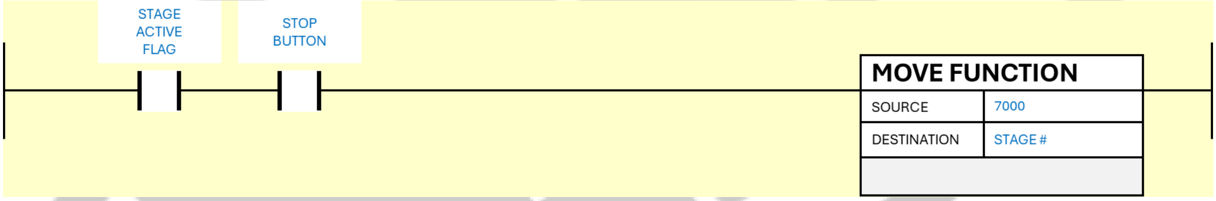
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



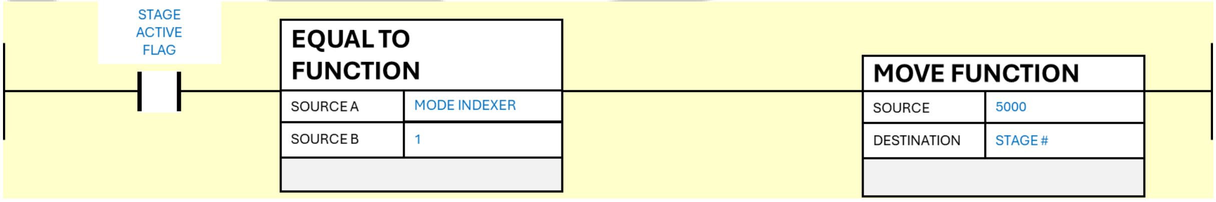
RUNG DESCRIPTION: The following rung is for progressing to the Quick Stop Stage.



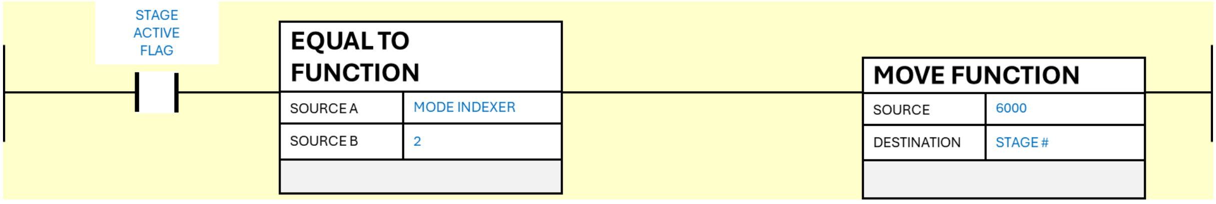
RUNG DESCRIPTION: The following rung is for progressing to the Normal Stop Stage.



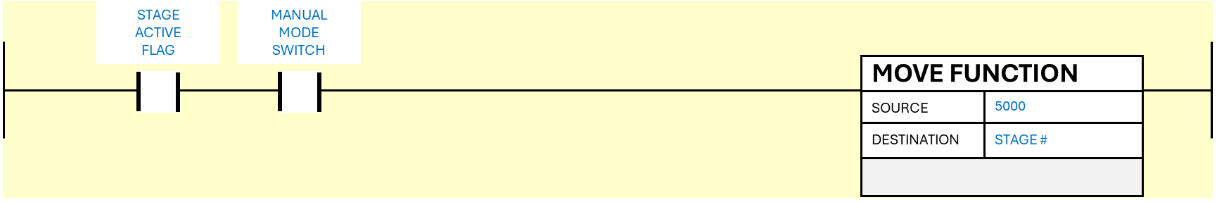
RUNG DESCRIPTION: The following rung if for progressing to the Run – MANUAL Stage using a Mode Indexing variable.



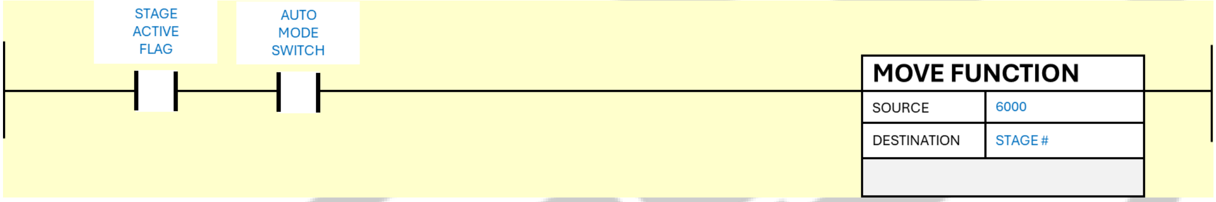
RUNG DESCRIPTION: The following rung if for progressing to the Run – AUTO Stage using a Mode Indexing variable.



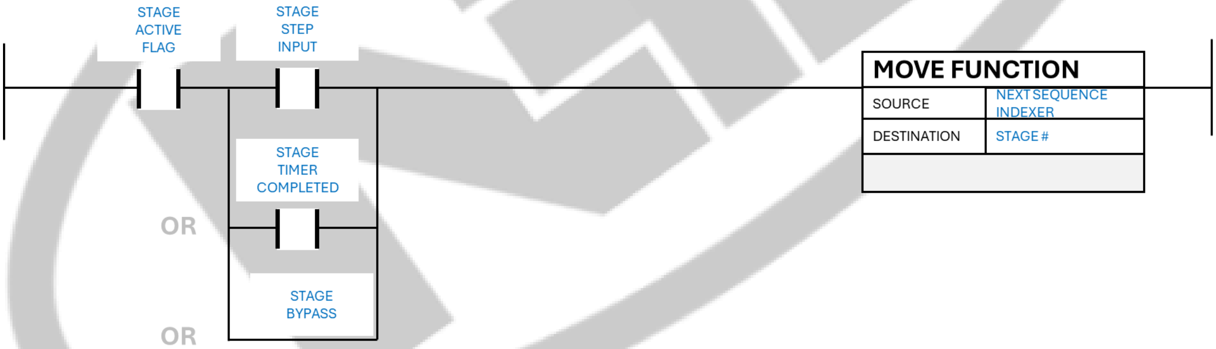
RUNG DESCRIPTION: The following rung if for progressing to the Run – MANUAL Stage using a Switch.



RUNG DESCRIPTION: The following rung if for progressing to the Run – AUTO Stage using a Switch.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

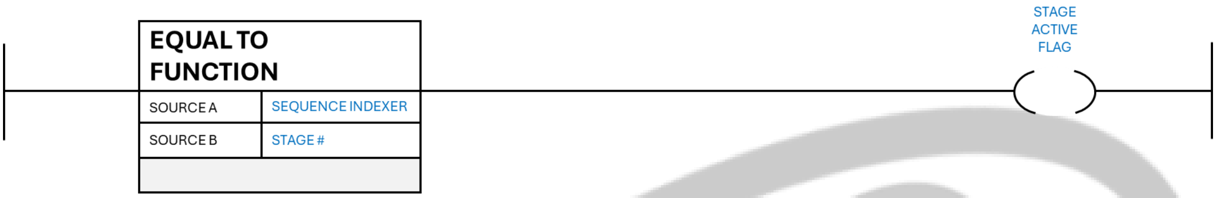


RUNNING ▶ MANUAL (Index: 5000)

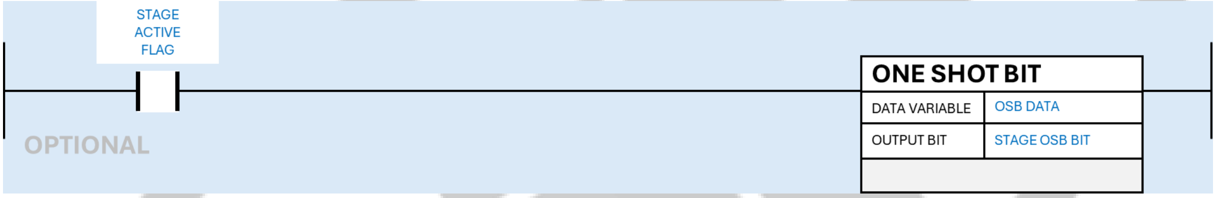
==== SEQUENCING [RUNNING - MANUAL] ====

SECTION DESCRIPTION: The following section is for running the System in Manual Mode.

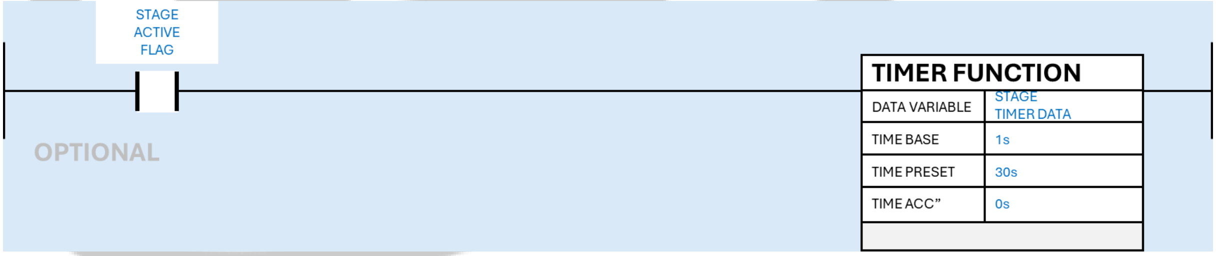
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.

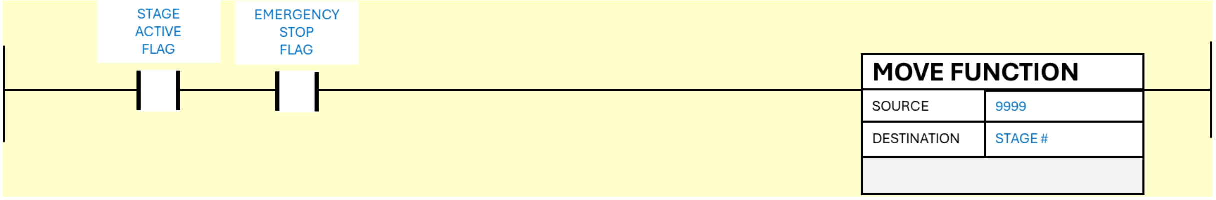


RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.

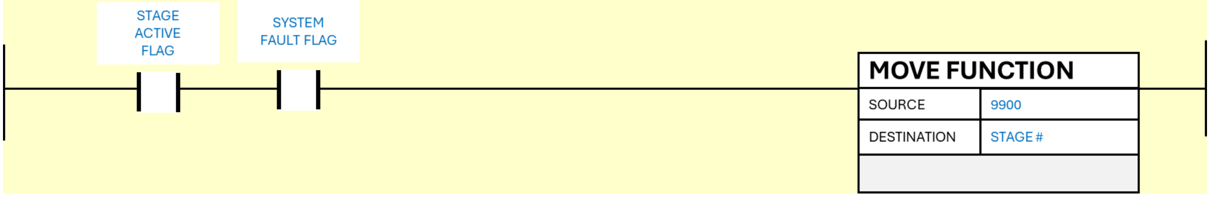


RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.

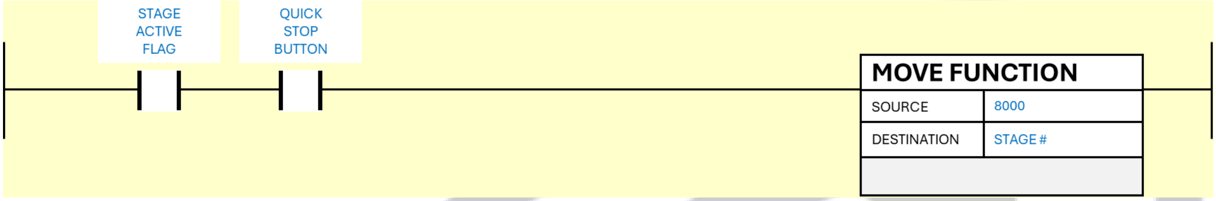
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



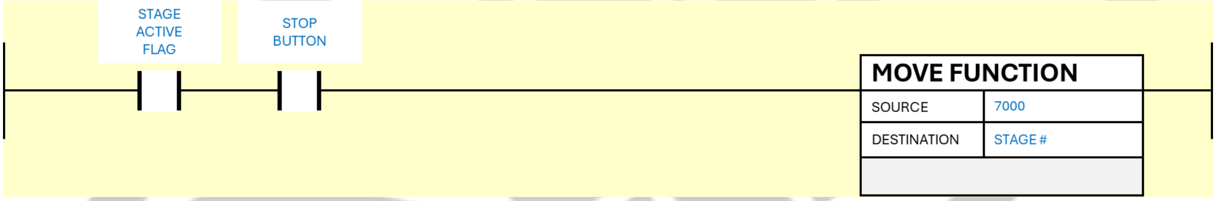
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



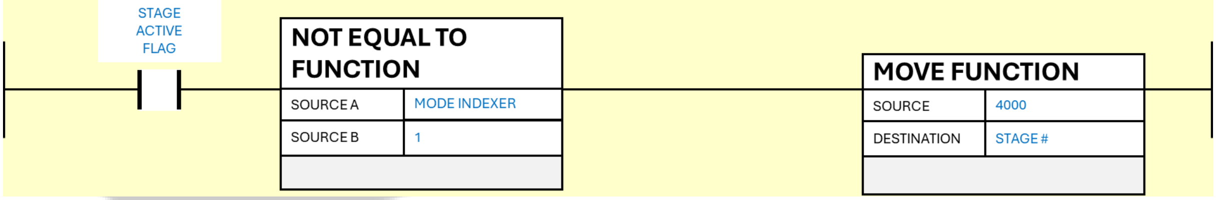
RUNG DESCRIPTION: The following rung is for progressing to the Quick Stop Stage.



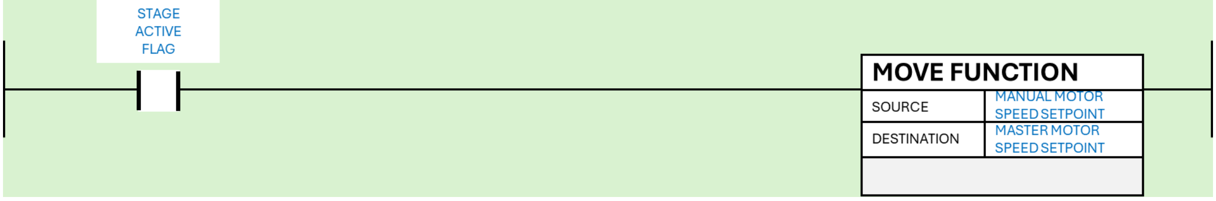
RUNG DESCRIPTION: The following rung is for progressing to the Normal Stop Stage.



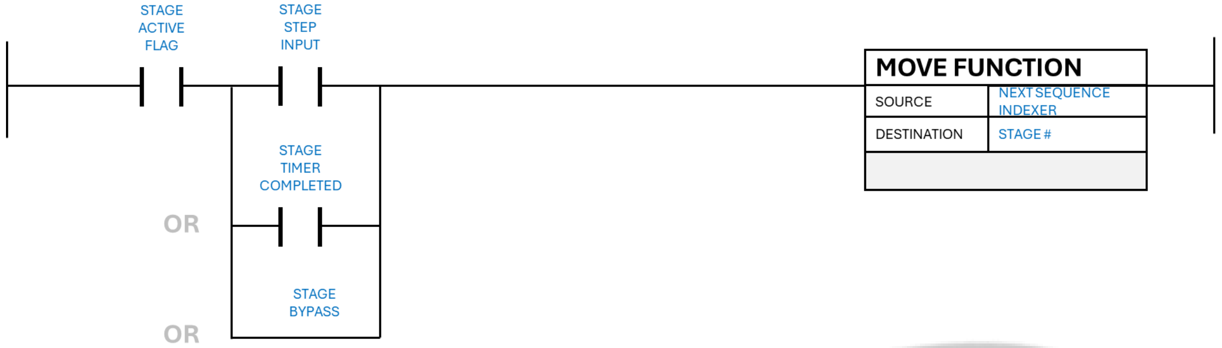
RUNG DESCRIPTION: The following rung is for detecting that the mode has been changed and progressing back to the Run Initial Stage



RUNG DESCRIPTION: The following rung is for setting the speed and or state of an output.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

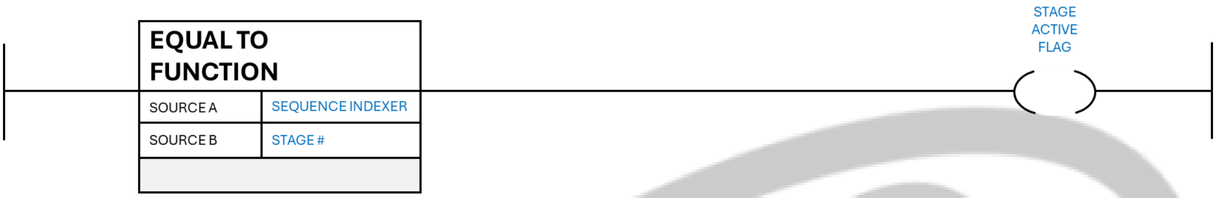


RUNNING ▶ AUTO (Index: 6000)

==== SEQUENCING [RUNNING - AUTO] ====

SECTION DESCRIPTION: The following section is for running the System in Auto Mode.

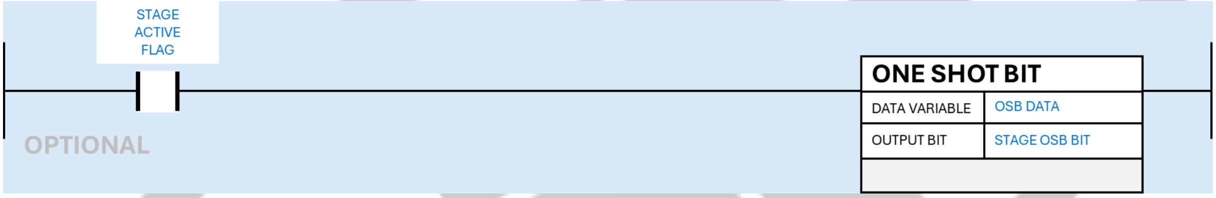
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



EQUAL TO FUNCTION

SOURCE A	SEQUENCE INDEXER
SOURCE B	STAGE #

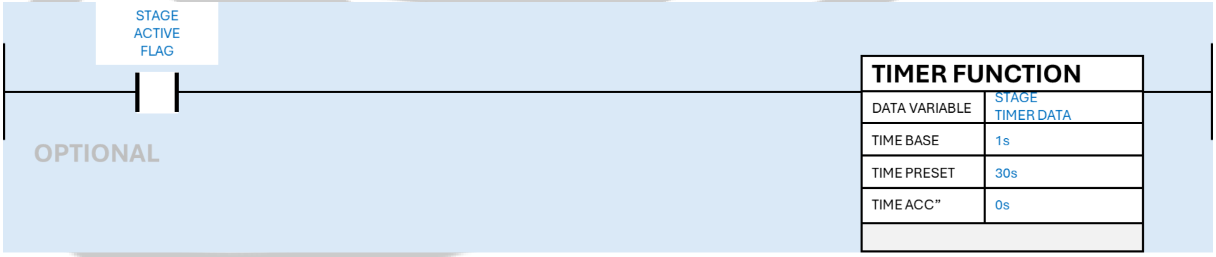
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



ONE SHOT BIT

DATA VARIABLE	OSB DATA
OUTPUT BIT	STAGE OSB BIT

RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.

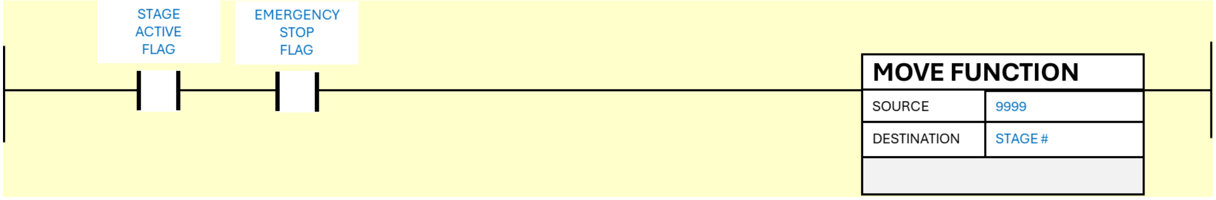


TIMER FUNCTION

DATA VARIABLE	STAGE TIMER DATA
TIME BASE	1s
TIME PRESET	30s
TIME ACC"	0s

RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.

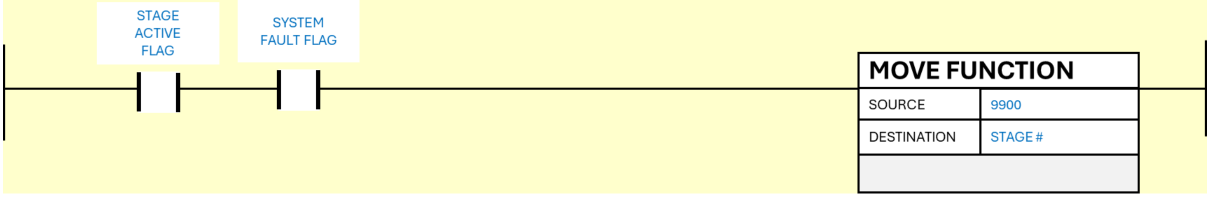
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



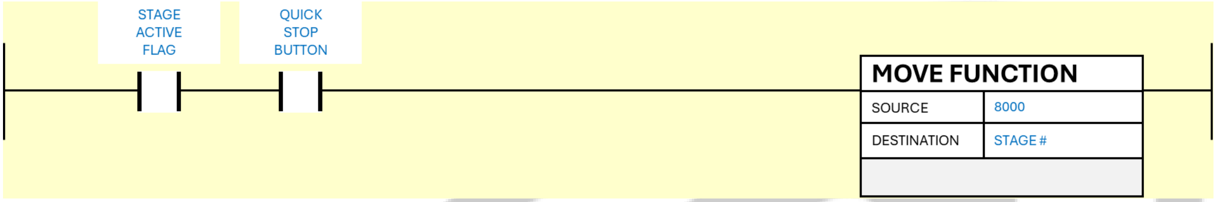
MOVE FUNCTION

SOURCE	9999
DESTINATION	STAGE #

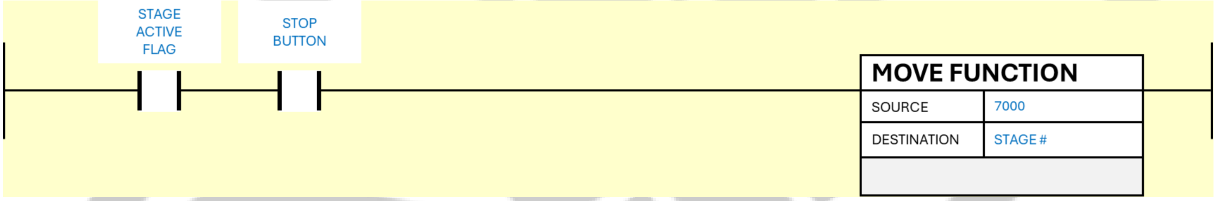
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



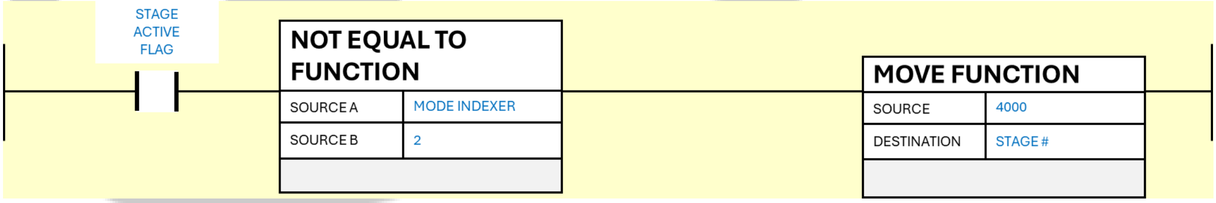
RUNG DESCRIPTION: The following rung is for progressing to the Quick Stop Stage.



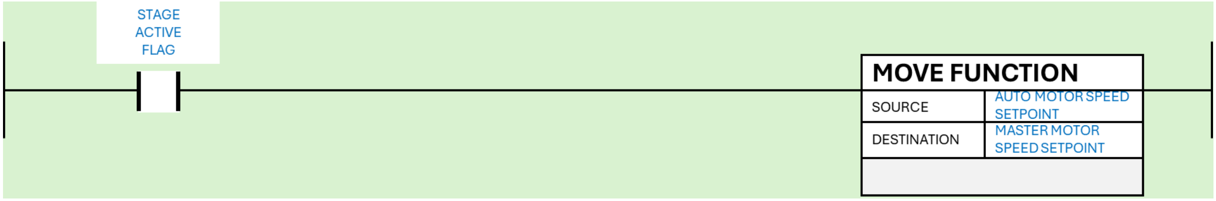
RUNG DESCRIPTION: The following rung is for progressing to the Normal Stop Stage.



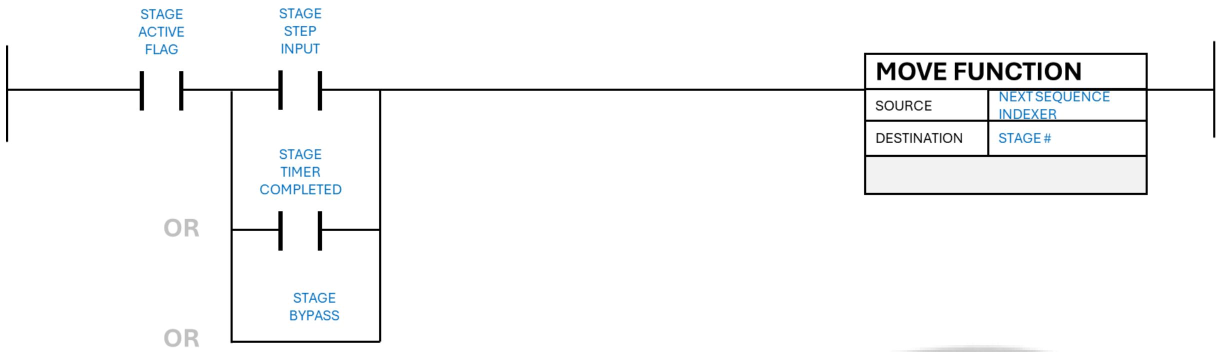
RUNG DESCRIPTION: The following rung is for detecting that the mode has been changed and progressing back to the Run Initial Stage.



RUNG DESCRIPTION: The following rung is for setting the speed and or state of an output.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

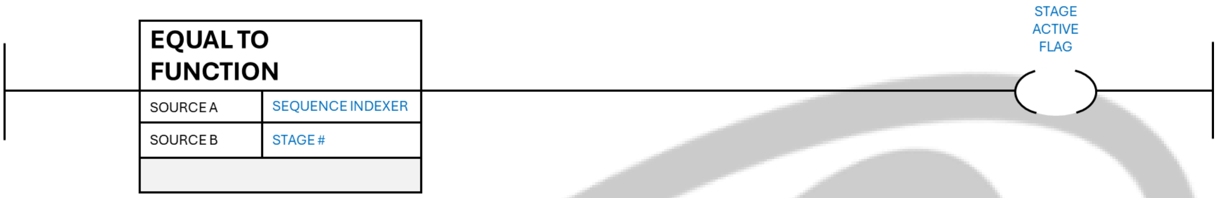


WARM-DOWN (Index: 7000)

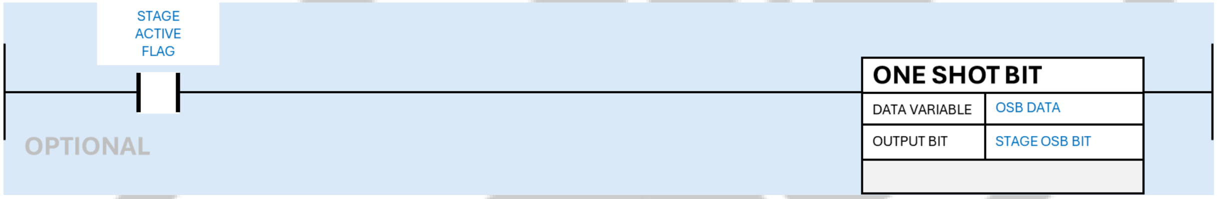
==== SEQUENCING [WARM-DOWN] ====

SECTION DESCRIPTION: The following section is for allowing system components to cool down under controlled measures.

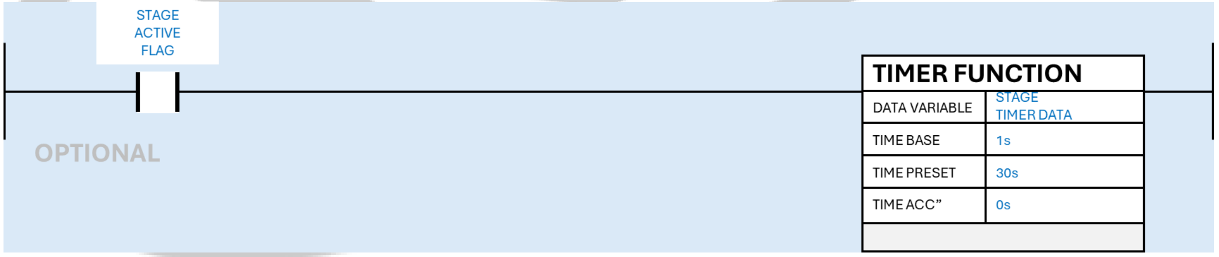
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



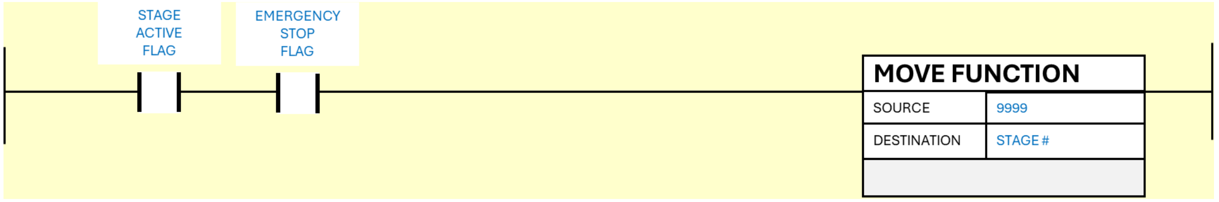
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



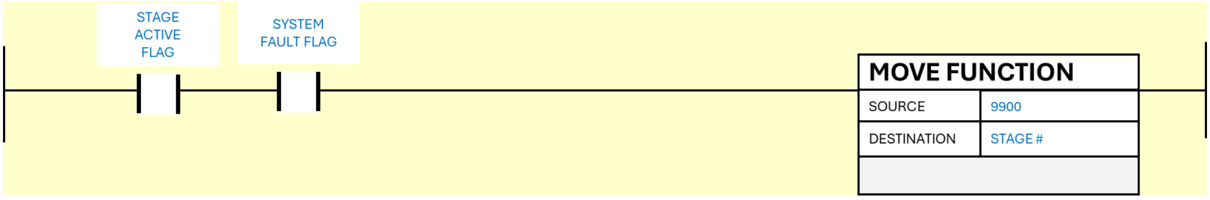
RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.



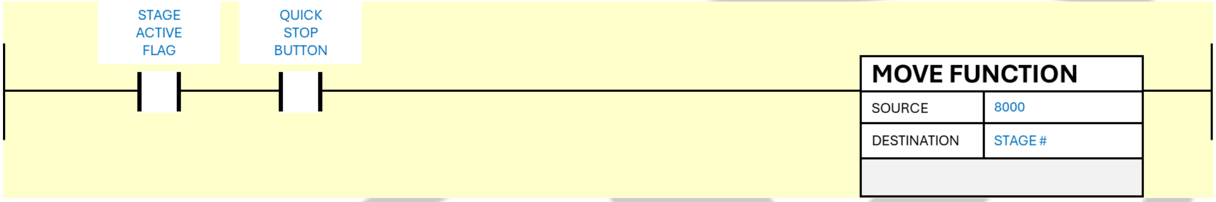
RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



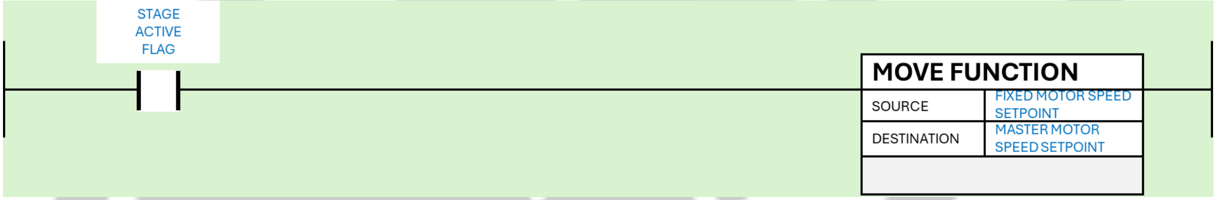
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



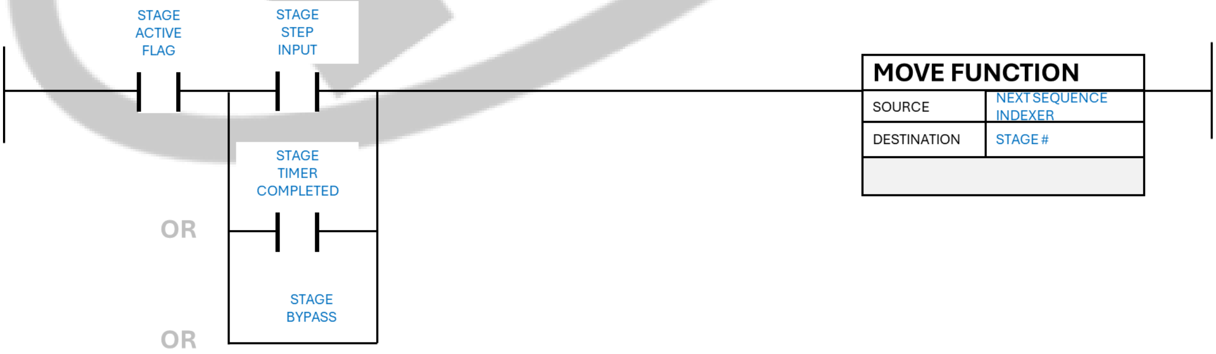
RUNG DESCRIPTION: The following rung is for progressing to the Quick Stop Stage.



RUNG DESCRIPTION: The following rung is for setting the speed and or state of an output.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

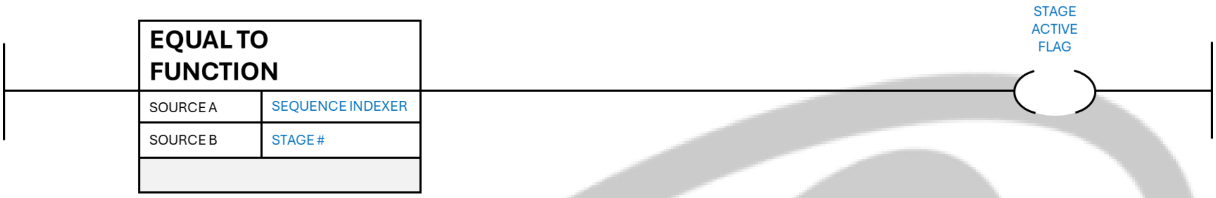


STOPPING (Index: 8000)

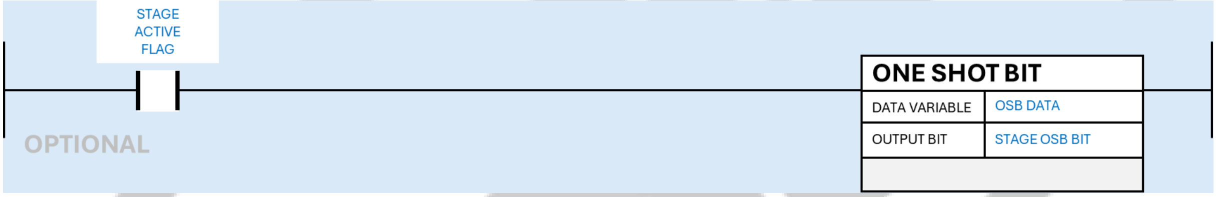
==== SEQUENCING [STOPPING] ====

SECTION DESCRIPTION: The following section is typically for completing System / Assembly Post Run Checks and or de-energizing supplies.

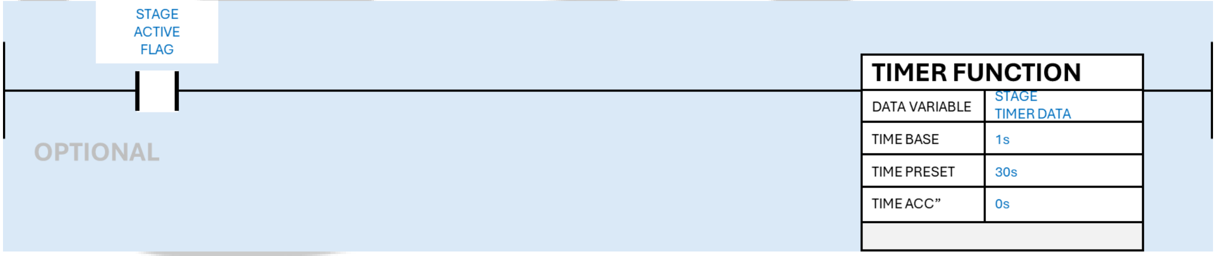
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



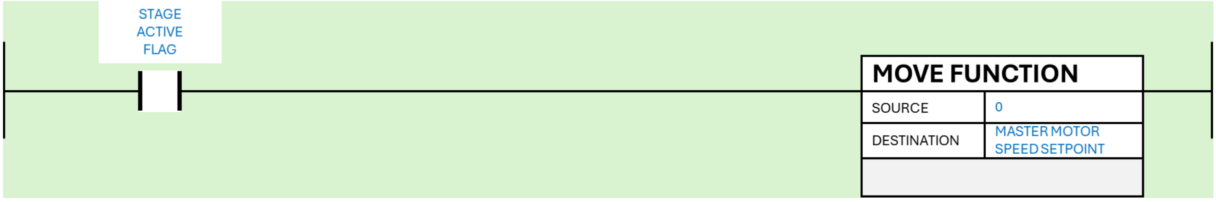
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.

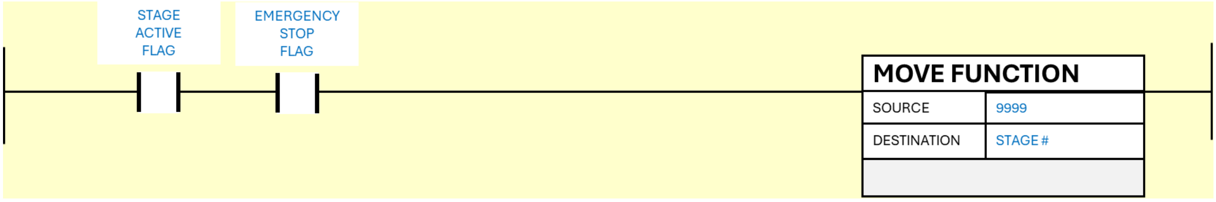


RUNG DESCRIPTION: The following rung is for setting the speed and or state of an output.

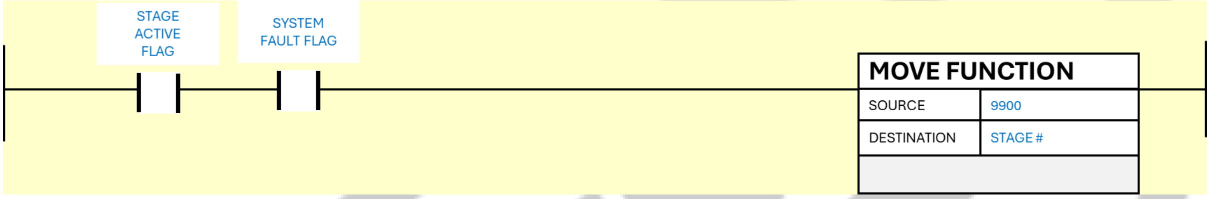


RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.

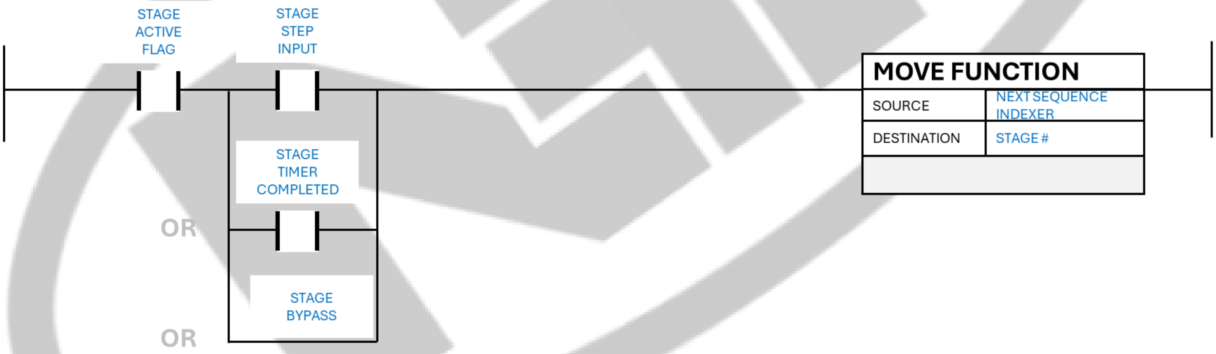
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

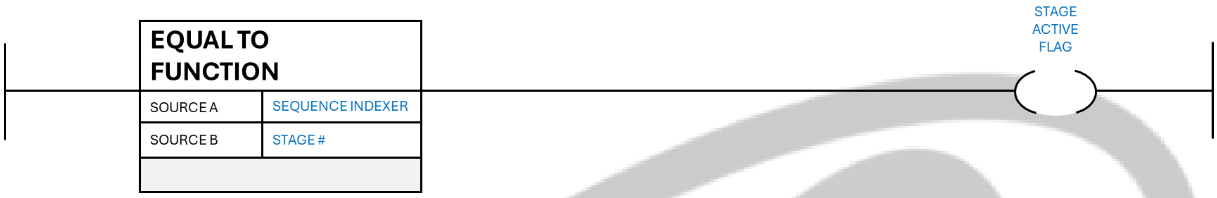


STOPPED (Index: 9000)

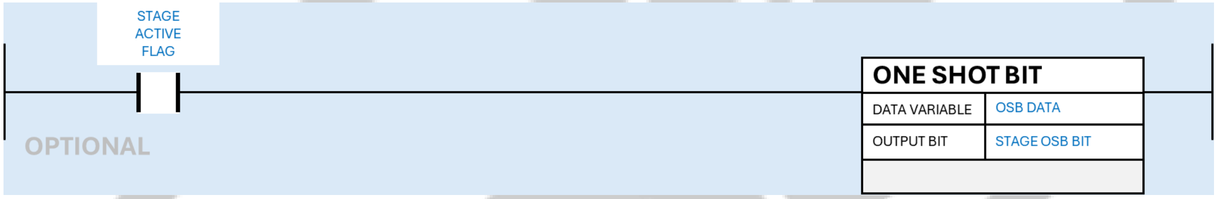
==== SEQUENCING [STOPPED] ====

SECTION DESCRIPTION: The following section is typically for completing System / Assembly Post-Stopped Checks.

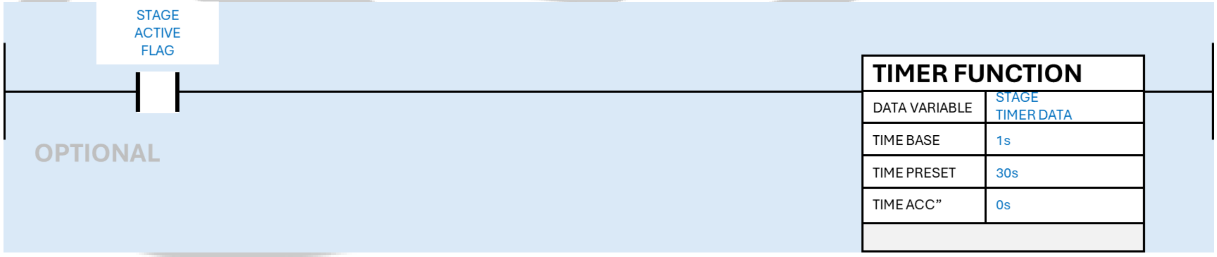
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.

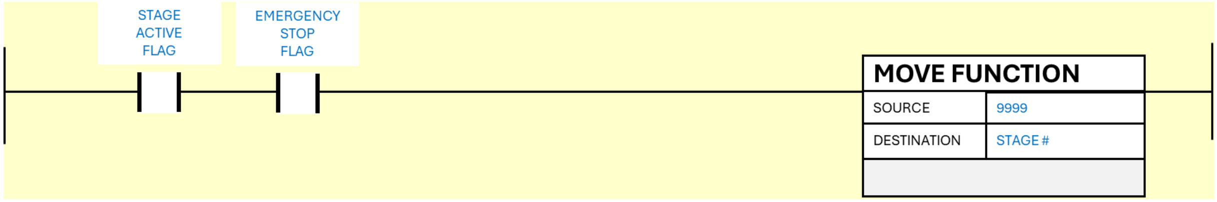


RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.

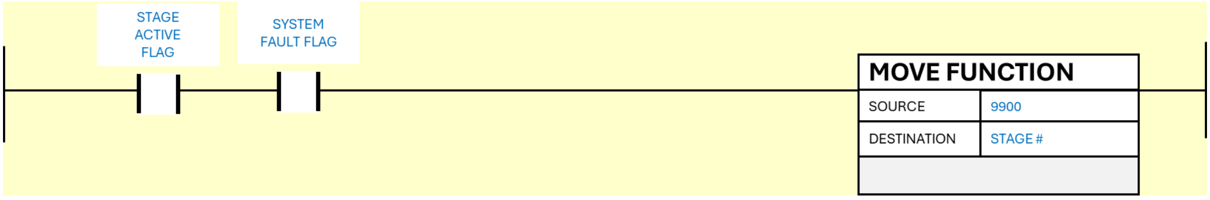


RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.

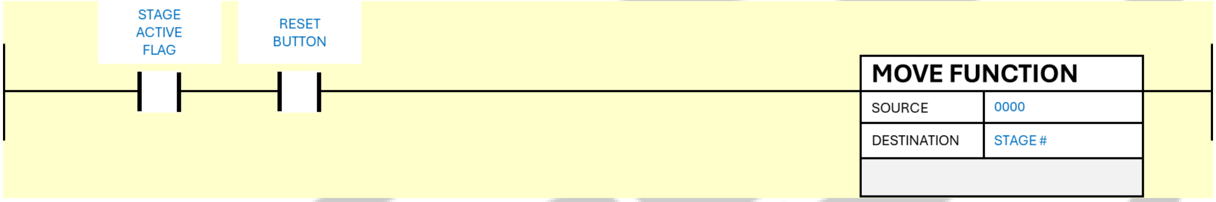
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



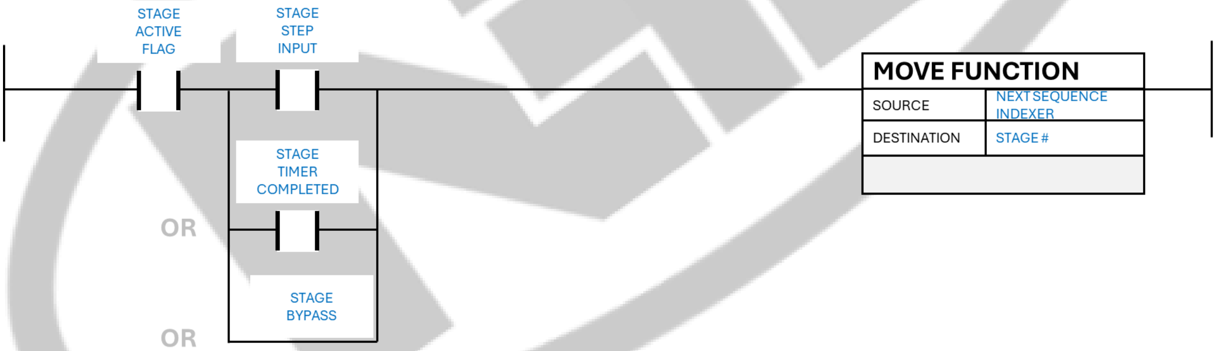
RUNG DESCRIPTION: The following rung is used for skipping to the Equipment Fault Stage, if any of the Equipment Fault Flags are triggered.



RUNG DESCRIPTION: The following rung is for detecting that the fault has been acknowledged and progressing back to the Standby Stage.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

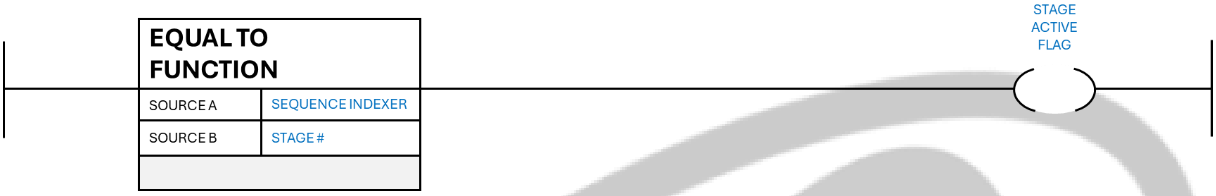


OPERATOR ATTENTION (Index: 9800)

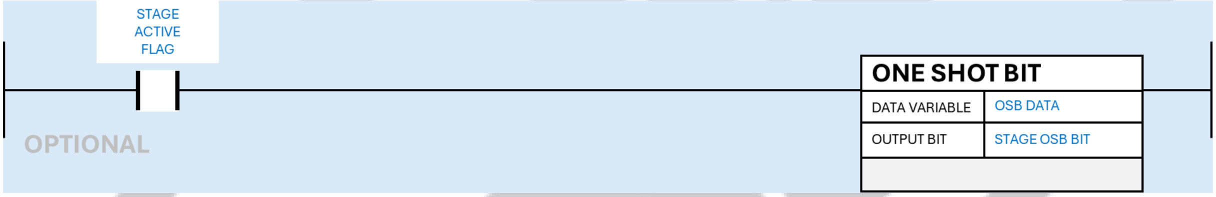
==== SEQUENCING [OPERATOR ATTENTION] ====

SECTION DESCRIPTION: The following section is typically used for notifying the operators of tasks which need to be completed before further operation.

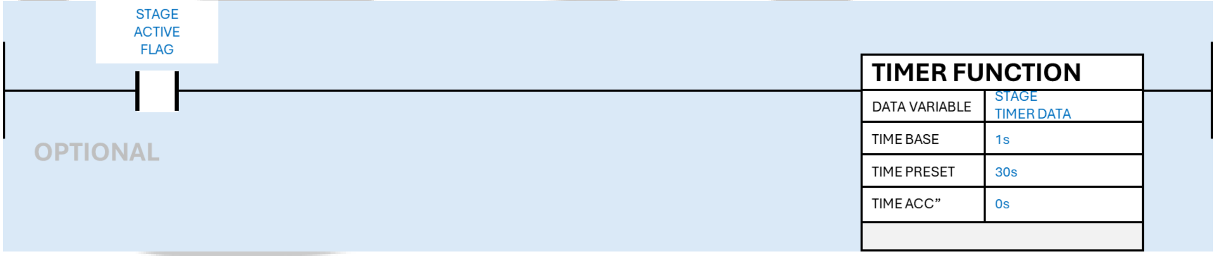
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



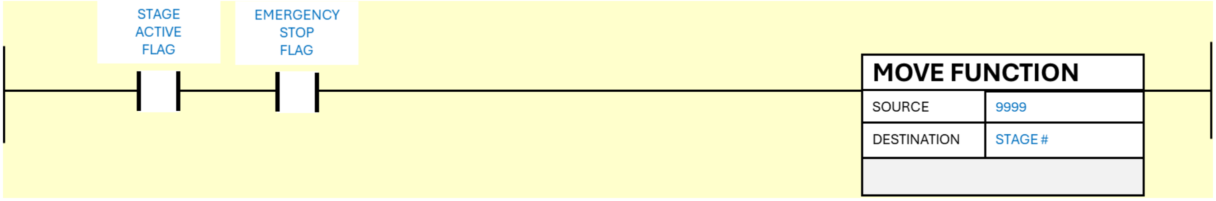
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



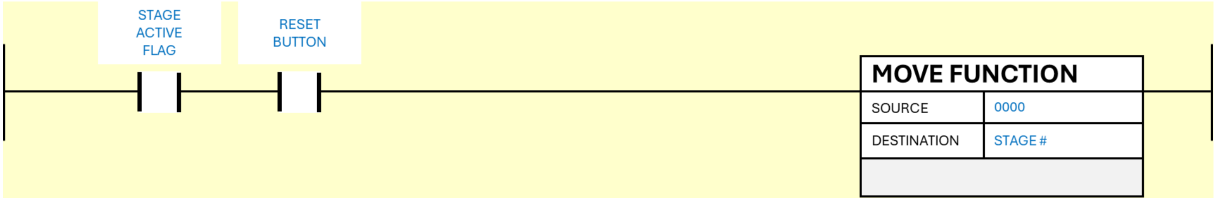
RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.



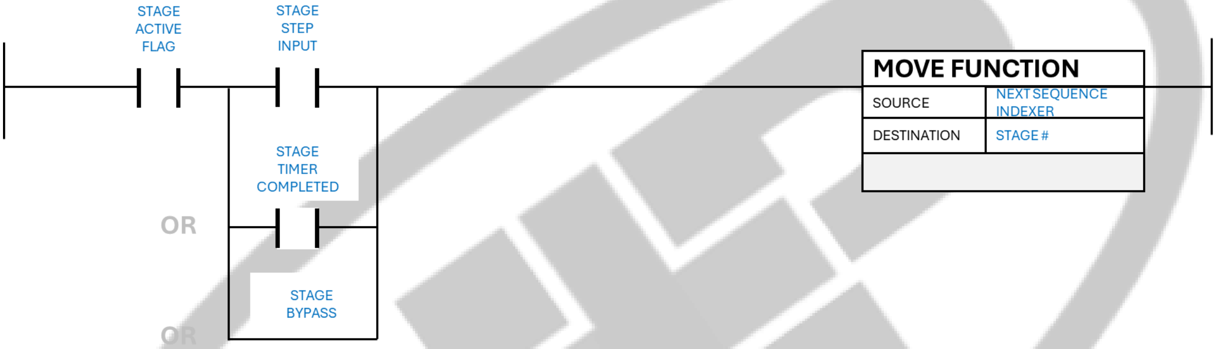
RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.
IMPORTANT: This is not a replacement for a certified Emergency Stop system.



RUNG DESCRIPTION: The following rung is for detecting that the call has been acknowledged and progressing back to the Standby Stage.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

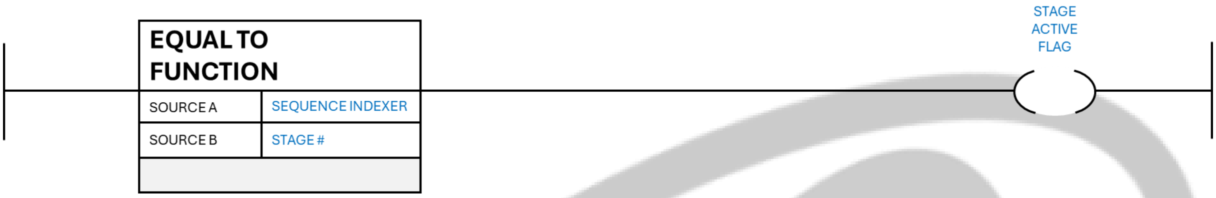


SYSTEM FAULT (Index: 9900)

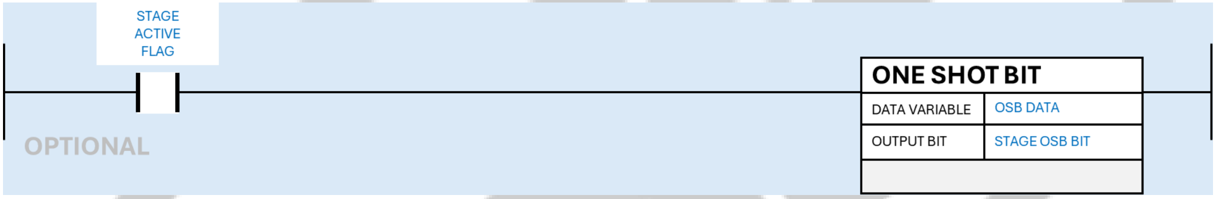
==== SEQUENCING [SYSTEM FAULT] ====

SECTION DESCRIPTION: The following section is typically used for shutting equipment down and informing the operator / maintenance of System / Assembly faults.

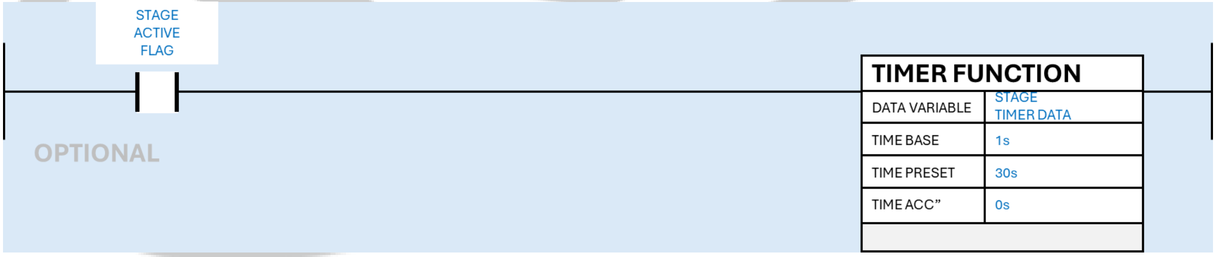
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.

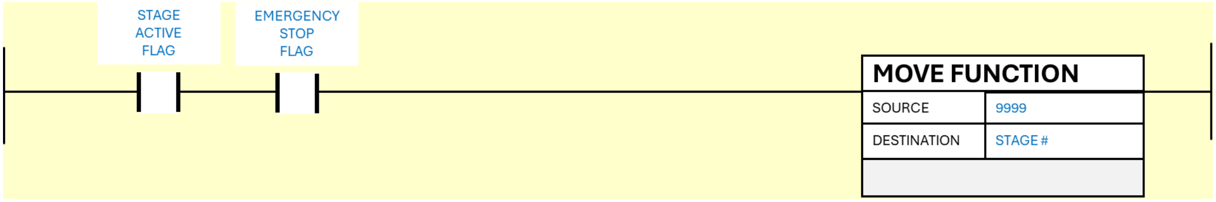


RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.

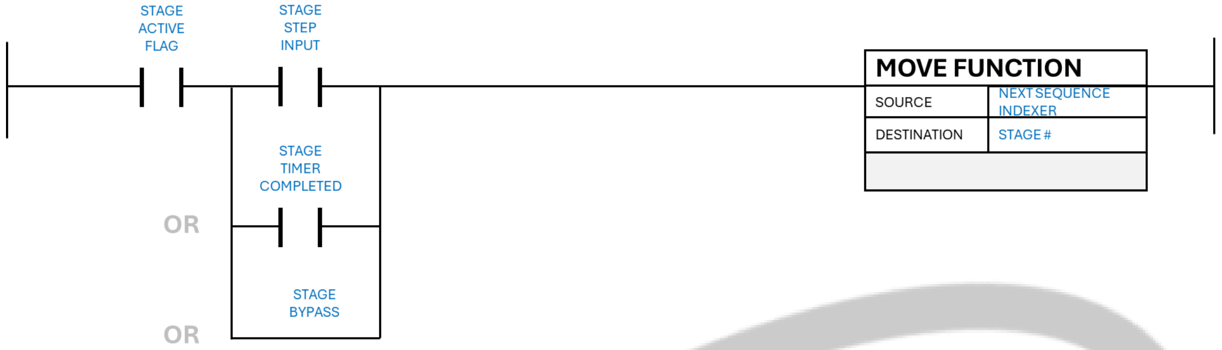


RUNG DESCRIPTION: The following rung is used for skipping to the Emergency Stage, if the Emergency Flag (the emergency input(s)) are triggered.

IMPORTANT: This is not a replacement for a certified Emergency Stop system.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).

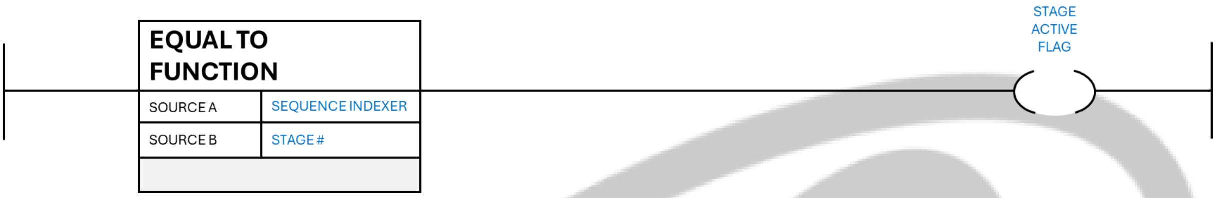


EMERGENCY STOP (Index: 9999)

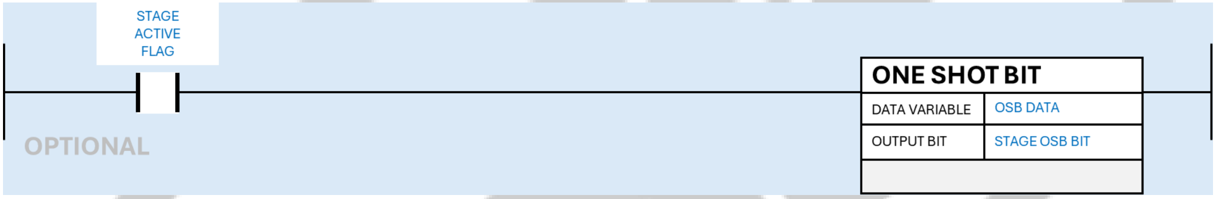
==== SEQUENCING [EMERGENCY STOP] ====

SECTION DESCRIPTION: The following section is typically used for shutting equipment down and informing the operator / maintenance of System / Assembly Emergency Stop(s).

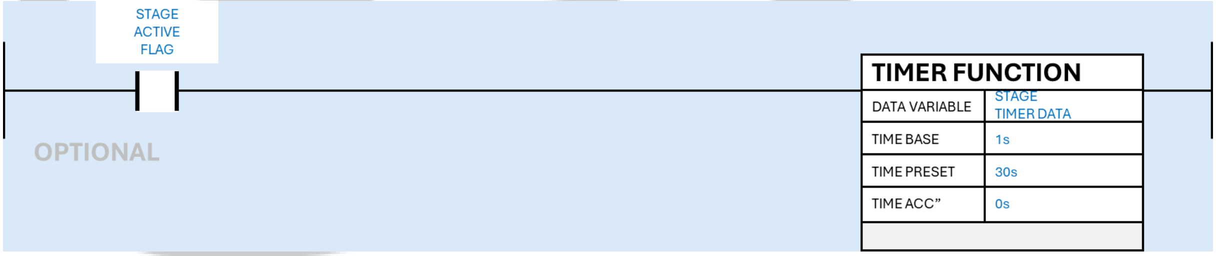
RUNG DESCRIPTION: The following rung is for detecting the Stage Index value and setting a Stage Flag for the remaining rungs.



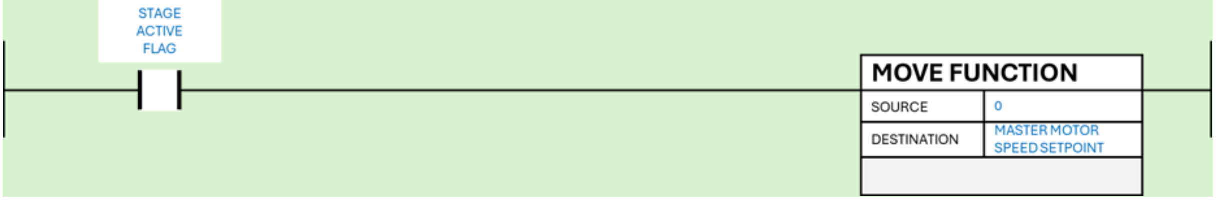
RUNG DESCRIPTION: The following rung is for setting a One Shot Bit, which may be required by certain functions like Timers, Latching Outputs and so forth.



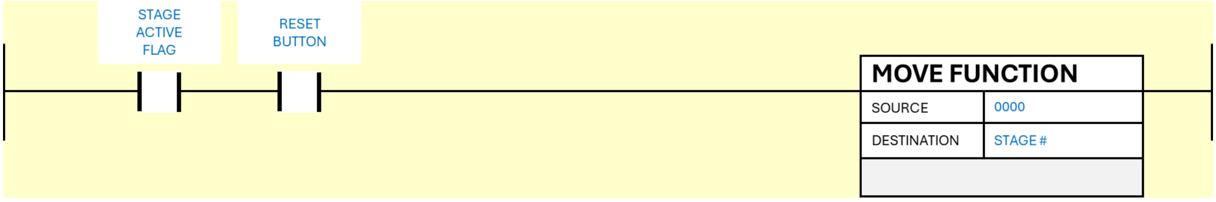
RUNG DESCRIPTION: The following rung is for setting up Timers within the Stage. These are typically required for (although not restricted to) automatically skipping to the next Stage.



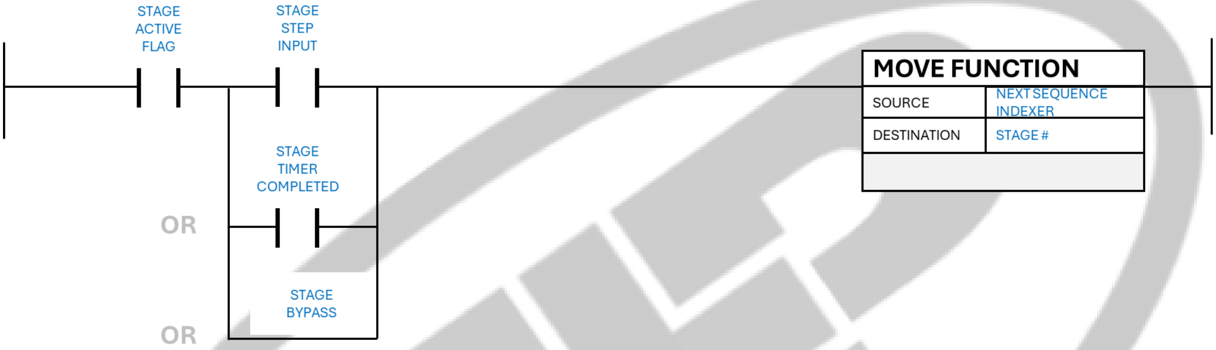
RUNG DESCRIPTION: The following rung is for setting the speed and or state of an output.



RUNG DESCRIPTION: The following rung is for detecting that the call has been acknowledged and progressing back to the Standby Stage.



RUNG DESCRIPTION: The following rung is for progressing to the next stage. This can be triggered by an input, timer and or bypass (skipping).



Outputs Section

The output section is somewhat like the inputs section, however, focuses only on mapping Output variables with working variables. Please refer to the Input Section for further information.

Programming

==== OUTPUTS ====

SECTION DESCRIPTION: The following section is for programming all outputs specific for the control of this program module. Any communal outputs should be included in a separate Input Module.

Add-On Instruction (AOI) Layouts

Add On Instructions (AOIs) refers to modules of code which are used repeatedly within a PLC. These can also be known as Function Blocks within some PLCs. An example of when a AOI may be used could be when programming a series of conveyors which are designed to run automatically. The programming for each physical section will be similar (like motor ramping, run-out timers etc) however the inputs and outputs will be different.

A programmer can use a AOI to save time writing specific code for each physical section of conveyor and program a module which can be deployed to multiple sections. This then standardises the functionality across all sections as well as making it easier to manage programming changes. Ideally, each AOI should be programmed in the same way as describe before using sequence programming. However, in some circumstances given the type of function (like indicator stack module) this will not be required.

Mode Selection Module

The following Add On Instruction (function block) is for converting and Manual / Auto sustained switched Inputs into a working index.

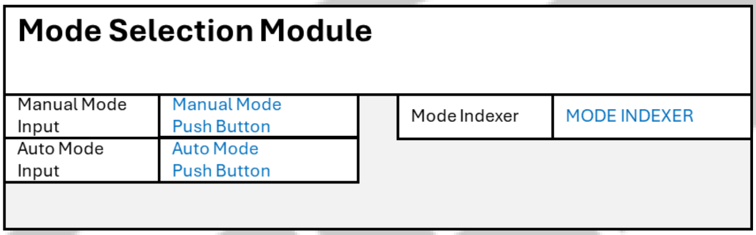
The mode layout for this block is as follows:

(0) Default – No mode selected.

(1) Manual Mode

(2) Auto Mode

Programming

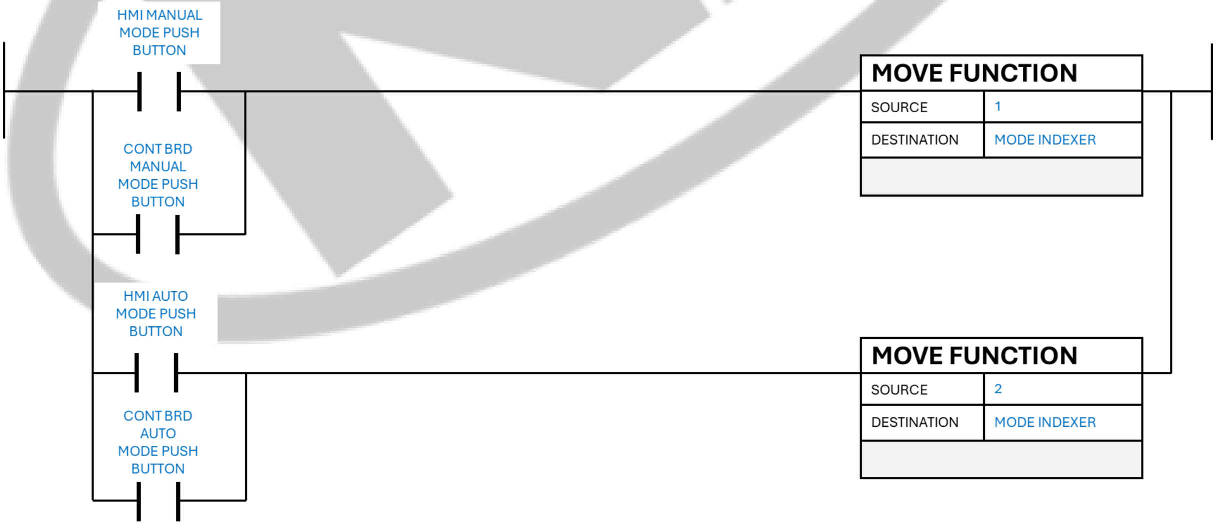


Mode Selection Module

Manual Mode Input	Manual Mode Push Button	Mode Indexer	MODE INDEXER
Auto Mode Input	Auto Mode Push Button		

==== MODE SELECTION MODULE ====

SECTION DESCRIPTION: The following mode is for converting a mode selection switch into working mode index value.


HMI MANUAL MODE PUSH BUTTON
CONT BRD MANUAL MODE PUSH BUTTON
HMI AUTO MODE PUSH BUTTON
CONT BRD AUTO MODE PUSH BUTTON
MOVE FUNCTION
SOURCE
1
DESTINATION
MODE INDEXER
MOVE FUNCTION
SOURCE
2
DESTINATION
MODE INDEXER

Indicator Stack Translation Module

This Add On Instruction (function block) is for converting a Sequencing Index value into light stack outputs.

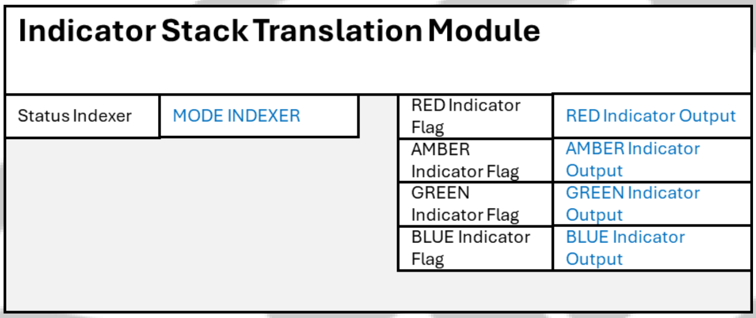
RED INDICATOR – System is Stopped and or has detected an Emergency Stop.

AMBER INDICATOR – System has stopped and requires operator intervention (for example refilling / restocking equipment consumables)

GREEN INDICATOR – System is running. Flashing denotes a change in state (Warm-up / Warm-down)

BLUE INDICATOR – System is in Standby and ready for use.

Programming



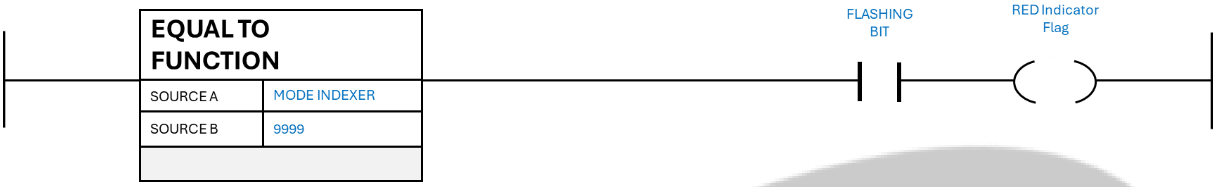
Indicator Stack Translation Module

Status Indexer	MODE INDEXER	RED Indicator Flag	RED Indicator Output
		AMBER Indicator Flag	AMBER Indicator Output
		GREEN Indicator Flag	GREEN Indicator Output
		BLUE Indicator Flag	BLUE Indicator Output

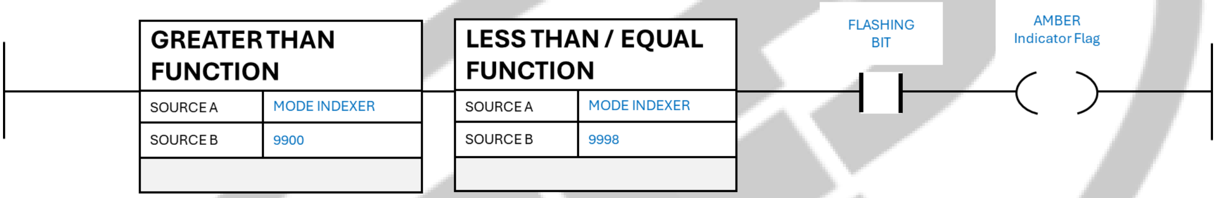
==== INDICATOR STACK TRANSLATION MODULE ====

SECTION DESCRIPTION: The following mode is for converting a Sequence Indexer into outputs which are linked to an Indicator Stack.

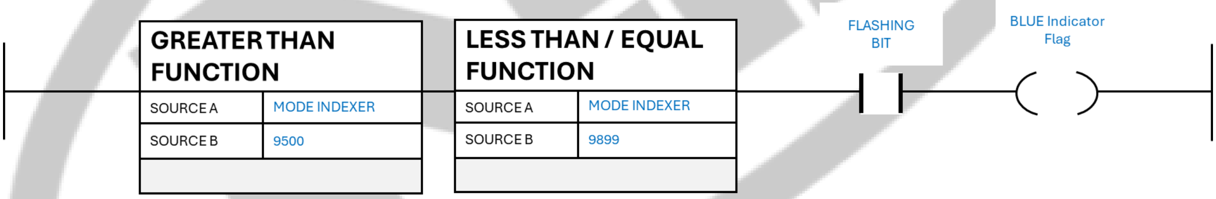
RUNG DESCRIPTION: Red Indicator Flashing Programming



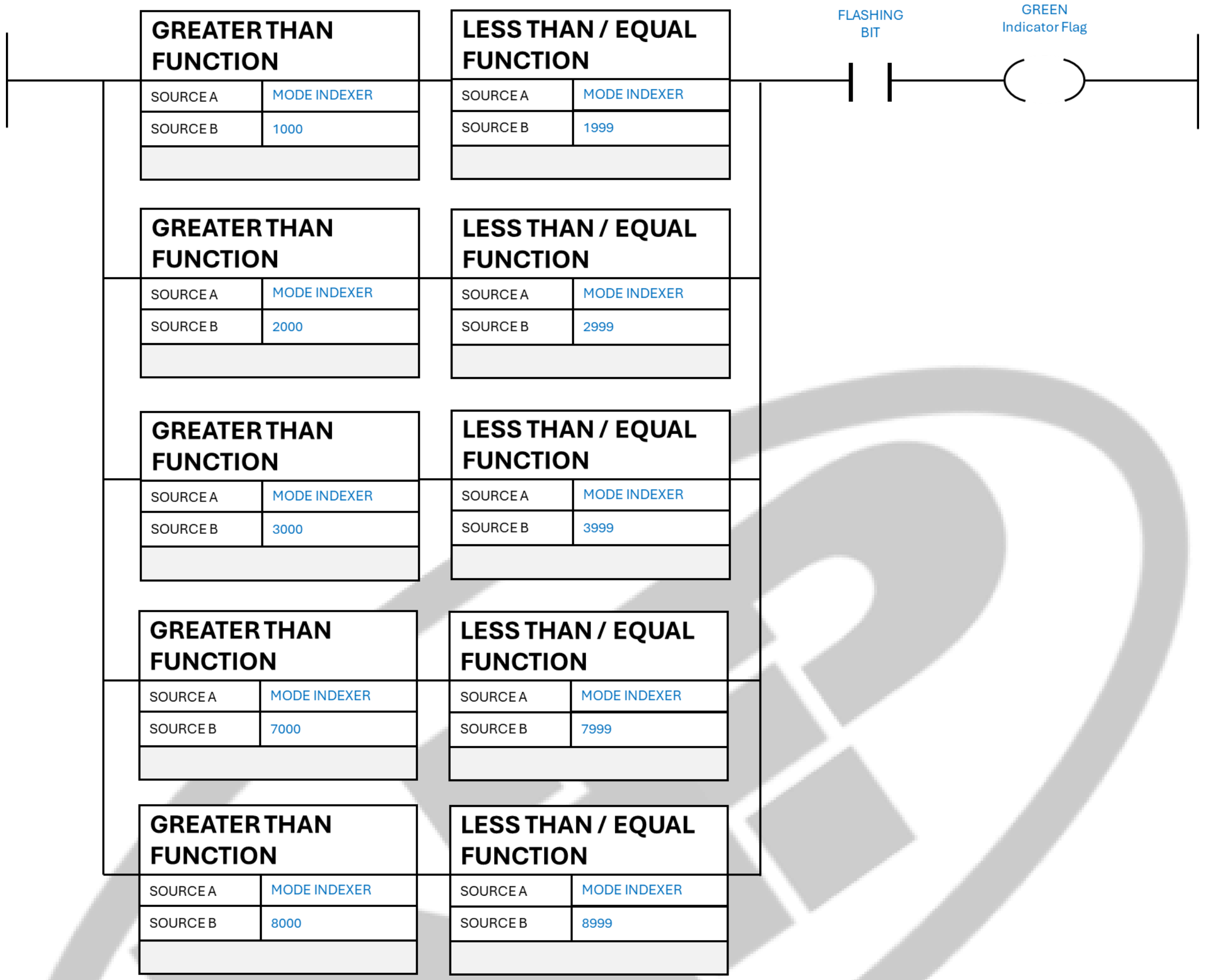
RUNG DESCRIPTION: Amber Indicator Flashing Programming



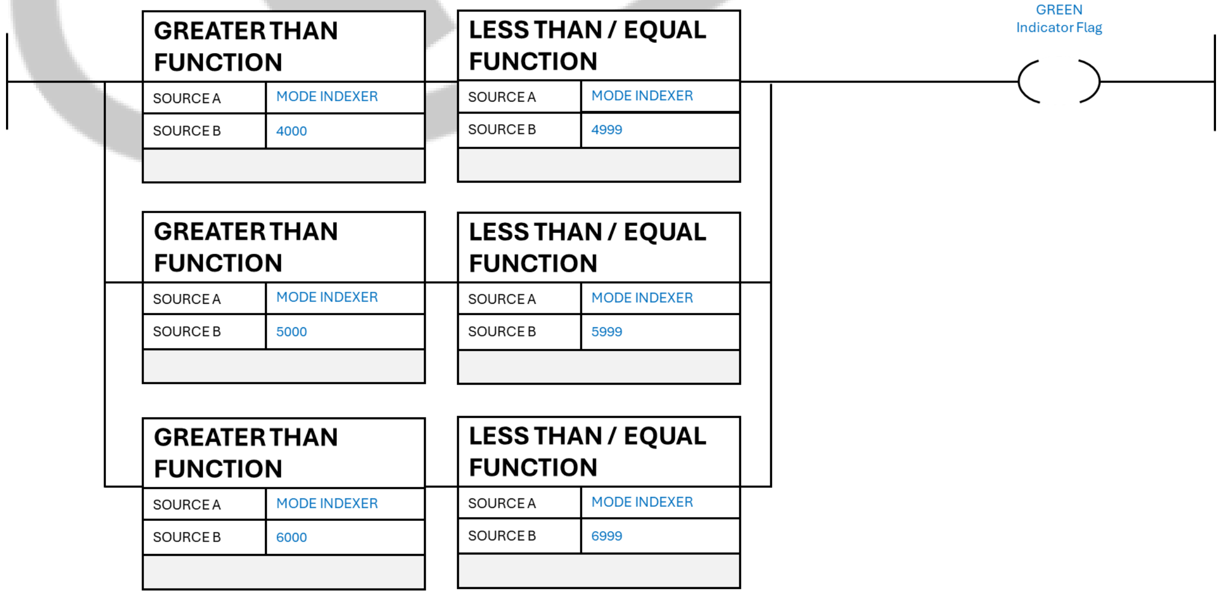
RUNG DESCRIPTION: Blue Indicator Flashing Programming



RUNG DESCRIPTION: Green Indicator Flashing Programming



RUNG DESCRIPTION: Green Indicator Solid Programming



Indicator Push Button Translation Module

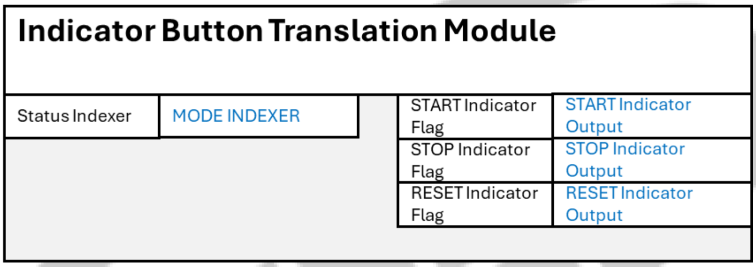
The following Add On Instruction (function block) is for linking a Sequence Index variable with the indicators on a Start / Stop / Reset button set.

START BUTTON – …

STOP BUTTON – …

RESET BUTTON – …

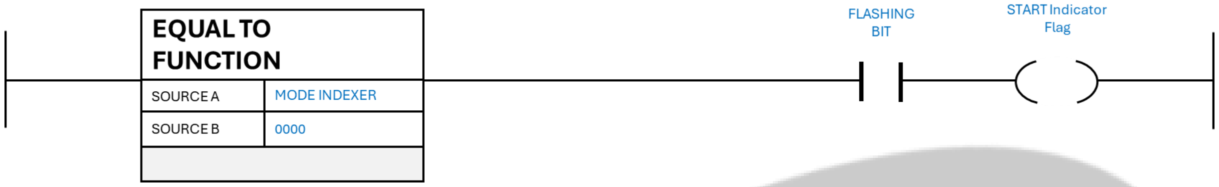
Programming



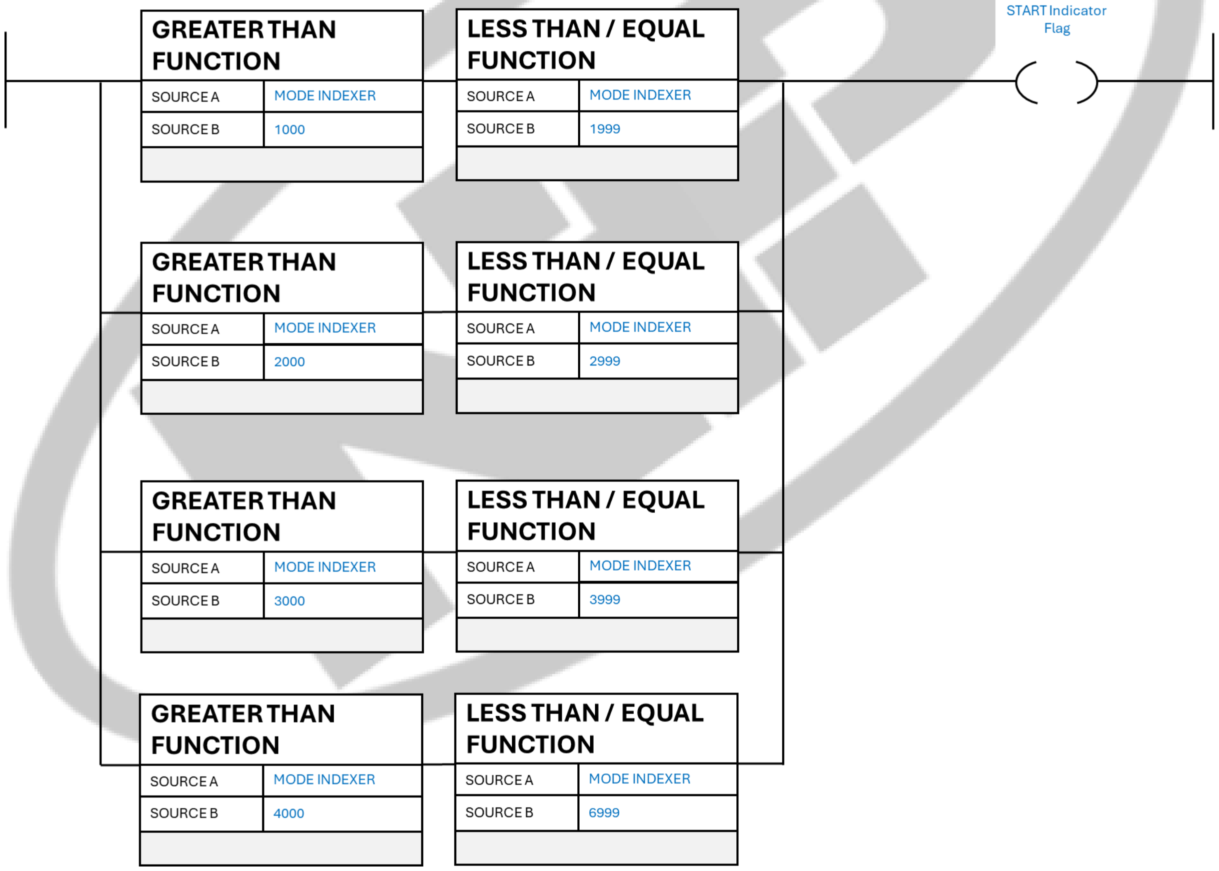
==== INDICATOR BUTTON TRANSLATION MODULE ====

SECTION DESCRIPTION: The following mode is for converting a Sequence Indexer into outputs which are linked to Start / Stop / Reset Button Set

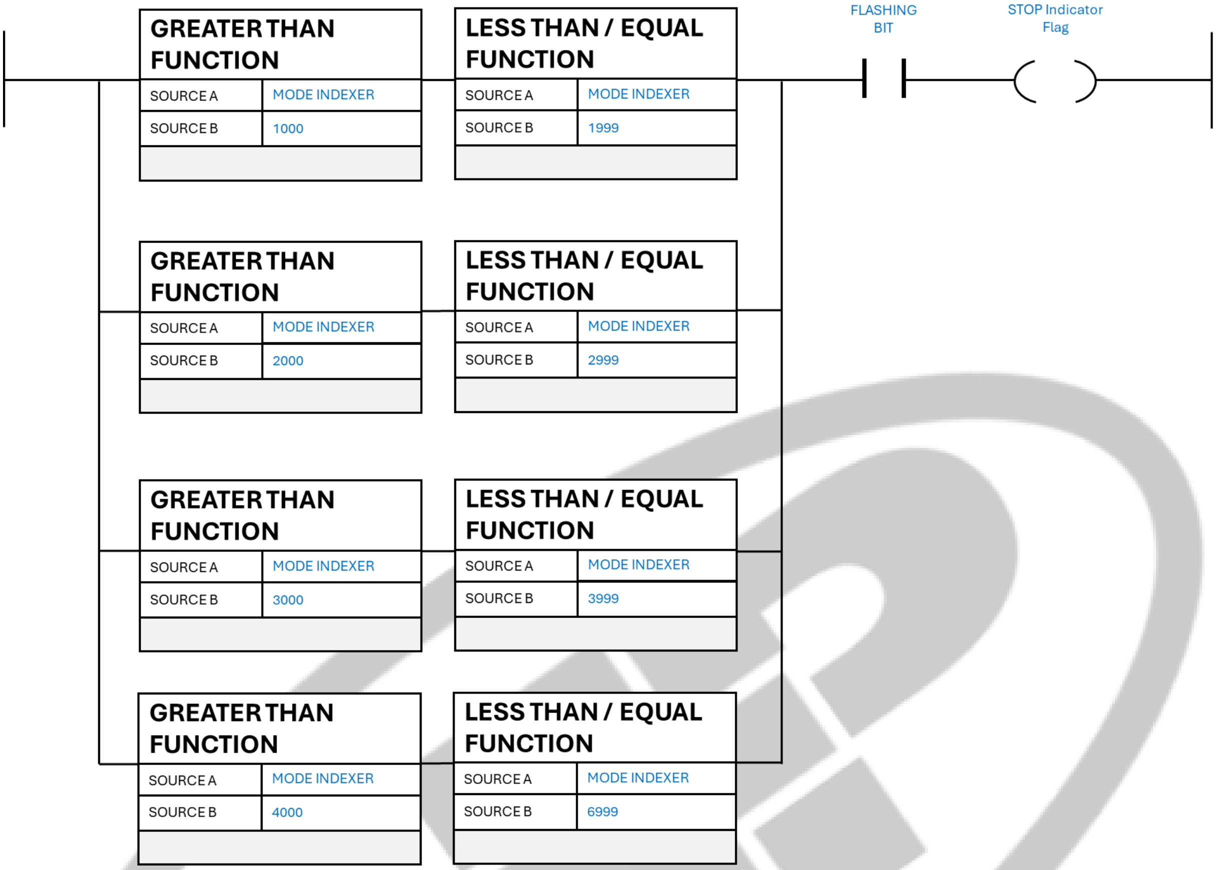
RUNG DESCRIPTION: Start Indicator Flashing Programming



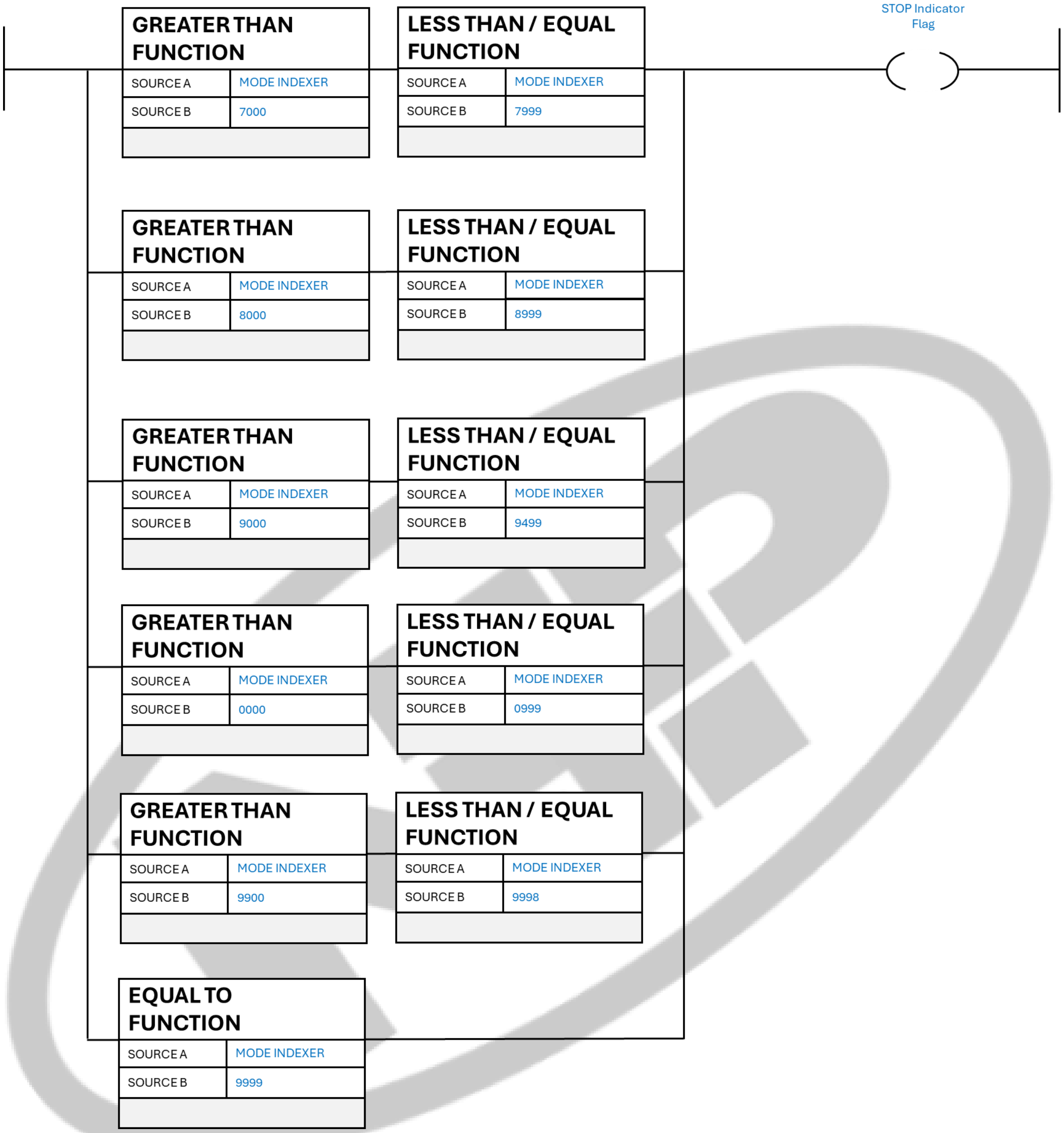
RUNG DESCRIPTION: Start Indicator Solid Programming



RUNG DESCRIPTION: Stop Indicator Flashing Programming



RUNG DESCRIPTION: Stop Indicator Solid Programming



STOP Indicator Flag

GREATER THAN FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	7000

LESS THAN / EQUAL FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	7999

GREATER THAN FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	8000

LESS THAN / EQUAL FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	8999

GREATER THAN FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	9000

LESS THAN / EQUAL FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	9499

GREATER THAN FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	0000

LESS THAN / EQUAL FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	0999

GREATER THAN FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	9900

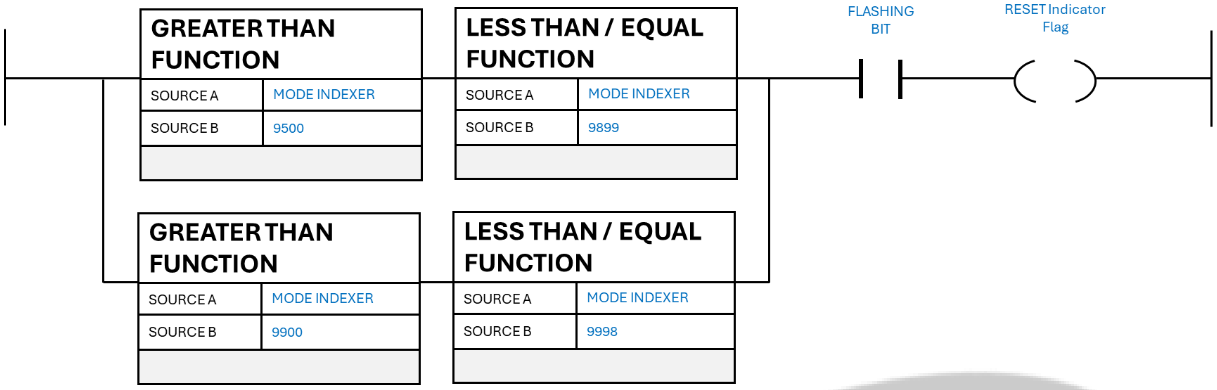
LESS THAN / EQUAL FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	9998

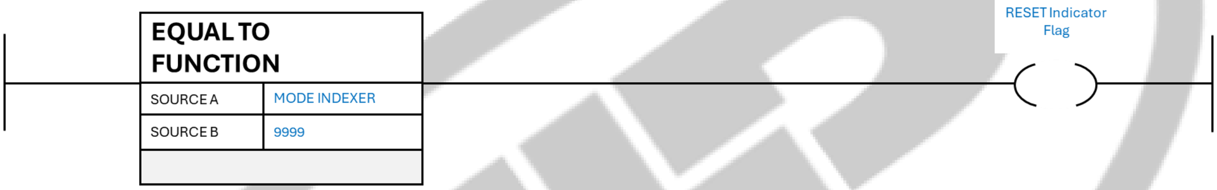
EQUAL TO FUNCTION

SOURCE A	MODE INDEXER
SOURCE B	9999

RUNG DESCRIPTION: Reset Indicator Flashing Programming

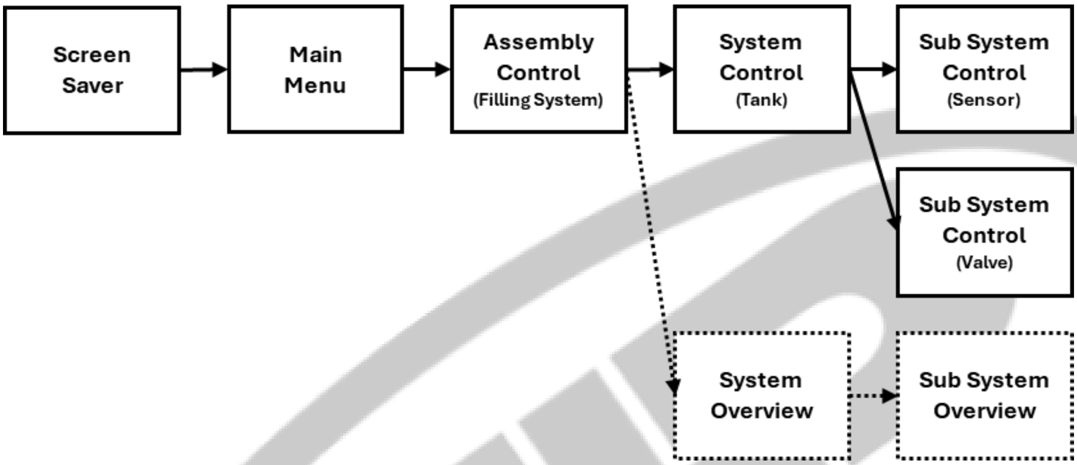


RUNG DESCRIPTION: Reset Indicator Solid Programming



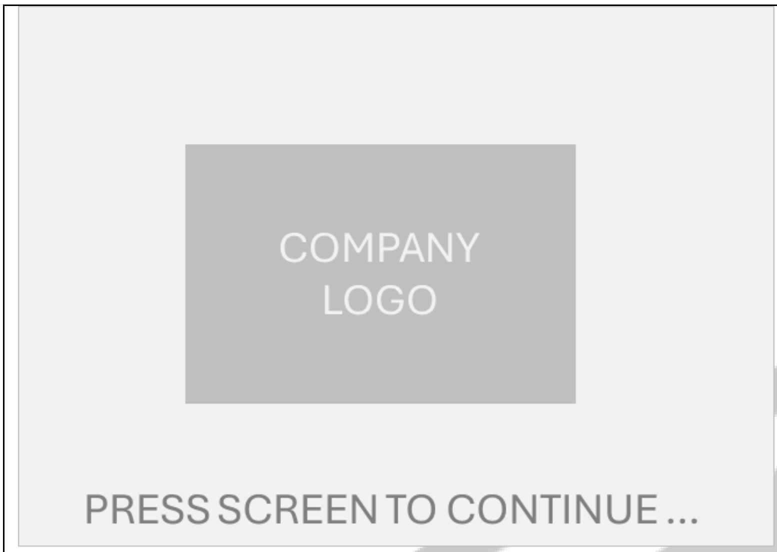
HMI Programming

The aim of the HMI Programming is to provide an intuitive means of operation for operators and technicians. The System is programmed on the basis of controlling Sequences. When entering the Control Page of each Assembly / System, the operator will be able to START and STOP the Sequence. In the event that a sequence is not clear – a HELP screen can be provided detailing what conditions are expected and what should be occurring.

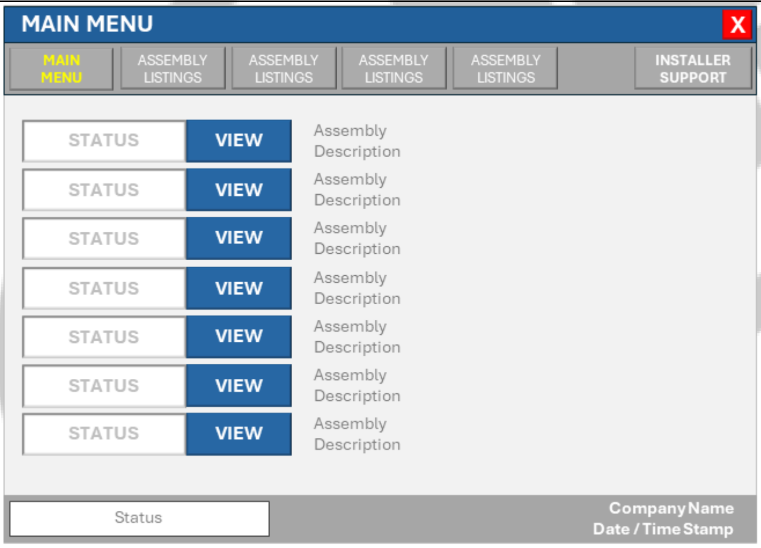

Screen Saver
Main Menu
Assembly Control (Filling System)
System Control (Tank)
Sub System Control (Sensor)
Sub System Control (Valve)
System Overview
Sub System Overview

HMI's should be programmed in a way which is both functional and is intuitive. The aim of this standard is to design a HMI's interface which will allow operators to easily Start / Stop / Reset systems as well as allow maintenance staff to manually control and monitor Sub-Systems.

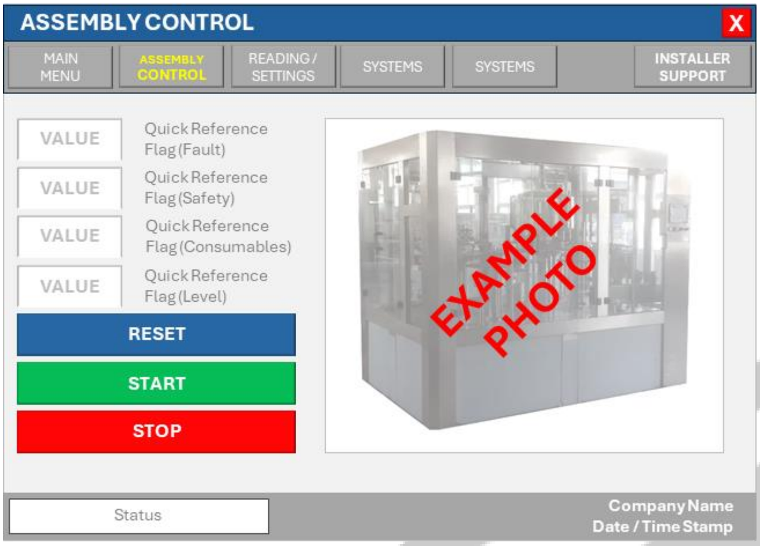
Screen Saver Screen

	**Details:** The following screen is a basic screen saver which has the logo of the company that owns the equipment. An invisible button component should be placed over the screen so that a user just needs to press the screen (anywhere) to progress to the next screen.

Main Menu Screen

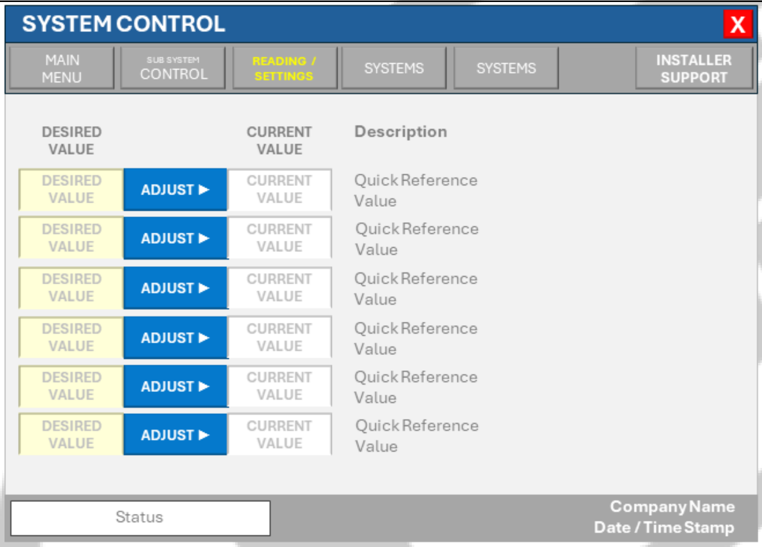
 MAIN MENU X MAIN MENU ASSEMBLY LISTINGS ASSEMBLY LISTINGS ASSEMBLY LISTINGS ASSEMBLY LISTINGS INSTALLER SUPPORT STATUS VIEW Assembly Description STATUS VIEW Assembly Description STATUS VIEW Assembly Description STATUS VIEW Assembly Description STATUS VIEW Assembly Description STATUS VIEW Assembly Description STATUS VIEW Assembly Description Status Company Name Date / Time Stamp	**Details**: The following screen is for accessing all the difference Assembly’s which the HMI is programmed to communicate with.

Assembly Control Screen



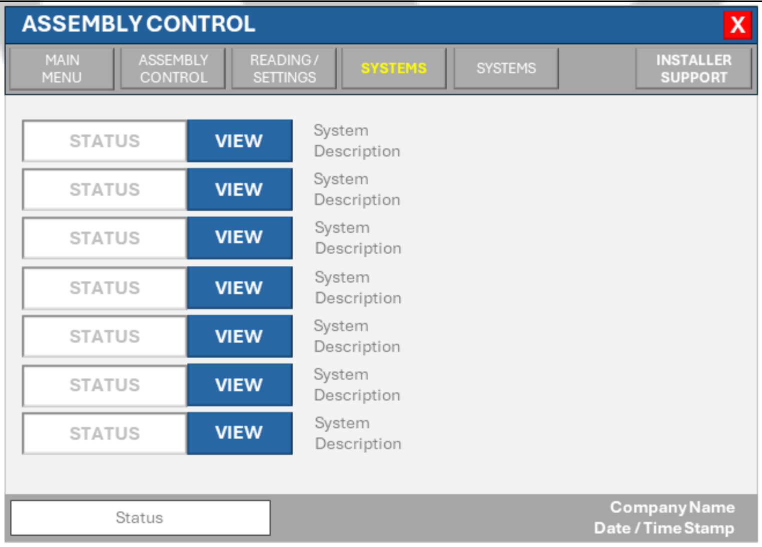
Details: The following screen provide quick reference values / flags which are typically required for the operators to monitor (Bins Full / Safety Tripped / Duct Blocked etc). This should have a basic Start / Stop / Reset control at the base of the page.

Access Level: Operators



Details: The following screen allows for Assembly's set points to be programmed. An example of this could be the through-put counter which switches the Assembly in to Standby after a given number of cycles.

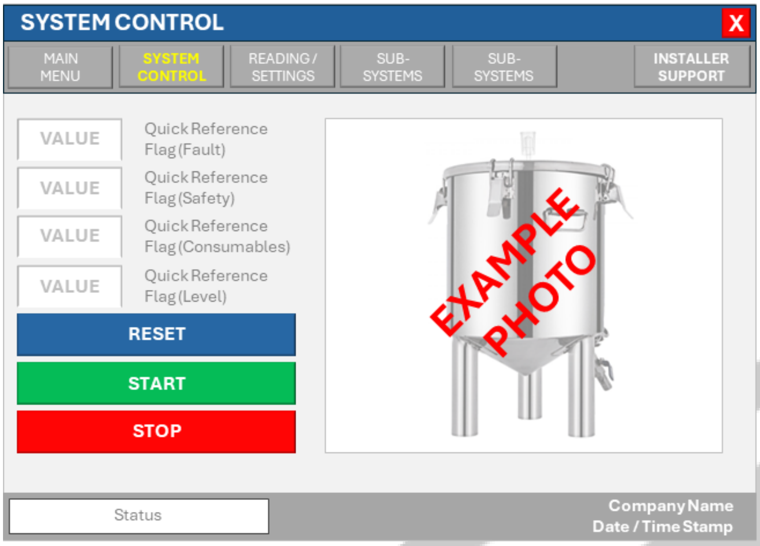
Access Level: Maintenance



Details: The following screen show System pages (these are those processes like Tank Level Management, which are connected to Sub Systems like level sensors and pumps). This page should also provide information regarding the status of every System.

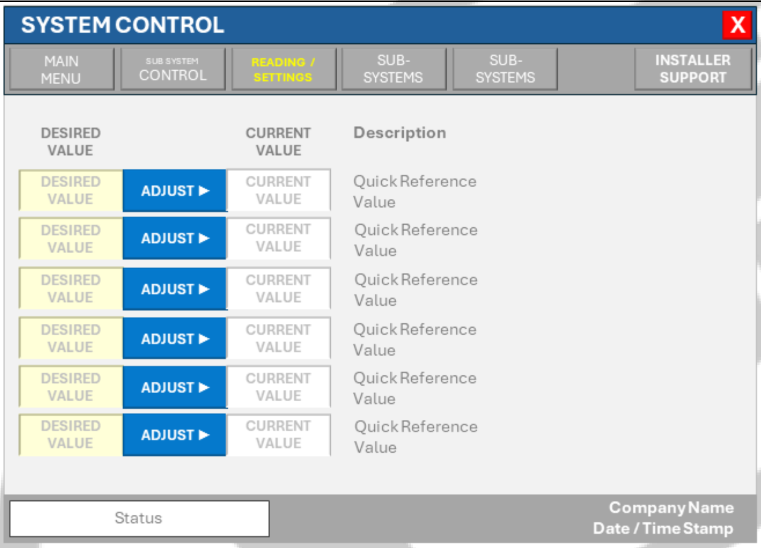
Access Level: Operators

System Control Screen



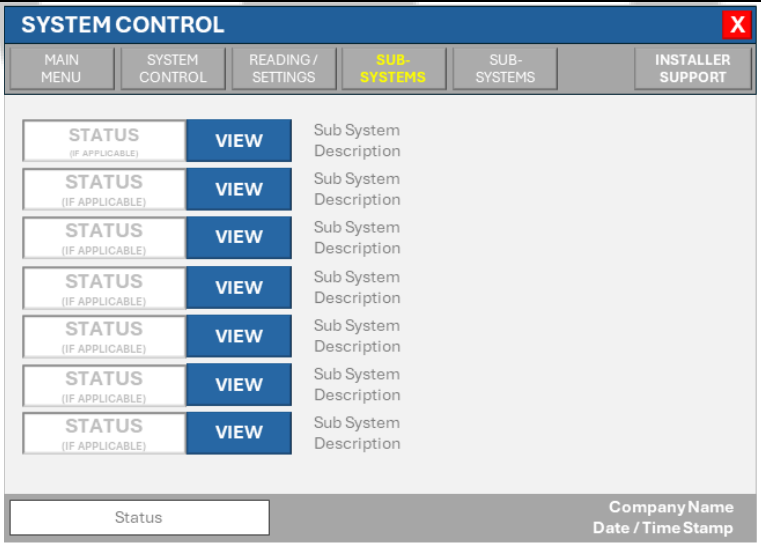
Details: The following screen provides quick reference values / flags which are typically required for the operators to monitor. This should have a basic Start / Stop / Reset control at the base of the page.

Access Level: Operators



Details: The following screen allows for System set points to be programmed into the system. An example of this could be the upper and lower levels of a storage water tank.

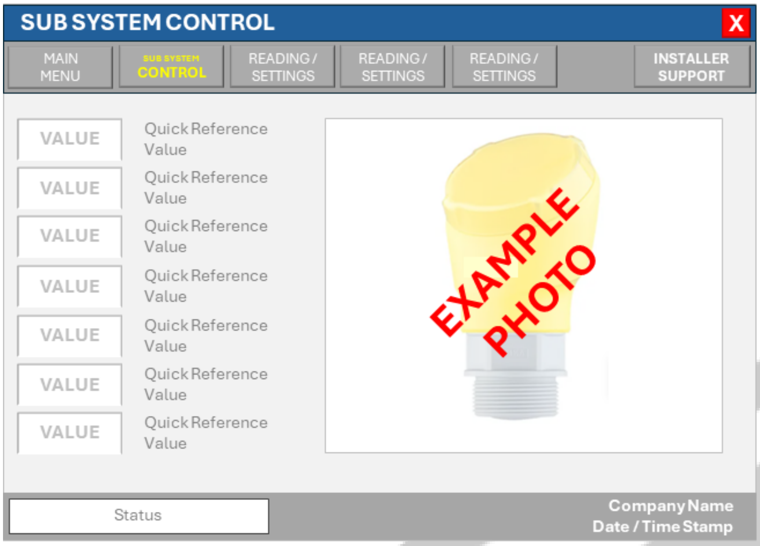
Access Level: Maintenance



Details: The following screen shows Sub-System pages. These are typically the individual components (for example sensors, VSDs, etc)

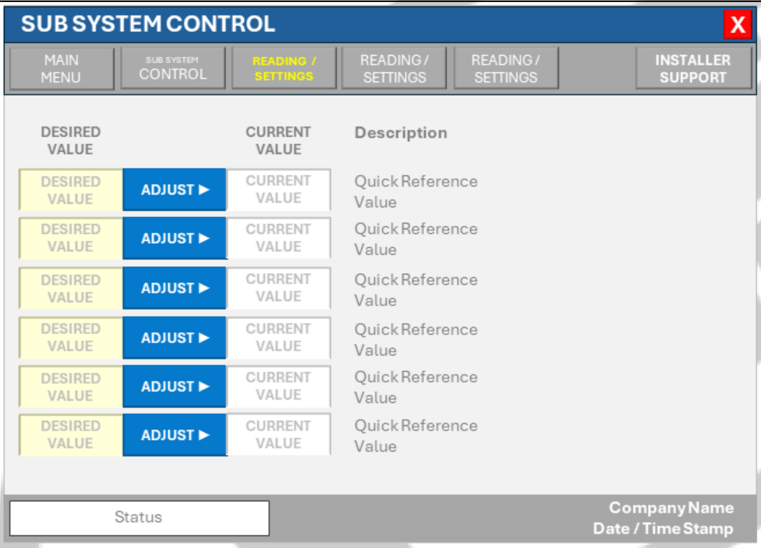
Access Level: Operators

Sub System Control Screen



Details: The following screen provide quick reference values / flags which are typically required for the operators to monitor. There should be **no** Start / Stop / Reset function at this level.

Access Level: Operators

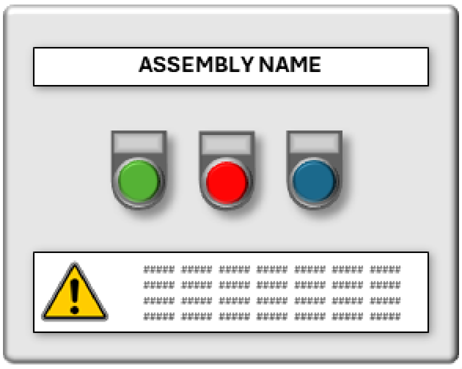


Details: This screen should provide feedback and settings which are programmable by the sensor / device connected. An example of this maybe a current reading of a sensor or the adjustment of a muting distance of a sensor.

Access Level: Maintenance

Push Button Configuration

Control Console



The following configuration is recommended for Push Button control. This configuration can be installed on a dedicated System (focusing on a single process like monitoring the liquid level of a tank) or for the entire Assembly.

To make the system intuitive and easy for upgrading, the button configuration matches what is recommended within an HMI.

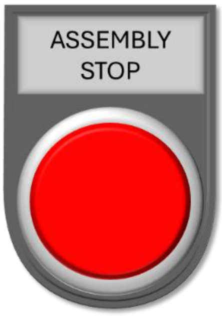
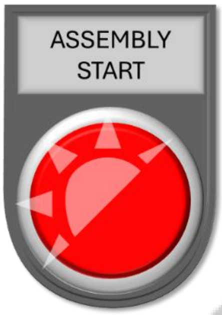
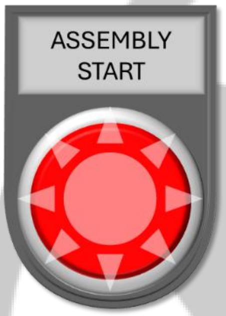
This can be installed on the control board, which houses the PLC controlling the Assembly, or on its own operator panel.

It is highly recommended that any controls, like that pictured above, are installed in a location that is in viewing distance of the machinery in which it controls.

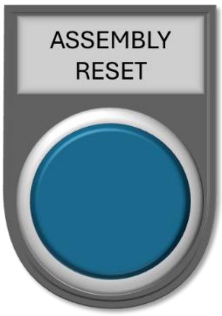
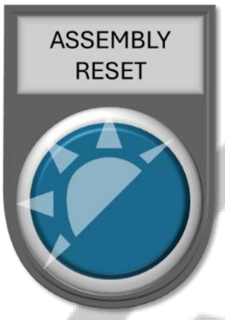
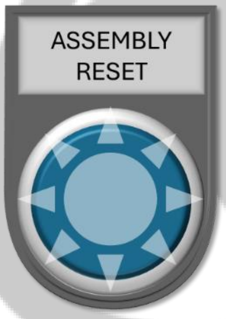
Assembly / System Start Button / Indicator

ASSEMBLY START	**Assembly Start Button / Indicator OFF** Presented when the Assembly / System is Stopped and not ready for Start-up (in fault / emergency condition). Presented when the Assembly / System is in one of the following stage(s); • Warm-down • Stopping • Stopped • Fault • Emergency
ASSEMBLY START	**Assembly Start Button / Indicator FLASHING (1Hz)** Presented when the Assembly / System is ready for Starting. Indicator flashing in attempt to gain the operators attention. Presented when the Assembly / System is in one of the following stage(s); • Standby
ASSEMBLY START	**Assembly Start Button / Indicator SOLID ON** Presented when the Assembly / System is Running. Presented when the Assembly / System is in one of the following stage(s); • Start-up • Started • Warm-up • Running

Assembly / System Stop Button / Indicator

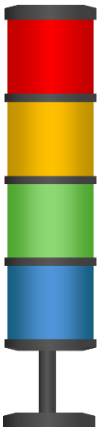
	**Assembly Stop Button / Indicator OFF** Presented when the Assembly / System is Started. Presented when the Assembly / System is in one of the following stage(s); • Start-up • Started • Warm-up
	**Assembly Stop Button / Indicator FLASHING (1Hz)** Presented when the Assembly / System is ready for Shut-down. Indicator flashing in attempt to gain the operators attention. Presented when the Assembly / System is in one of the following stage(s); • Start-up • Started • Warm-up • Running
	**Assembly Stop Button / Indicator SOLID ON** Presented when the Assembly / System is Stopped / Standby. Presented when the Assembly / System is in one of the following stage(s); • Warm-up Down • Stopping • Stopped • Standby • Fault • Emergency

Assembly / System Reset Button / Indicator

	**Assembly Reset Button / Indicator OFF** Present when there are no faults present in the Assembly / System. Presented when the Assembly / System is in one of the following stage(s); • Standby • Starting • Started • Warm-up • Running (All) • Warm-down • Stopping • Stopped • Standby
	**Assembly Reset Button / Indicator FLASHING (1Hz)** Present when there is a Resettable fault within the Assembly / System. Presented when the Assembly / System is in one of the following stage(s); • Fault
	**Assembly Reset Button / Indicator SOLID ON** Present when there is a Non-Resettable fault (from the perspective that the Safety System is independent to the Control PLC) within the Assembly / System. Presented when the Assembly / System is in one of the following stage(s); • Emergency

Indicator Stat Configuration

Stack Configuration



The light stack is recommended to be installed in the following manner. Given that the Red Indicator is typically setup as a Emergency Type indicator which requires immediate attention – it is held to the highest vertical position. Subsequently, the next prioritised stage is that of any maintenance issues represented by the Orange Indicator. Both of these will shut the Assembly / System down if they were to occur.

The next step down if the Green Indicator which indicates that the system is Running. This allows operators and Team Leaders to clearly identify that systems are within acceptable ranges from a distance. The last indicator being Blue, is typically used for operator invention (for example re-stocking consumables).

It is recommended that the urgency of the indicator be made from most urgent and critical being at the top – to least critical being at the bottom.

Assembly Stack Red Indicator

	Assembly Red Indicator Stack OFF Used to indicate that there are no Emergency Triggers, and that the system is in a work safe state. Presented when the Assembly / System is in one of the following stage(s); - All other stages
	Assembly Red Indicator Stack FLASHING (1Hz) Used to indicate that an Emergency Trigger has occurred and requires immediate attention. Presented when the Assembly / System is in one of the following stage(s); - Emergency
	Assembly Red Indicator Stack SOLID ON Not used.

Assembly Stack Orange Indicator

	Assembly Orange Indicator Stack OFF Used to indicate that there are no Faults present and that the system is in a serviceable state. Presented when the Assembly / System is in one of the following stage(s); - …
	Assembly Orange Indicator Stack FLASHING (1Hz) Used to indicate that a Fault has occurred and requires attention. Presented when the Assembly / System is in one of the following stage(s); - System Fault
	Assembly Orange Indicator Stack SOLID ON Not used.

Assembly Stack Green Indicator

	Assembly Green Indicator Stack OFF Used to indicate that the Assembly is no Running. Presented when the Assembly / System is in one of the following stage(s); - All other stages
	Assembly Green Indicator Stack FLASHING (1Hz) Used to indicate that the Assembly / System is in the process of either starting or stopping. Typically used in a change of operational state. Presented when the Assembly / System is in one of the following stage(s); - Starting - Started - Warm-up - Warm-down - Stopping
	Assembly Green Indicator Stack SOLID ON … Presented when the Assembly / System is in one of the following stage(s); - Running (All)

Assembly Stack Blue Indicator

	Assembly Blue Indicator Stack OFF Typically used in indicating that no operator attention is required. This indicator in this state indicates that there are no operate type interventions required.
	Assembly Blue Indicator Stack FLASHING (1Hz) Typically used to indicate that the Assembly / System required operator attention. This is used to indicate that a operate duty is due shortly.
	Assembly Blue Indicator Stack SOLID ON Typically used to indicate that the Assembly / System required immediate operator attention. This is used to indicate that a operate duty is required before starting the Assembly / System again.