L19 - Develop Advanced Safety Applications Using the Integrated Safety Controller Platform
Get Familiar with the Demo Cases

DEMO-L8PERF1

DEMO-K570021

DEMO-TUNING460
Safety Controller

GuardLogix® 5570 Controller
1756-L73S

Compact GuardLogix® 5370 Controller
1769-L36ERMS
GuardLogix® is a high performance, multi-discipline controller that provides the full capabilities of the popular ControlLogix™ Controller, but also includes integrated safety capability:

- SIL3, PLe, CAT 4 safety control
- **Standard, safety** and motion control in the same controller 🌟
- CIP Safety over EtherNet/IP
- Certified Safety Instructions
- High Integrity Add-On Instruction
Safety Inputs and Outputs

POINT Guard Safety I/O Modules™

• 1734-IB8S
• 1734-OB8S
The drive that spins the motors

Kinetix® 5700 Drives

Network based STO using CIP safety protocol
Safety Function to mitigate risk

- Safety Function PLr = PLd
  - Sensor
    - Locking Gate Switch (TLS3-GD2 Power to Release)
    - Wired to 1734-IB8S PointGUARD Safety Input Module
  - Logic Device
    - GuardLogix 5570 in this lab
      - GuardLogix 5580 (future)
        - PLd / SIL CL 2 (primary only)
        - PLe / SIL CL 3 (primary and partner)
  - Actuator
    - Kinetix 5700 STO
Safety Function Application Technique

- SAFETY-AT135B
- Logic: GuardLogix 5370 or 5570
- Actuator: Kinetix 5500 or 5700 with integrated STO
Safety Function Application Technique

- Functional description (for use in SRD)
- BOM
- Wiring diagram
- Program/Configuration
- Sistema Calculation
- Verification & Validation checklists
Sistema Calculation for Safety Function

- Safety Function
  - Sensor
    - Locking Gate Switch (TLS3-GD2 Power to Release)
    - Wired to 1734-IB8S PointGUARD Safety input module
  - Logic Device
    - GuardLogix 5570
  - Actuator
    - Kinetix 5700 STO

![Safety function diagram](image)

<table>
<thead>
<tr>
<th>Status</th>
<th>Name</th>
<th>PL</th>
<th>PL-Software</th>
<th>PFHD (1/h)</th>
<th>CCF score</th>
<th>MTTFD [h]</th>
<th>Category</th>
<th>Requirements of the category</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️ SB</td>
<td>Locking Gate Switch</td>
<td>c</td>
<td>n.a.</td>
<td>9.1E-10</td>
<td>65</td>
<td>2500</td>
<td>4</td>
<td>fulfilled</td>
</tr>
<tr>
<td>✔️ SB</td>
<td>POINT Guard I/O: 1734-E6S</td>
<td>c</td>
<td>c</td>
<td>5.1E-10</td>
<td>not relevant</td>
<td>not relevant</td>
<td>4</td>
<td>fulfilled</td>
</tr>
<tr>
<td>✔️ SB</td>
<td>Safety PLC: GuardLogix 1756-L7x3 &amp; L7SP</td>
<td>c</td>
<td>c</td>
<td>1.2E-9</td>
<td>not relevant</td>
<td>not relevant</td>
<td>4</td>
<td>fulfilled</td>
</tr>
<tr>
<td>✔️ SB</td>
<td>Serve Drive: Kinetix 5700 Single Axis Inverter with Safe Torque Off</td>
<td>c</td>
<td>c</td>
<td>1.6E-9</td>
<td>not relevant</td>
<td>not relevant</td>
<td>3</td>
<td>fulfilled</td>
</tr>
</tbody>
</table>
How to get the Motor to Spin

1. Cycle the Emergency Stop button
2. Press blue reset button
3. If Faulted indicator on HMI is red, press [Clear Faults] on HMI
4. Press green START button on HMI to start motion
Questions so far?
<table>
<thead>
<tr>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety Task</td>
</tr>
<tr>
<td>Safety Tags</td>
</tr>
<tr>
<td>Mapping Tool</td>
</tr>
<tr>
<td>Safety Input FBs</td>
</tr>
<tr>
<td>Safety Output FBs</td>
</tr>
<tr>
<td>Diagnostics</td>
</tr>
<tr>
<td>CIP Safety</td>
</tr>
<tr>
<td>Safety Signature / Lock</td>
</tr>
</tbody>
</table>
Integrated Safety Controller

GuardLogix® Controller
1756-L73S

Controller scans
• Safety Tasks
• Standard Tasks
  • Motion
  • Control
  • HMI
  • etc…
Safety Task Lab

- Pages 15 to 17
- Two minutes for the lab
- Go directly to step 2 on page 16
  - We already did step 1 for you by opening project
Safety Task Scan

- Safety Task is a Periodic Task
- At periodic rate, Primary ‘wakes up’ Partner
- Safety input data ‘frozen’; does not change during scan
- Primary sends raw input data to Partner
- Both Primary and Partner execute safety logic to completion using the frozen safety input data
- Safety Task results are cross checked by both controllers (primary ↔ partner)
- If results are OK, Primary and Partner each build half of the safety output data packet, and packet is transmitted to the safety outputs
Safety Tag Versus Standard Tag

- What is the difference between safety tags and standard tags?
  - Safety tags are all generated 1oo2
    - Generated by Safety I/O and Safety task
    - Controlled by Safety task
  - Standard tags are all generated 1oo1
    - Generated by Standard I/O and Standard task(s)
    - Controlled by Standard task(s)
Safety Tag Lab

- Pages 18 and 19
- Three minutes for the lab
Safety and Standard Tag Summary

- Every controller tag has a class; either ‘standard’ or ‘safety’
  - Rule #1 Cannot use standard tags in safety task
  - Rule #2 Cannot ‘drive’ or ‘control’ a safety output tag in standard task
Question

- Can I wire a reset button to a standard input to reset my safety functions?
Question

- Can I wire a reset button to a **standard** input to reset my safety functions?

- Do the Mapping Tool Lab on pages 20 and 21 to find out the answer
  - 2 minutes to do the lab
Old Rule #1 Cannot directly use standard tags in safety task

New Rule #1 Can use standard tags in safety task, via mapping tool

Rule #2 Cannot ‘drive’ or ‘control’ a safety output tag in standard task
Safety Input Instructions
All the Certified Safety Input Instructions are Dual Channel

- Dual Channel instructions help confirm both channels are within tolerance
- If they remain out of tolerance for longer than the discrepancy time, a fault is declared
DCS is the base safety input instruction
DCS Instruction

- Input Type
  - Equivalent or Complementary
- Discrepancy Time
  - How long can the inputs be diverse before a fault is declared
- Restart Type
  - Is 'Reset' required to set O1 HI?
    - AUTOMATIC – NO
    - MANUAL - YES
- Cold Start Type
  - Is demand / cycle required on power-up?
    - AUTOMATIC – NO
    - MANUAL - YES
- Input Status
  - Is input channel data valid?
    - If status LO, output O1 is de-energized
- Reset
  - Reset faults (FP)
  - Restarts O1 if configured for manual ‘Restart Type’

Dual Channel Input Stop

```
<table>
<thead>
<tr>
<th>DCS</th>
<th>CMSS_EStop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety</td>
<td>EMERGENCY STOP</td>
</tr>
<tr>
<td>Input Type</td>
<td>EQUIVALENT - ACTIVE HIGH</td>
</tr>
<tr>
<td>Discrepancy Time (Msec)</td>
<td>3000</td>
</tr>
<tr>
<td>Restart Type</td>
<td>AUTOMATIC</td>
</tr>
<tr>
<td>Cold Start Type</td>
<td>AUTOMATIC</td>
</tr>
<tr>
<td>Channel A</td>
<td>CMSS_EStop_ChannelA</td>
</tr>
<tr>
<td></td>
<td>Point:IO:21:Pt02Data</td>
</tr>
<tr>
<td>Channel B</td>
<td>CMSS_EStop_ChannelB</td>
</tr>
<tr>
<td></td>
<td>Point:IO:21:Pt03Data</td>
</tr>
<tr>
<td>Input Status</td>
<td>Point:IO:21:CombinedInputStatus</td>
</tr>
<tr>
<td>Reset</td>
<td>Fault_Reset_Safety</td>
</tr>
</tbody>
</table>
```
• Pages 22 – 27
• Four minutes to do the lab
Why did the device have to be cycled?

- Although you know the fault was caused by a wire off, the instruction thinks that a much more dangerous fault occurred.

- It thinks the high channel was stuck HI, and that the LO channel operated properly.

- Therefore, you have to cycle the device to prove no channels are stuck HI.
What instructions would be used for single channel safety circuits?
What instructions would be used for single channel safety circuits?

- Boolean instructions, such as:
  - XIC

- Note even the Booleans instructions have been certified by the TUV
  - That is why they have the red triangle
  - That is why they are available in the safety task
Safety Output Instructions
Output Function Blocks

- The CROUT is the **only** updated instruction for safety output devices
Safety Output Instruction / CROOUT

- Dual Channel with feedback
CROUT Instruction

- Feedback Type
  - Positive or Negative
- Reaction Time
  - How long to wait for feedback to follow outputs before a fault is declared
- Actuate
  - No restart function
  - Outputs O1 and O2 simply follow actuate if no faults
- Input and Output Status (embedded interlocks)
  - Is feedback data valid?
  - Are output channels being driven by CROUT fault free?
- Reset
  - Reset feedback faults (FP)

Configurable Redundant Output

<table>
<thead>
<tr>
<th>Configurable Redundant Output</th>
<th>CROUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feedback Type</td>
<td>POSITIVE</td>
</tr>
<tr>
<td>Feedback Reaction Time (Msec)</td>
<td>250</td>
</tr>
<tr>
<td>Actuate</td>
<td>0MSS_Estop_OutputEnable1</td>
</tr>
</tbody>
</table>

Feedback 1: Point_J0:1.1.P00Data
Feedback 2: Point_J0:1.1.P01Data
Input Status: Point_J0:1.1.CombinedInputStatus
Output Status: Point_J0:3.1.CombinedOutputStatus
Reset: Fault_Reset_Safety
CROUT Lab

- Pages 28 – 32
- Four minutes to do the lab
Typical Safety Interlock Code

- Safety Inputs OK
- Fault
- Feedback
- Redundant Outputs
- Reset button
What instruction used for **Safety Outputs** without feedback?
What instruction used for Safety Outputs without feedback?

- Just use a Boolean instruction, such as:
  - OTE

- Some drives no longer provide electromechanical feedback for the STO outputs; including the new CIP Safety based STO on the Kinetix®5500 Drives
  - Safety solutions are moving from electromechanical (known failure modes) safety devices to complex (unknown failure modes) safety devices
  - The standards are adapting by moving from deterministic to performance based criteria; i.e.; what is the probability of a failure
Programming Required

- Does your safety application code have to be ‘certified’?
Does your safety application code have to be ‘certified’? Not certified
Does your safety application code have to be ‘certified’? Not certified

- Your application code must be ‘validated’
- You can do that yourself
- If you wish to have your code validated by a third party, they will spend much more time checking your software processes than checking your code.
How do I develop and validate safety code?

- Follow the V-Model (IEC 61508-3 and ISO 13849-1)
- If you wish to have your code validated by a third party, they will spend much more time checking your software processes than checking your code.
  - Documented developmental lifecycle according to V-model
    - example: Do you have a SRASW (safety related application software) specification?
  - Modular and Structured Programming
  - Code shall be readable, understandable, and testable
Dangerous Undetected (DU) fault

- The highlighted fault is undetectable with this wiring; sourced from power rail
- The channel-to-channel short is not detected, and thus a second fault could lead to a loss of the safety function.
Pulse testing

- The channel-channel short can be detected using a pulse test.
- Safety Controllers and Safety I/O have configurable pulse test capability.
- Test Outputs are used to source the 24V for the channel, and are intermittently pulse tested to detect shorts to 24V.
Diagnostics Lab

- Pages 33 – 43
- Eight minutes to do the lab
- Skip steps 3 thru 8
How to Configure Safety Input Channels

- Channel Configuration / Single or Equivalent?
Where to Detect Discrepancy Faults

Hardware Detection

Equivalent

Software Detection

Discrepancy Time

chA
HI
LO

chB
HI
LO

Single
Safety Networks

Communication Errors
- Message Repetition
- Message Loss
- Message Insertion
- Incorrect Sequence
- Message Corruption
- Message Delayed

CIP Safety
- Data sent Twice
  - Once inverted
- Separate Data CRCs
  - Overall CRC
- Messages Timestamped
- CRTL
  - Connection Reaction Time limit

- Diagnostic capabilities of CIP Safety make the network path irrelevant from a safety aspect
CIP Safety Lab

- Pages 44 – 46
- Three minutes to do the lab
Determined by RPI / Timeout Multiplier / Network Delay Multiplier

- RPI = 10
- \(2 \times 10 = 20\)
- \(\frac{200}{100} \times 10 = 20\)

CRTL = 40
- Recommendation is to use the default CRTL of x4 RPI
  - Good compromise between a shorter safety reaction time and avoidance of nuisance trips
  - CRTL, not RPI, must be used in safe distance and safe reaction time calculations
Protection from Unwanted Change
Protection from Unwanted Change

- Safety systems need to protect against
  - Offline edits to the safety program
  - Online changes to the safety program
  - Parameter changes from HMIs
  - Program downloads that overwrite the safety program

- Malicious?
- Inadvertent?
Safety Signature / Lock Lab

- Pages 47 – 49 for safety signature lab
- Pages 50 - 56 for safety lock lab
- Six minutes to do the lab
Protection from Unwanted Change

- **GuardLogix® Controller** uses signature and lock
  - Offline edits to the safety program
    - Safety **Signature** or Safety Lock
  - Online changes to the safety program
    - Safety **Signature** or Safety Lock
  - Parameter changes from HMIs
    - Safety **Signature**
  - Program downloads that overwrite the safety program
    - Safety Lock
Safety Signature

- **With a signature in place**
  - The program in the safety controller has been **validated**
  - Online edits cannot be made to the safety task
  - Offline edits cannot be made to the safety task
  - Forcing of safety I/O is prohibited
  - External devices, such as HMIs or the standard portion of cGLX, are prohibited from writing into safety memory on the cGLX controller
  - Background memory check between the primary and partner is begun
  - SAFETY RUN status indicator on controller goes solid green

- **Note the partner always runs the safety task, even without a signature**
What prevents inadvertently downloading a project with a different safety task?

Safety Lock

To download to this controller you must:

Unlock and proceed with the download.
Questions ?
Safety Automation System of Rockwell Automation
Networked Functional Safety System
Thank You!