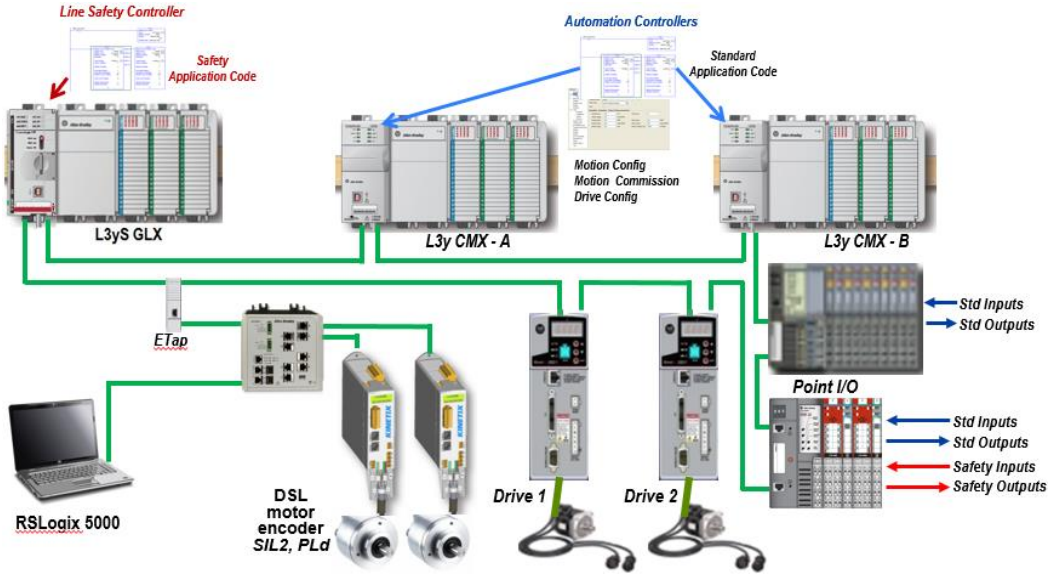
Safety Parameter Data

Kevin Colloton
Technology Manager - Safety
Rockwell Automation

Presented at the ODVA
2017 Industry Conference & 18th Annual Meeting
February 21-23, 2017
Palm Harbor, Florida, USA

Abstract

A functional gap in CIP Safety is being able to change parametric data in a safety I/O class device (or get data from a safety device) on-demand (when the need arises) based on safety program execution. Today, parametric data can only be sent when the device CIP Safety connection is opened, or by embedding parametric data as part of a safety output data assembly. Sending parametric data in the Forward_Open configuration block is not suitable for program driven parametric data exchange with a safety device. In fact, for a safety device, the configuration block of the Safety Open is part of the Safety Signature and cannot be modified programmatically without violating the devices safety signature.
Using the Output Assembly for on-demand parameter data transfer is both wasteful and a misuse of I/O connection's intent, not to mention that the number and complexity of the resulting assembly structures would explode.
With no effective mechanism for on-demand programmatic data exchange, CIP Safety systems limit the flexibility customers have for advanced safety applications, especially for motion safety related functions, and looking forward for high availability (SIS) safety systems.



This document identifies a functional gap in CIP Safety, being able to change parametric data in a safety I/O class device (servos, drives, etc), or get safety data from the device on-demand (when the need arises) based on program execution.
Today CIP Safety is limited to two types of data exchange between the owner (safety controller) and the target safety device. Parametric data can only be sent when the device connection is opened, or by embedding parametric data as part of the output data assembly. Sending parametric data in the Forward_Open configuration block is not suitable for program driven parametric data exchange with the device. In fact, for a safety device, the configuration block of the Safety Open is part of the Safety Signature and cannot be modified programmatically without violating the signature. Using the Output Assembly for on-demand parameter data transfer is both wasteful and a misuse of I/O Connection's intent, not to mention that the number and complexity of the resulting assembly structures would explode. With no effective mechanism for on-demand programmatic data exchange, the user must resort to message instructions for standard I/O class devices, but is left with no solution for safety I/O class devices. (safety messaging, including explicit safety messaging,is not supported), so we have no CIP Safety capable message other than I/O connections (class 1), which doesn't provide a solution for this problem. Clearly, there is a need for a new on-demand SIL 3 capable data transfer mechanism. This paper highlights the problem, and suggests requirements to be considered for future solutions.

1.2 Objectives

Identify a method that allows CIP Safety I/O device safety related parameters to be changed or read during runtime by the safety application.

1.3 References

[2] The CIP Networks Library Volume 5, CIP Safety Edition 2.11 April 2015

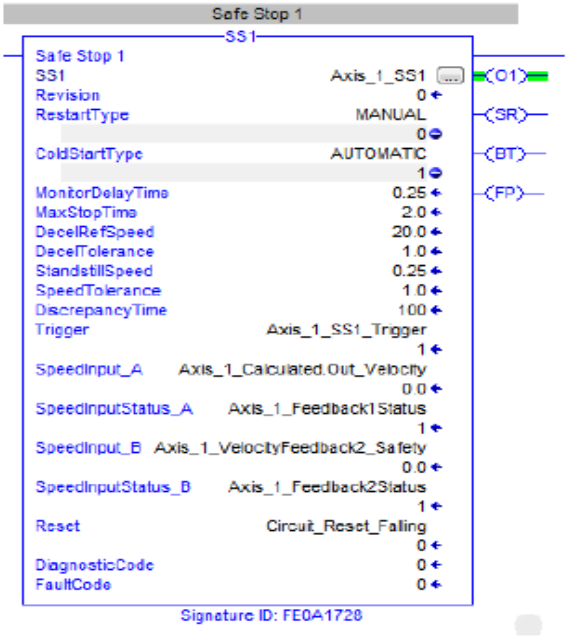
2. Use cases

For use in this document, parametric data that can be changed by the controller during runtime is referred to as Class 3 Parameters. Device data that can be read by the controller during runtime is referred to as "C3D", i.e. Class 3 Data.

2.1. Motion Safety Functions

Motion Safety Functions can execute in the controllers safety task as safety instructions, in the target motion device, or in both (safety task + target device).
We will use the SS1 (Safe Stop 1) function executing in the drive as an example for this use case, but consider that there are other motion safety functions (instructions) that are typical in customer applications. Also consider that a device may have multiple instructions that the user implements (pointing out that one device does not always equal one instruction).
In the illustration, the primary SS1 safety parameters are:

1. Monitor Delay Time
2. Max Stop Time
3. Decel Ref Speed
4. Decel Tolerance
5. Stand Still Speed
6. Speed Tolerance
7. Discrepancy Time



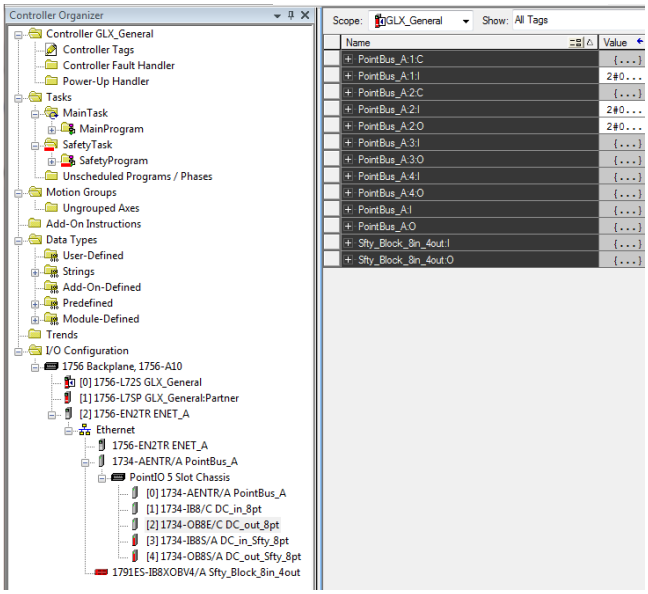
Although the need to change any of these parameters during runtime exists (based on user application requirements), the most likely needing to change are:

1. Monitor Delay Time
2. Max Stop Time
3. Decel Ref Speed

When the user safety application determines a C3P change is needed, typically it's related to machine speed, product/production variables, maintenance requirements (speed, inertia, activity, etc.). With that stated, a typical C3P change could be anytime, daily, weekly, or monthly. The point is we really don't know, however we do know that sending C3D data as part of the output assembly is clearly an unworkable solution.
Generally, the C3P data that is sent does not have the time criticality of periodic I/O connection data. In evaluating mechanisms to pass C3D, it is unacceptable to break the I/O connection (which for safety effectively initiates a safety demand, which will result in a safe state condition - machine off). Technically this is a loss of availability (it's not an actual safety demand), which typically results in major customer dissatisfaction.

2.2 C3P tags are defined by the safety device profile

It is the responsibility of safety device designers to identify parameters that have runtime variability (read or read / write). Being an optional feature, it's up to the device vendor to identify which, if any, parameters can be modified during runtime.



2.3 C3P tags can be read only, or read / write

Parameter tags can be read only or read / write. Device developers determine which tags are read only, and which tags are read / write for their specific devices.

2.4. Writing C3P tag values

It is required that multiple parameter tag members in a device can be changed in a single update event initiated by the application program.

Where multiple devices are part of an application, each device is handled individually (user application responsibility). It's up to the application developer to properly manage when messaging is performed.

3.1. Process Safety Systems

The initial development and use of CIP Safety has primarily been in machine safety applications IEC 61508, IEC 62061 (high demand systems). Although the safety ratings and functionality of CIP Safety allows its use in process safety applications IEC 61508 and IEC 61151 (low demand systems), CIP Safety based systems in process has been restricted due to a lack of system availability (redundancy for availability).

Process safety applications typically have significant high availability requirements due to nuisance / spurious trip avoidance. The ability to change safety parameters during runtime would be a high value feature for CIP Safety use in process safety applications as CIP Safety adoption expands.

****