SafeMotion as a system independent solution

Introduction

The introduction of safety-approved networks makes the control of drives and motors more and more a software item: hard-wired systems are replaced by software commands over networked systems. This is also pushed by initiatives like Industry 4.0 to create more flexibility preferably at the same quality and price levels. However, the different networks come with different solutions, which create problems at the users especially in production environments with heterogeneous networks. To harmonize this, PLCopen started a working group on SafeMotion, which created a generic proposal to solve the motion control safety aspects over the different networks like ProfiSafe, Safety over Ethercat, CIP Safety over Sercos, OpenSafety, CC-Link IE and Mechatrolink, as well as user area's as described in OMAC.

The Changing Motion Control environment

The emergence of (fast) digital networks made it possible to link many motors to a controller. In many cases servo technology replaces the large single motor solution, making the functionalities locally available in the machine and also replacing the previously mechanical solutions by software control. This solution is also referred to as "mechatronic solutions".

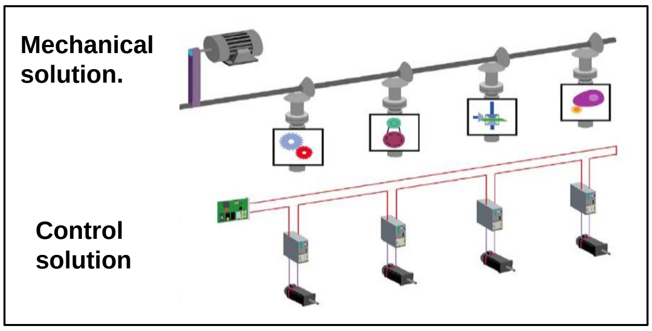

Figure 1: From mechanical to mechatronics

Usage of multiple motors changes the safety connections: from a hardwired safety solution to a solution over a safety network with a dedicated safety controller added to the functional controller (PLC).

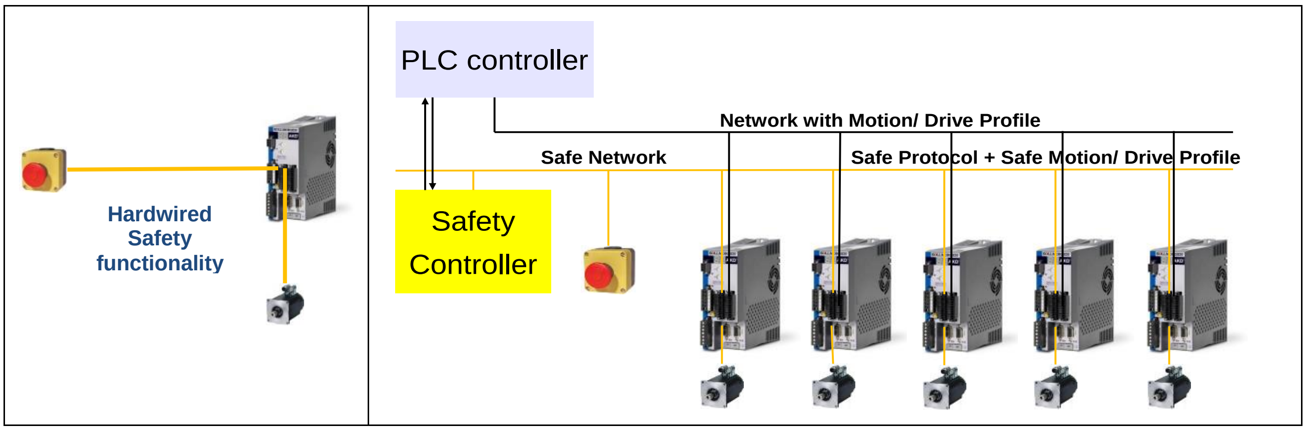
Hardwired
Safety
functionality
PLC controller
Network with Motion/ Drive Profile
Safe Network
Safe Protocol + Safe Motion/ Drive Profile
Safety
Controller

Figure 2: From hardwired to networked safety functionalities

The next step is to integrate the transfer of the safety data in the standard network, along with the logic and motion data. As a result only one network is necessary for the whole machine, integrating logic, motion and safety. Nowadays this kind of network types is widely available on the market.
The next logical step is to integrate the controllers in one platform too. In this way the development environment for safety and non-safety can be integrated in one common system, which creates the possibility to include the safety functionality in the project from the start.

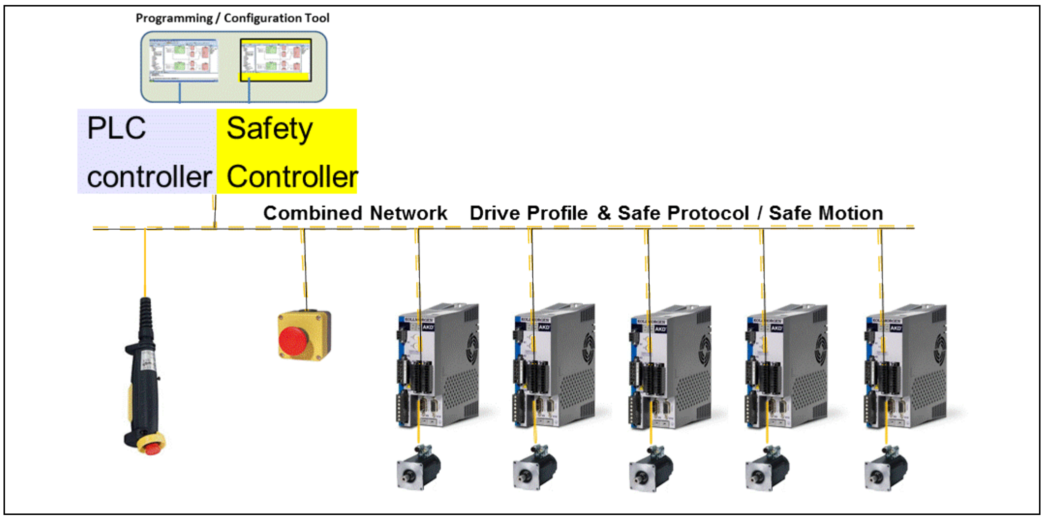
Programming / Configuration Tool
PLC
controller
Safety
Controller
Combined Network
Drive Profile & Safe Protocol / Safe Motion

Figure 3: Combined logic motion and safety networks with integrated controls

Looking to the future it will be able to transfer the information on movements per axis over the safety protocols too, and use safe motion sensors which can provide accurate safe positions in a digital form. In that way it will be possible to calculate the overall behavior of a machine. For instance the movements of the working piece in a robot arm can be limited to a safe speed.
In order to support these trends, it is necessary to reflect the safety functionality in the software environment. This paper identifies user friendly solutions for this.
Note that a “safety drive” is not equal to “safe drive [system]” like referred to in the automotive.

Overview Motion related Safety

The following motion related safety functionalities can be identified:

Safe Stop and brake functions	
STO	Safe Torque Off
SS1	Safe Stop 1
SOS	Safe Operating Stop
SS2	Safe Stop 2
SBC	Safe Brake Control

Safe Motion Basics Functions	
SLS	Safety Limited Speed
SS1	Safety Stop 1. maintaining power and after certain time delay OFF
SS2	(Stop function) like Stop Cat.2 of EN60204, Controlled shut down while maintaining the supply of power, and after Standstill Can includes SOS
SLP	Safely-Limited position
…	SafeEncoder. Combine to encoders

Safe Motion Advance Functions	
SLA	Safety Limited Acceleration
SAR	Safe Acceleration range
SSR	Safe Speed Range
SLT	Safely-Limited Torque
STR	Safe Torque Range
SLI	Safely-Limited Increment
SDI	Safe Direction
SCA	Safe Cam
SSM	Safe Speed Monitor
SMT	Safe Motor Temperature

Figure 4: Grouped overview of safety functions

Mapping to the safe networks

In order to use the safety integrated networks the safety functions have to be mapped. This chapters provides an overview, however this can be incomplete and is not intended for implementations. For this purpose the applicable network standards have to be used. Also note that there is no preference or order applicable here: it just is a small overview.

What is generic in the current status of the safety integrated networks is that there are basically two bytes or words (double bytes) used for communicating the safety commands and status. However there are slight difference in the content and meaning of the words at bit level.

Bit	**OMAC**	**ProfiSafe**	**EtherCAT**	**OpenSafety**	**CIP / Sercos**	**CC-Link IE**	**MECHATROLINK**
0	STO	STO	STO	Reset	Mode	STO	STO
1	SS1	SS1	SS1	Activate	E-Stop	SS1	SS1
2	SS2	SS2	SS2	STO	Enabling	SS2	SS2
3	SOS	SOS	SOS	SBC	SMM1	SOS	SOS
4	SLS	SLS	SSR	SS1	SMM2	SSR	SSR
5	UserD	SLT	SDIp	Reserved	SMM3	SDIp	SDIp
6	UserD	SLP	SDIn	Vendor	SMM4	SDIn	SDIn
7	Error Ack	Internal	Error Ack	Vendor	SMM5	Error Ack	Error Ack

Figure 5: Overview of the Control Byte

Proposal for Safety Drives

SF_SafetyRequest for activating & monitoring drive function

In PLCopen Safety Part 1 the function block SF_SafetyRequest is defined graphically as shown in figure 6.

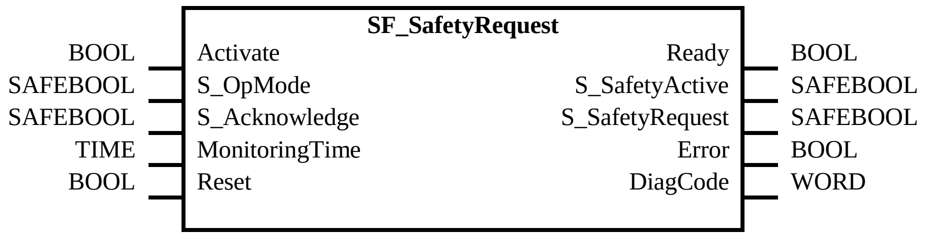

Figure 6: Graphical representation of SF_SafeRequest

The description of SF_SafeRequest provides a basic overview of the functionality is shown in picture 7.

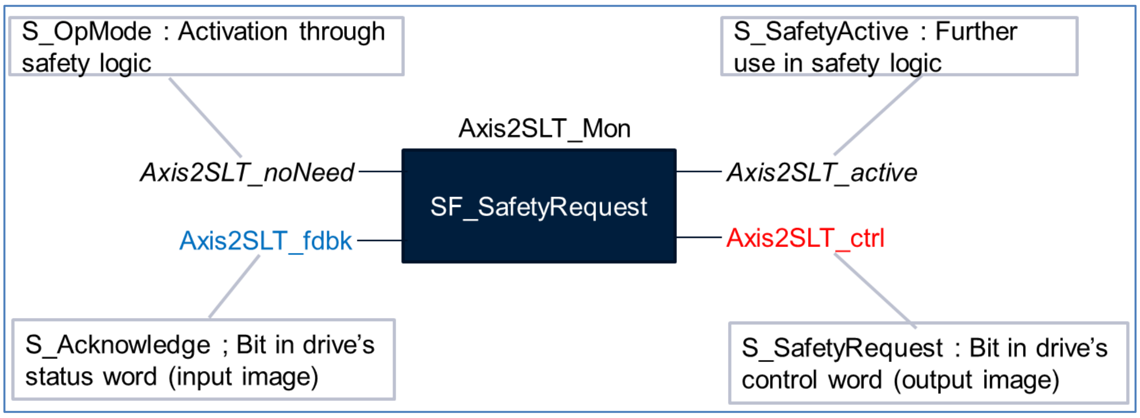
S_OpMode : Activation through safety logic
S_SafetyActive : Further use in safety logic
Axis2SLT_Mon
Axis2SLT_noNeed
SF_SafetyRequest
Axis2SLT_active
Axis2SLT_fdbk
Axis2SLT_ctrl
S_Acknowledge ; Bit in drive's status word (input image)
S_SafetyRequest : Bit in drive's control word (output image)

Figure 7: Description of SF_SafeRequest with reduced set of inputs and outputs

The main inputs and outputs of SF_SafetyRequest shall be linked as follows:

- Top-left (*Activation through safety logic*): Input **S_OpMode** is linked to the logic determining whether the safety drive function (e.g. safely limited torque) needs to become active within the time specified on input MonitoringTime (not shown). This may be the output value of a monitoring a device (e.g. monitoring a light curtain, or monitoring a mode selecting device), or some combination of conditions.
- Top right (*Further use in safety logic*): Output **S_SafetyActive** provides the feedback whether the safety drive function has become active *within the specified monitoring time* after its need has been determined (input S_OpMode). If this output is active it can be used to activate the next step, like e.g. opening a gate.
- Bottom left (*Bit in drive's status word*): Input **S_Acknowledge** reflects the status of the drive relevant to the safety drive function requested by this FB (e.g. whether the limitation of torque is currently active), and is the reflection of the process input image w.r.t. to the drive.
- Bottom right (*Bit in drive's control word*): Output **S_SafetyRequested** is the bit that goes to the drive in the safe control word for activating the function in the drive. Note that the control word can be a byte and the drive can be safe logic.

If the input on the top left is FALSE, there is no activation so output top right is FALSE. However if there is a safety request and the input top left is TRUE, the top right output reflects the status of the drive or its exceeding of the monitoring time. The bottom right bit reflects the operation by setting the

relevant bit in the control word for safety active, and after acknowledge of this in the safety drive, the bottom left input reflects this change (to TRUE).
This SF_SafetyRequest functionality can be used in a broad safety sense, incl. the safe motion functionalities. For instance, in order to use this FB for SLT (Safely Limited Torque) one combines the inputs Axis2SLT_noNeed and Axis2SLT_fdbk to generate the relevant safety outputs Axis2SLT_active and Axis2SLT_ctrl, as depicted in 0 providing the required safety functionality. Multiple instances of SF_SafetyRequest may be used to cover all safety drive functions (IEC61800, profiles, vendor-specific).

In this way most of the safety drive functionalities can be mapped easily, especially by providing a "guideline" with defined naming conventions, containing a generic scheme how to name signals related to the functions supported by a safety drive as described below. Combined with an I/O or Drive configurator these names can be generated automatically for the bits in the drive's status / control word, matching the drive's profile (symbolic names).

In this way PLCopen provides all these SafeMotion functionalities, while it covers multiple instances of the same function on the same drive, e.g. 2 x SLS with different velocities. And it allows the application to distinguish the activation of SS1 from the activation of STO, and also the activation of SS2 from the activation SOS, in drives that supports these functions separately.
Also for the supplier the certification process gets much simpler, while the user gets a consistent user interface that supports all needed functionalities, and even can group the same functionalities easily to provide a better overview of the safety application program.

The Safety Drive Naming Scheme
In order to have a consistent interface, the following naming scheme is proposed. For every safety drive *d* and every safety drive function *f* supported by the drive according to its profile and configuration, the corresponding FB instance and I/O signals shall be given the following names:

	if *d* has one instance of *f*	if *d* has multiple instances of *f*
Name of the SF_SafetyRequest instance	SF_<*d*><*f*>**_Mon**	SF_<*d*><*f*><*k*>**_Mon**
Symbolic name of the bit in the input image indicating that the drive function is now active	S_<*d*><*f*>**_fdbk**	S_<*d*><*f*><*k*>**_fdbk**
Symbolic name of the bit in the output image by which the drive function is requested	S_<*d*><*f*>**_ctrl**	S_<*d*><*f*><*k*>**_ctrl**

Where:
d is the name given to the safety drive in the application;
f is the acronym for the safety drive function; for standard safety drive functions the following common acronyms shall be used:
f ∈ { STO, SS1, SS2, SOS, SBC, SLA, SLI, SLP, SLS, SLT, SAR, SSR, STR, SDI, SEL, SCA, SSM, SMT };
k is the number of the instance of the safety drive function (if the drive is configured to have multiple instances of the same drive instance.

The specification can be downloaded from the PLCopen website.

info@PLCopen.org
www.PLCopen.org