A104 Safety Integrated

Module Description:

This Module describes the step by step procedures to start-up and commission Sinumerik System Based Safety Integrated for the Sinumerik 840D sl.

The procedures described within this module predominantly use Step7 & HMI Operate.
HMI Advanced (or Startup Tool) & SinucomNC are also required.

This Module is valid for >= S/W 04.05.xx.xx

It is expected that the student have a good understanding of the Sinumerik 840D sl control, including PLC programming, NC & Drive Commissioning prior to using this module.

Module Objective:

On completion of this module, the student will have a comprehensive knowledge of the Sinumerik System Based Safety Integrated Functions. This module aims to provide a defined step by step procedural approach to commissioning Sinumerik System Based Safety Integrated on the 840D sl control.

Caution: These procedures result in axis movement. All necessary safety precautions must be taken into account.

Content:

Sinumerik

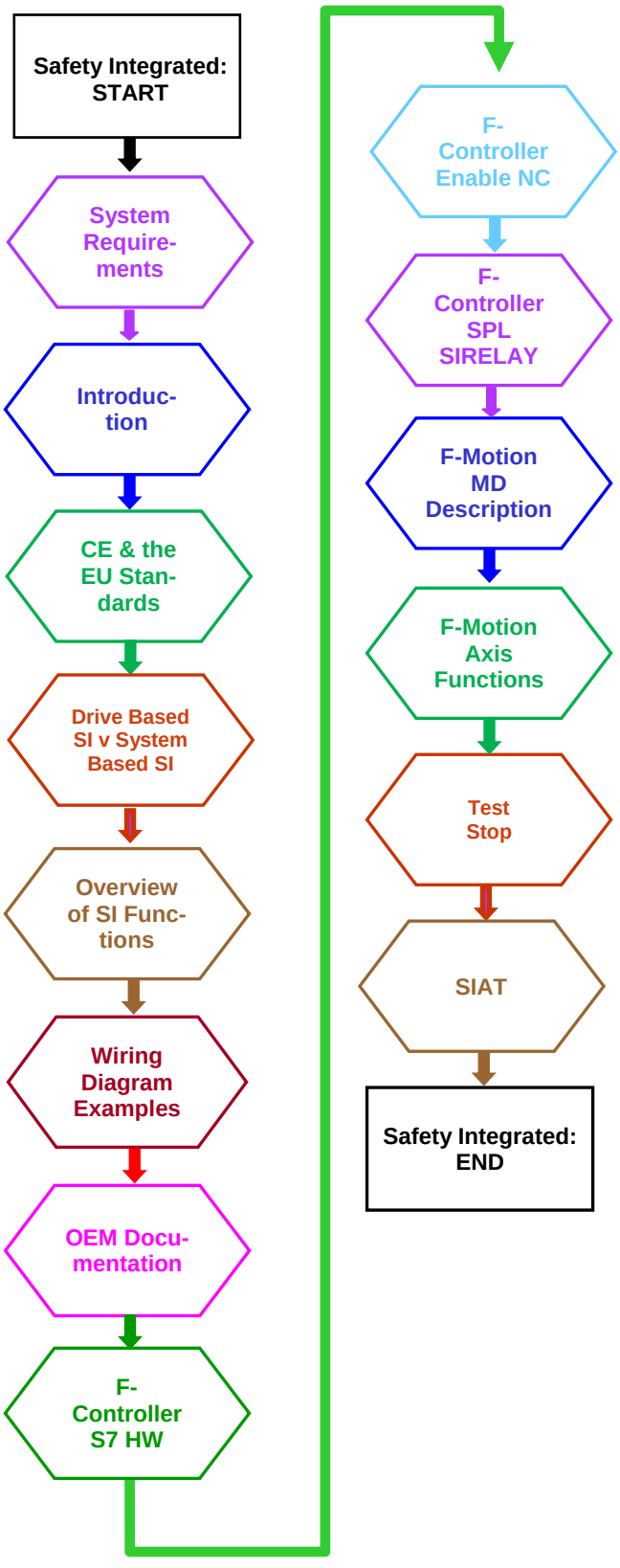
Safety Integrated: Description

This Module describes the step by step procedures to start-up and commission Sinumerik System Based Safety Integrated for the Sinumerik 840D sl.

The procedures described within this module use Step7 & HMI Operate.

This Module is valid for >= S/W 04.05.xx.xx

As a pre-requisite for this module, It is expected that the student have a good understanding of the Sinumerik 840D sl control, including PLC programming, NC & Drive Commissioning.



Notes :

System Requirements:

- Sinumerik 840D sl. All NCU types can be used.
- Safe Motion Monitoring can only be used in combination with the Sinamics S120 drive system. ADI4, Hydraulic axes...etc are not supported.
- Software Options 'Sinumerik Safety Integrated'…….

'SI Basic' Includes 1 Axis and a maximum of 4 SPL I/O	**6FC5800-0AM63-0YB0**
'SI Comfort' Includes 1 Axis and a maximum of 64 SPL I/O	**6FC5800-0AM64-0YB0**
'SI High Feature' Includes 1 Axis and a maximum of 192 SPL I/O (Requires Software >= 4.5)	**6FC5800-0AS68-0YB0**
'Axis/Spindle' Each additional axis/spindle	**6FC5800-0AC70-0YB0**
'Axis/Spindle Package' Additional 15 axis/spindle	**6FC5800-0AC60-0YB0**

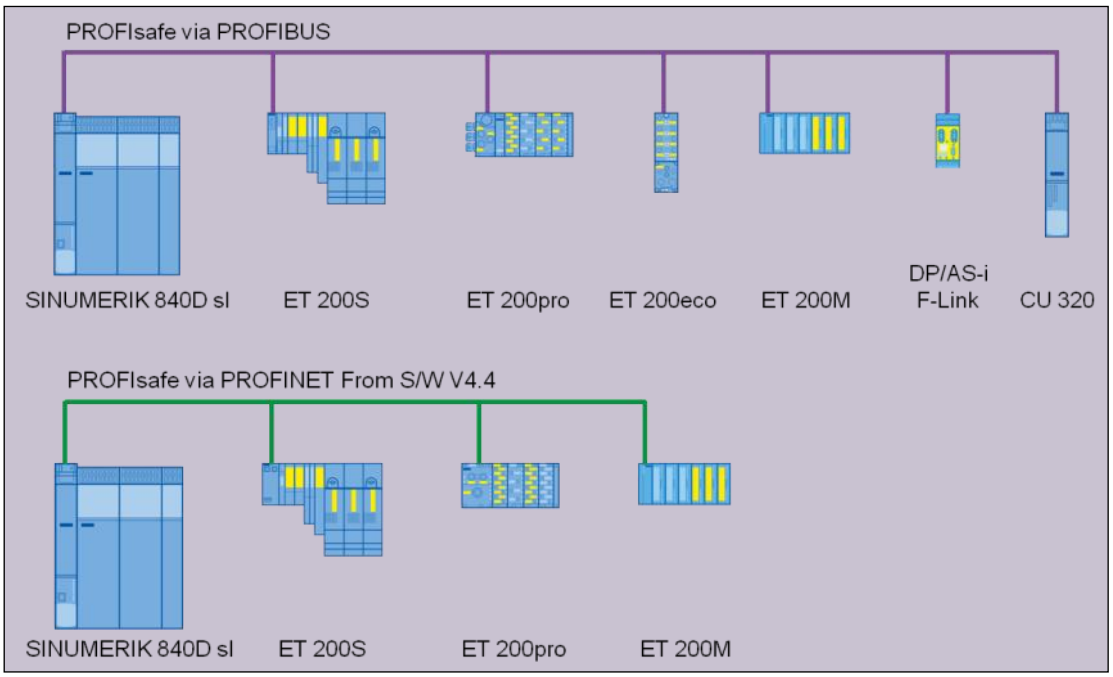
- Step 7 V5.5 SP3 or higher (No TIA Portal support at this time).
- * Step 7 F-Configuration Tool V5.5 SP9 or higher www.support.automation.siemens.com/WW/view/en/15208817
  This is a supplement to Step7 and is required so that ET200 F Modules can be integrated within the hardware configuration.
- SinucomNC. V7.6 or higher.
- Only explicitly released encoder systems may be used. A list of the Siemens encoders and motors permissible for Safety Integrated Functions can be obtained form your local Siemens contact partner.
  Only 1v Peak to Peak EDat V2.1 (ENDat 2.2 with additional incremental track - Order designation 02)
  or Incremental Drive-CliQ connected encoders are allowed. i.e SMI or SMC, SME...etc.
  The use of TTL, HTL, SSI, Resolver, ADI4….etc is not possible for Safety Integrated Functions.
- S120 Book size and up to two Chassis drives may be used. From S/W V4.4 the S120 Combo drive may also be used.

* The Step 7 add-on package 'Distributed Safety Programming' is not released for Sinumerik.

Notes :

System Requirements: Cont.

- The following F-Capable (Safe) I/O Peripherals are released for use;



The following specifically applies to Safety Functions integrated within the drives:

- The drive based 'Extended' Safety Integrated functions (Sinamics) cannot be used in conjunction with Sinumerik.
- A software option/license is not required for the drive based 'Basic' Safety Integrated functions (Sinamics).

  The following drive based 'Basic' Safety Integrated Functions can be used with Sinumerik but not in conjunction with an axis/spindle configured for 'System' based Safety.

  - Safe Torque Off STO
  - Safe Stop 1 SS1

  The drive based 'Basic' Safety Integrated Function for 'Safe Brake Control' (SBC) can be used in conjunction with Sinumerik 'System' based Safety Integrated.

  - Safe Brake Control SBC

Notes :

Introduction

- Why Safety?
- Advantages of Integrated Safety Technology

Notes :

Why safety? Machine safety is a must!

Machine manufacturers and operators are obliged by law to ensure the safety of people and the environment.

In other words: machinery which is manufactured or operated in Europe must be safe – whether it is new or used.

Functional safety also protects you from high costs!

There are also sound economic reasons for avoiding as far as possible every risk a machine may pose, right from the start:

- Avoidance of direct consequential costs of personal injuries (for medical care, wages and compensations)
- Avoidance of indirect consequential costs of injuries (e.g. fines due to non-compliance with regulations or repair costs)
- Enhanced productivity through increased machine availability: reduction of unplanned downtimes and smoother production workflows
- Prolonged service life of the systems
- Enhanced global competitiveness through improved export capability of machines

Advantages of integrated safety technology

The integration of safety technology in standard automation concepts entails considerable and sustainable user benefits for enhanced competitiveness. Machine manufacturers benefit from reduced hardware and significantly simplified engineering.

The result: Considerably faster realization of machines and systems as well as easier adjustability to new requirements.

The advantages for system operators: They are provided with safe and more productive machines and systems. A single integrated system of safety technology and standard automation reduces downtimes due to improved diagnostics and this also increases the system availability.

Retrofits and modernizations are simplified: Due to flexible, modularly expandable concepts, machines and systems can be upgraded to state-of-the-art technology way more efficiently.

Advantages of integrated safety technology...cont

Increased Efficiency

- Minimized type differentiations
- Minimized costs due to a single bus as well as engineering system
- Easily reproducible machines due to intelligent software solutions

Increased Standardization

- Simplified operation due to uniform user interfaces
- Improved reusability due to the application of libraries
- Reduced variety of control cabinets on machines
- Easier installation due to bus systems

Advantages of integrated safety technology...cont

Increased Productivity

- Faster commissioning, prewired and certified components
- Minimized downtimes due to quick fault localization and comprehensive diagnostics functions
- Quicker restarts after system modifications
- Production without standstills due to additionally available safe, fault-tolerant systems
- Space-, time- and cost-saving assembly

Increased Flexibility

- Tailored solutions due to modular system
- Easy expandability and integration in the Totally Integrated Automation architecture
- Improved global market opportunities due to availability of required approvals and compliance with EU directives
- Easier maintenance and system expandability due to long-term product and system availability

Notes :

CE and the European Standards

- CE certification
- EU Standards (IEC 62061 & ISO EN 13849-1)
- Risk Assessment and Reduction
- Determination of SIL(IEC 62061) and pl(Performance Level - ISO EN 13849-1)
- Evaluation of Safety Related Devices using the SET, the online Safety Evaluation Tool

Notes :

CE and the EU Standards

Functional Safety for Machines and Plants

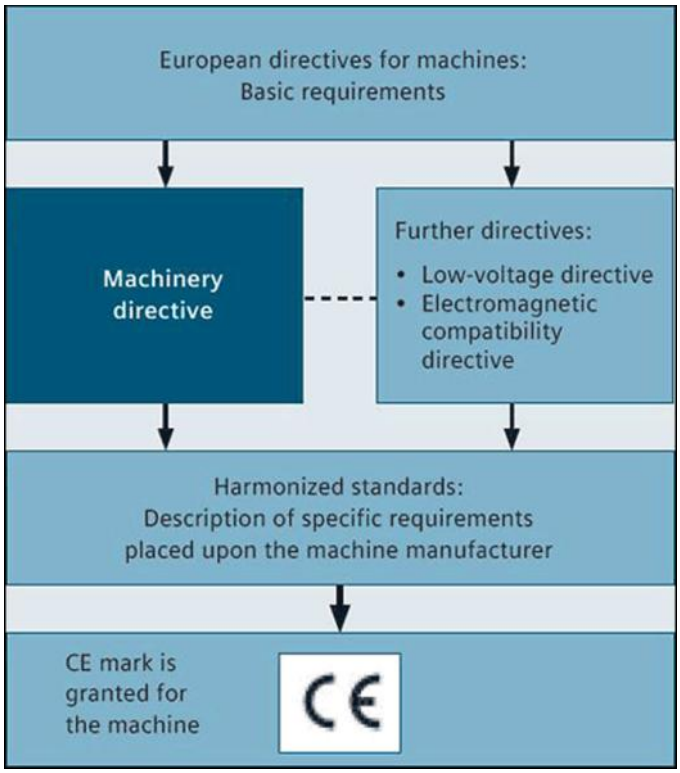
Mandatory in Europe, employed worldwide

European machine manufacturers (product safety) and machine operators (industrial safety) are required by law to ensure the protection of persons and the environment.

The safety awareness is constantly increasing in many other countries where such legal regulations are not yet in place.

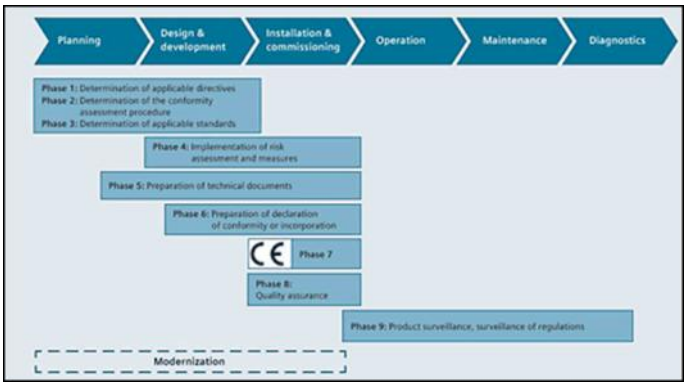
Machines “provided” in Europe have to be safe – whether they are new or old. In this context, the term “provision” implies that the machine is manufactured or significantly modified in Europe – or is imported to and operated in Europe.

The basic requirements for machine manufacturers or system operators who significantly modernize and modify their machines themselves are laid down in corresponding European directives – for example in the machinery directives, EMC directives, etc.



CE Mark

The CE conformity process according to the machinery directive is structured in a number of phases and describes individual activities which must be performed before the CE mark is applied.



Performing a Risk Assessment

The EU machinery directive demands 100% documentation of the risk assessment. Only with this can the directives, conformity assessments and standards drawn up in Phases 1-3 be appropriately applied to the machine.

The first step on the way to a safe machine is thus the risk assessment – in the very planning phase of the machine.

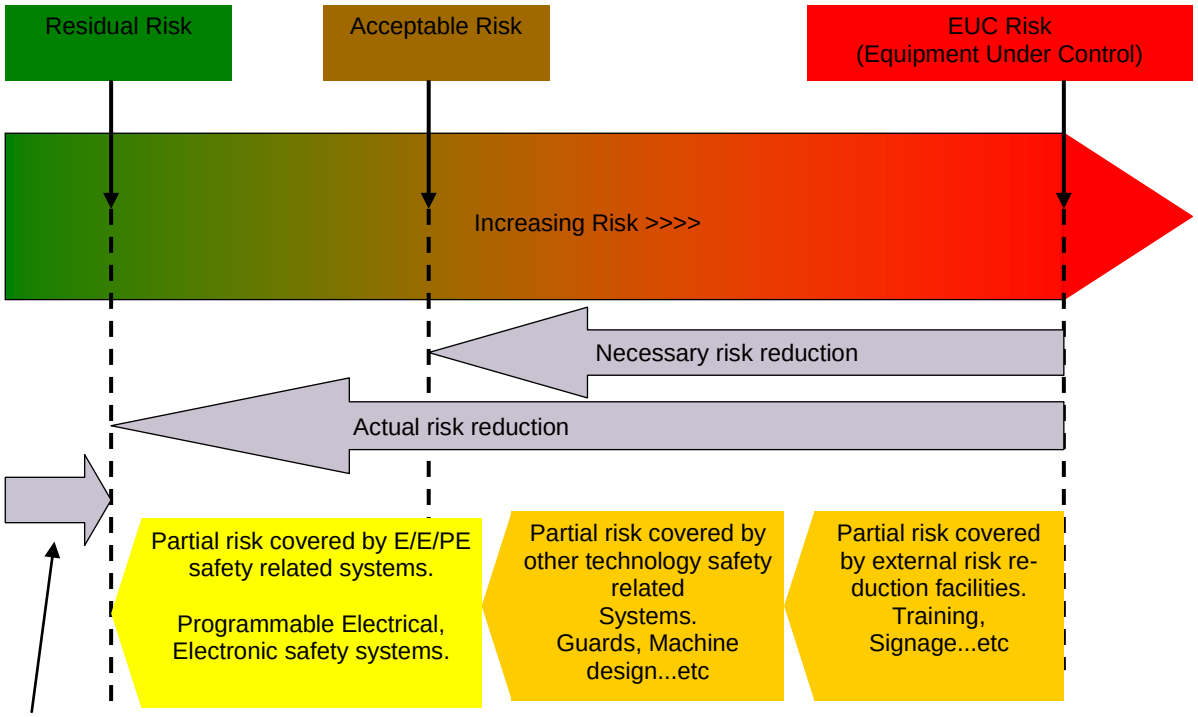
Risk assessment implementation is to be considered a design-accompanying process which is to be carried out by experts of various disciplines. In this context, the EN ISO 12100 standard offers support by description of an iterative procedure for risk assessment. Also as an example: in the US, this is covered by the ANSI B11.0 standard.

Following risk assessment, a decision as to whether risk reduction measures have to be initiated must be made. Such risk reduction comprises:

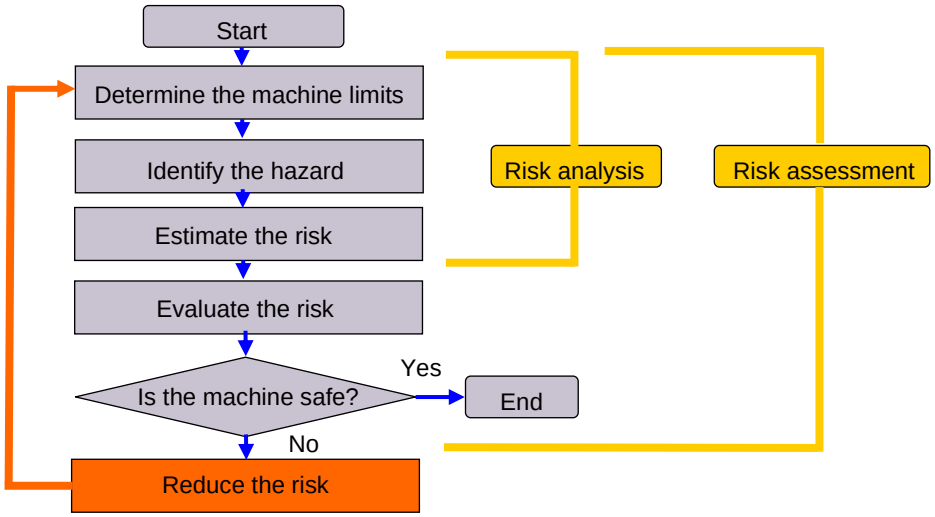
- Design measures
- Technical protective equipment as well as training measures for users

Notes :

Risk Reduction



Iterative Risk Reduction Process in accordance with EN 1050



Risk reduction and the selection of appropriate safety measures are not part of the risk assessment process. For further information, refer to EN ISO 12100-1/-2

Notes :

Application of the Safety Standards EN ISO 13849-1 and EN 62061

Conformity with the new machinery directive, and thus exportability and security against liability, can be achieved by machine manufacturers through application of the EN ISO 13849-1 and EN 62061 standards. These have introduced not only qualitative observations but also quantitative aspects, whereby “functional safety” and the “safety function” play an important part.
Risk-reducing protective measures described by way of safety functions can be derived from the risk evaluation process. The safety function solution is then checked and evaluated with hardware and possibly software components until the level of safety integrity specified in the risk assessment has been attained.

Once the risk assessment is completed, the OEM will then choose an appropriate standard for his machine:

EN 62061 - Based on SIL (Safety Integrity Level) assignment.
or
EN ISO 13849-1 - Based on PL (Performance Level) by risk graph.

The risk elements (**S, F, O** and **P**) serve as input variables for both EN62061 and EN ISO 13849-1. The risk elements are evaluated in different ways; according to EN 62061, a required safety integrity level (SIL) is determined.
According to EN ISO 13849-1, a performance level (PL) is determined.

Determination of required SIL (Safety Integrity Level) for EN 62061 (SIL 1, 2 or 3)

Frequency and/or duration of stay **F**		Occurrence Probability of Hazardous situation **O**		Prevention Possibilities **P**	
=< 1 hour	5	Frequently	5		
> 1 hour to =< 1 day	5	Probable	4		
> 1 Day to =< 2 Weeks	4	Possible	3	Impossible	5
> 2 weeks to =< 1 year	3	Rarely	2	Possible	3
> 1 year	2	Negligible	1	Probable	1

Effects	Severity **S**	Class **Cl = F + O + P**				
		3-4	5-7	8-10	**11-13**	14-15
Death, loss of eye or arm	4	SIL 2	SIL 2	SIL 2	SIL 3	SIL 3
Permanent, loss of fingers	**3**			SIL 1	**SIL 2**	SIL 3
Reversible, medical treatment	2	Other Measures			SIL 1	SIL 2
Reversible, first aid	1					SIL 1

Example:

Hazard	S	F	+	O	+	P	=	Cl	Safety Measures	Safe
Rotating Spindle	3	5	+	4	+	3	=	**12**	Monitor protective cover with SIL 2	Yes

Notes :

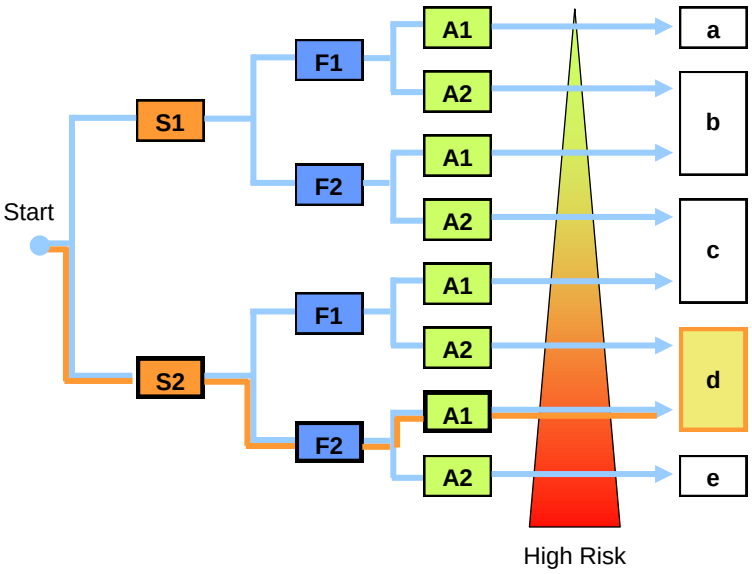
Determination of required Performance Level (PL) for EN ISO 13849-162061 (PL a - e)

Risk Parameters

S = Severity of Injury
- **S1** = Slight (Usually reversible) injury
- **S2** = Severe (Usually irreversible) injury including death

F = Frequency and/or duration of exposure to hazard
- **F1** = Rare to often and/or short exposure to hazard
- **F2** = Frequent to continuous and/or Long exposure to hazard

P = Probability of avoiding or limiting harm
- **P1** = Possible under certain circumstances
- **P2** = Hardly possible



Evaluation of Safety Related Devices

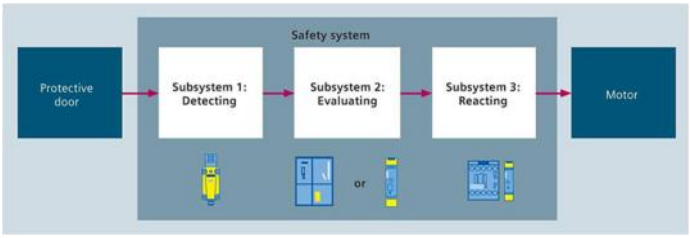
Once the 'SIL' or 'PL' has been determined for the machine. The OEM must then choose the various safety related devices that will make up the different 'Safe Systems' on the machine. i.e. Emergency Stop, Operator Door….etc

It is the responsibility of the OEM to prove and document that the combination and integration of the different safety related devices achieve the 'SIL' or 'PL' required.

The Safety Evaluation Tool for the IEC 62061 and ISO 13849-1 standards takes you to your goal directly.

This TÜV-tested online tool supports the fast and reliable assessment of your machine's safety functions.

As a result, you are provided with a standard-compliant report, which can be integrated in the documentation as a proof of



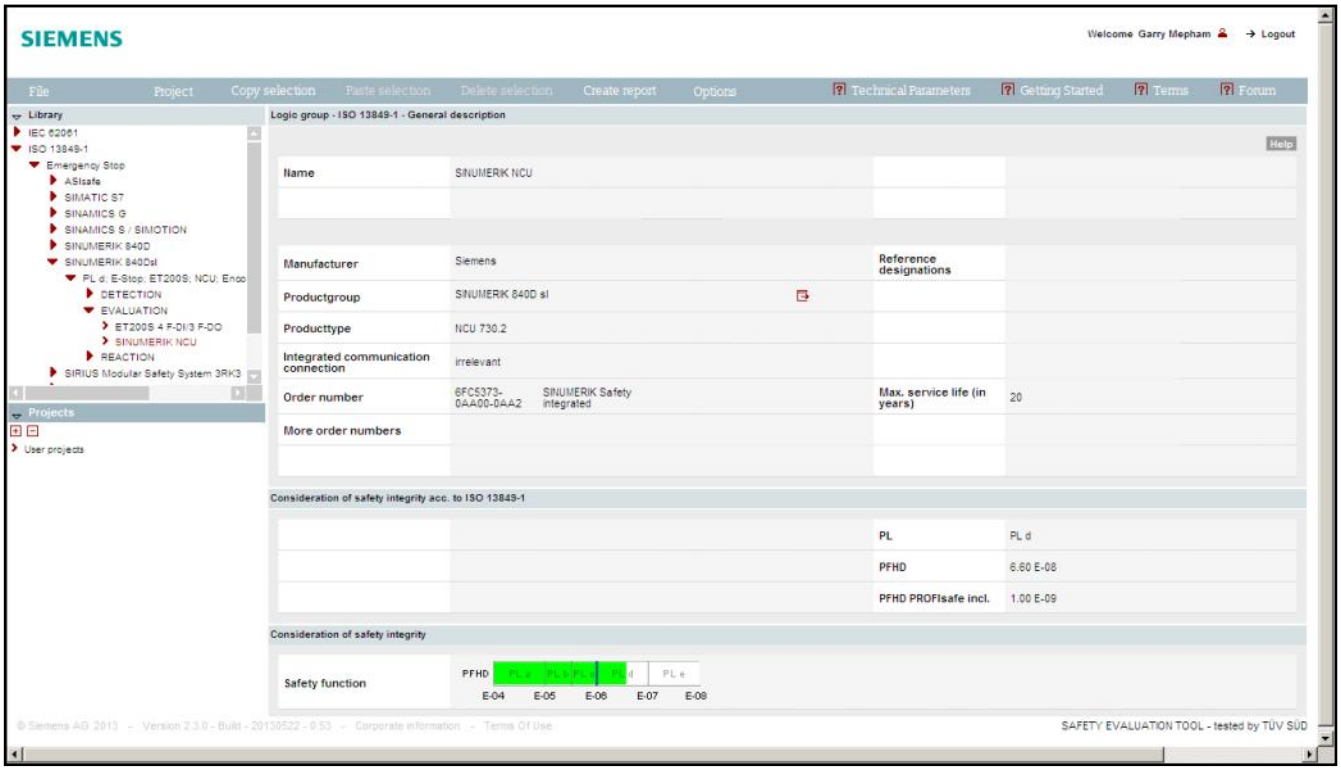
Advantages of using the Safety Evaluation Tool

- Certainty regarding compliance with the standards: TÜV-tested tool
- Free use of the online tool
- Automatic calculation in accordance with current standards
- Rapid result: report compliant with the standards
- Time savings in terms of safety function assessment
- Fast access to current product data
- Comfortable archiving: Projects can be saved and called up again as required
- Fast and easy handling: Comprehensive, pre-defined example libraries
- Selection menus for determining the DC and CCF
- Input of different operating cycles with a two-channel design
- Calculation of failure rate
- Selection assistant for drive components

Here is a link to the Safety Evaluation Tool

Notes :

Online Safety Evaluation Tool (SET).



The Safety Evaluation Tool contains useful examples based on Sinumerik 840D sl. These examples cover both IEC 62061 and ISO EN 13849-1.

The tool can produce a multi-page report that confirms the safety integrity of the safety related components used in your project. This report can form part of the technical file for the machine.

SET projects can be saved locally and loaded again in the future.

The tool provides online help and tutorials to help you get started.

Notes :

Drive Based SI v System Based SI

- Drive Based Functions v System Based Functions

Notes :

Drive vs System Based Safety Integrated Functions

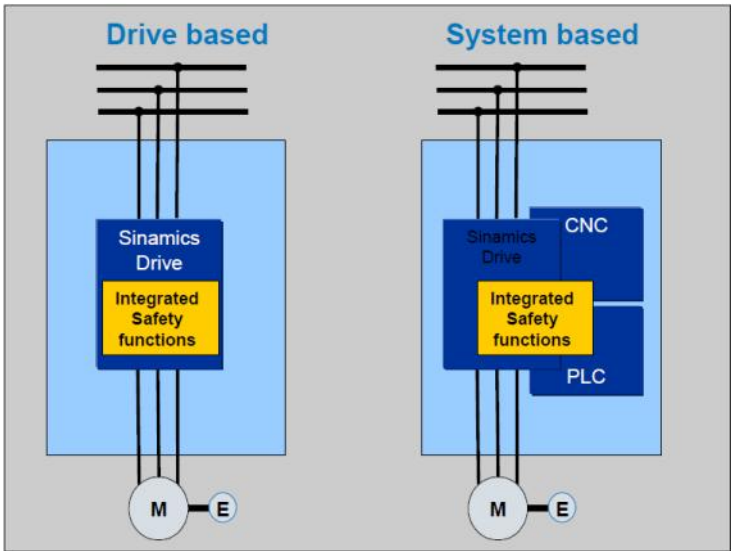
For the Sinumerik 840D sl, there are two concepts of Safety Integrated that can be applied:

- Drive Based Safety Functions (Only Basic Functions)

-STO Safe Torque Off
-SS1 Safe Stop 1
-SBC Safe Brake Control

- System Based Safety Functions

-SH (STO) Safe Standstill
-SBH (SOS) Safe Operational Stop
-Safe Stopping Process (External Stops A-E)
-SBR Safe Braking Ramp
-SG (SLS) Safe Reduced/Limited Speed
-Safe Speed Limit/Monitor ($n<n_x$) (SSM)
-SE (SLP) Safe Software Limit Switches
Safe Limited Position.
-SN (SCA) Safe Software Cams
-SBC Safe Brake Control
-SBT Safe Brake Test
-SGE/SGA Safety Related Inputs/Outputs:
- Profibus
- Profinet
-SPL Safe Programmable Logic
-Safe Communication:
- Profibus
- Profinet



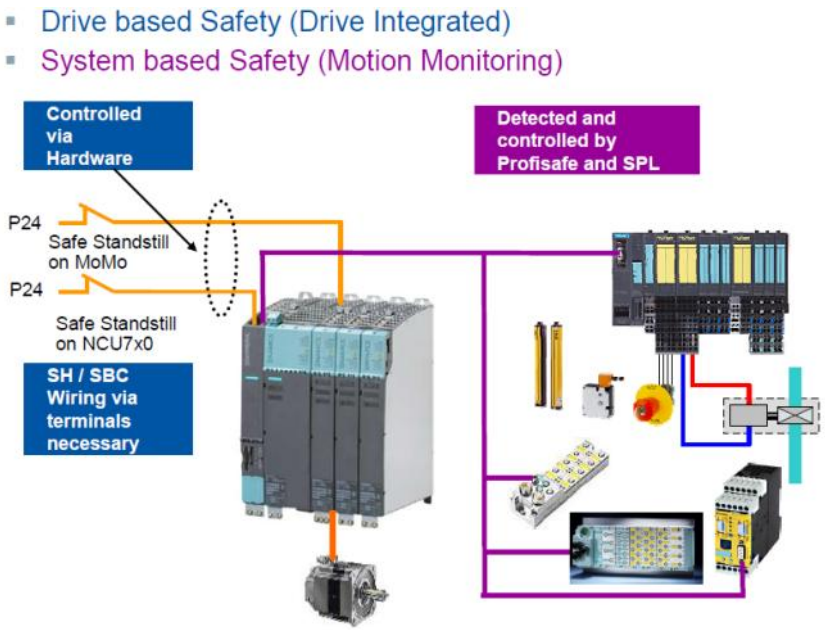
STO, SS1, SBC, SLS, SSM, SLP & SCA refer to naming conventions of IEC61800-5-2.

Drive vs System Based Safety Integrated Functions

The two concepts can both be applied within the same control to different axes/spindles. However, the 'Drive Based' function for SBC (Safe Brake Control) can also be applied to axes/spindles that are configured for 'System Based Safety'.

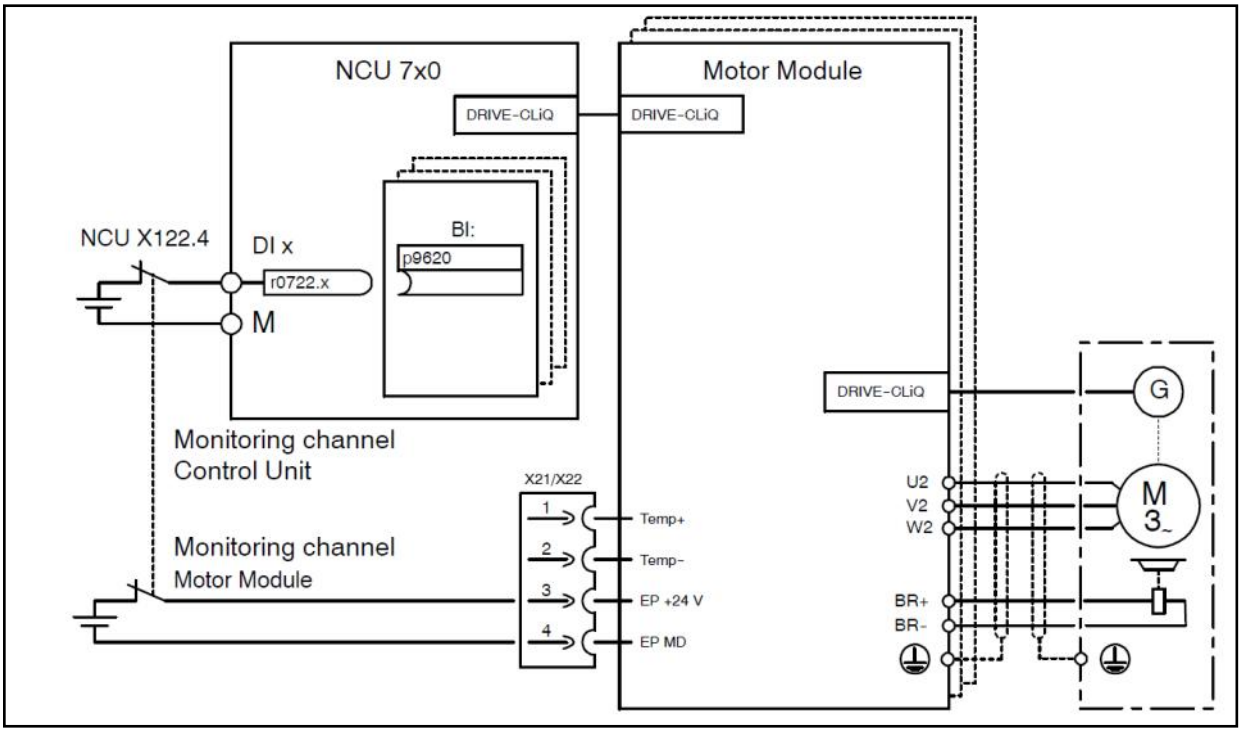
Sinamics Drive Based Safety requires an external hardware interface to control the Safety Functions.

Sinumerik System Based Safety allows for Safety Related Devices to be directly connected via F-Capable peripherals for monitoring and control by the SPL (Safe Programmable Logic) within the control. The SPL can thus control the 'System Based Safety Functions'.



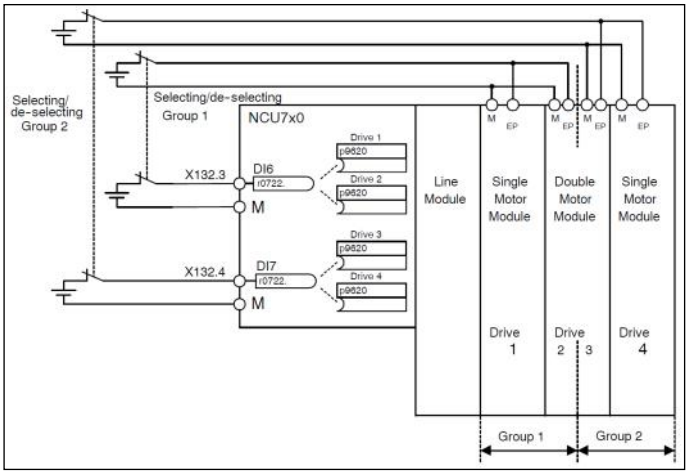
Notes :

Typical wiring diagram for Drive Based Safety:



By using a second CU input, e.g. NCU X122.3, a second group of axes can be created for independent Drive Based Safety Integrated (DBsi) functions:

Note: For System Based Safety Integrated, wiring of the CU inputs or EP terminals of the motor modules is not required.



Notes :

Overview of System Based Safety Integrated Functions

- Sinumerik 840D sl and Safety
- Scope of included functions
- Brief description of all functions
- Safe Standstill and Safe Operational Stop
- Safe Stopping Process
- Safe Braking Ramp
- Safe Reduced Velocity
- Safe Speed Monitor
- Safe End Switches
- Safe Cams
- Safe Brake Management, including Safe Brake Control and Safe Brake Test
- Safety Related I/O
- Safe Programmable Logic

Notes :

SINUMERIK 840D sl Safety Integrated

SINUMERIK Safety Integrated is a comprehensive safety package that helps protect both man and machinery.

It ensures high efficiency and profitability due to the complete integration of safety functions into the control and drive technology.

In addition, you can safely and practicably operate your machine under all operating conditions requested, for example, in setup and test mode with open protective door.

The safety functions fulfill all requirements according to DIN EN 61508 for applications up to and including SIL 2 (Safety Integrity Level) and Category 3 as well as PL d (Performance Level) according to DIN EN ISO 13849. You can thus conveniently and economically fulfill the major requirements placed on functional safety.

Included in the function scope are the following, e.g.:

- Functions to safely stop drives
- Functions to safely monitor motion
- Direct connection of all safety-related signals and internal logic operation.

Product features

- Functions for operator and machine protection
- Short response times due to integration into the drive system
- Suitable for all modes of operation
- Replaces external electromechanical devices

Customer benefits

- High safety: Contiguous realization of Class 3 / SIL 2 safety functions (EN 954-1)
- High flexibility: Practice-oriented safety and operating concepts can be realized
- High cost effectiveness: Reduction of hardware and installation costs
- High availability: Electromechanically operated contact mechanisms susceptible to interference are no longer required

Functions

The safety functions are available in all modes and can communicate with the process via safety-related input/output signals.

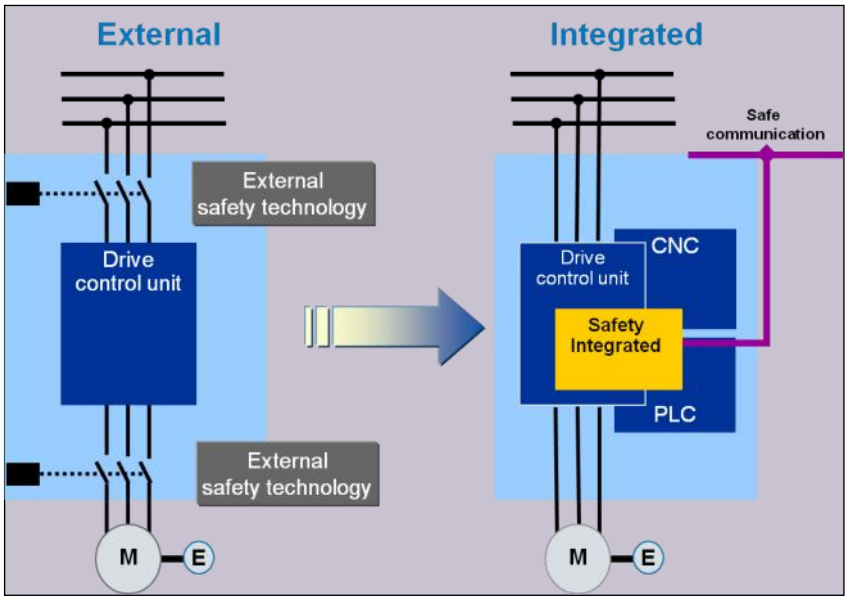
The functions can be realized for each individual axis and spindle:

- Safe stopping process Safely transfers the drives from motion to standstill as soon as a monitoring function or a sensor (e.g. a light barrier) responds.
- Safe acceleration monitoring (SAM) This function monitors the speed characteristic. The speed must be decreased after a stop request
- Safe operation stop (SBH) This safety function monitors the drives for standstill. The drives are fully functional in the bearing.
- Safe standstill (SH) Pulse suppression of the drives and safe, electronic interruption of power supply
- Safely reduced speed (SG) Monitoring of configurable speed limits, e.g. upon setup without pressing an enable key.
- Safety-related output n < nx Safe velocity detection of a drive
- Safe software limit switches (SE) Variable traversing range limits
- Safe software cams (SN) Area recognition
- Safety-relevant input/output signals (SGE/SGA) Process interface
- Safe programmable logic (SPL) Direct connection of all safety-relevant signals and their internal logic combination.
- Safe brake management (SBM) - 2-channel brake control (SBC) (integrated in the Motor Module) - Cyclic brake test (SBT)
- Safety-relevant communication via standard bus - Connection of distributed I/O for process and safety signals via PROFIBUS/PROFINET with the PROFIsafe protocol - Safety-related CPU – CPU communication
- Safe network transition between ASI and PROFIBUS The DP/AS-I F-Link provides for a safe transition from ASIsafe to PROFIsafe. All safety-related signals are collected via the AS-i bus.
- Integrated acceptance test Partially automated acceptance test for all safety-relevant functions. Convenient operation of the test sequence, automatic configuration of trace functions and automatic generation of the acceptance certificate.

Notes :

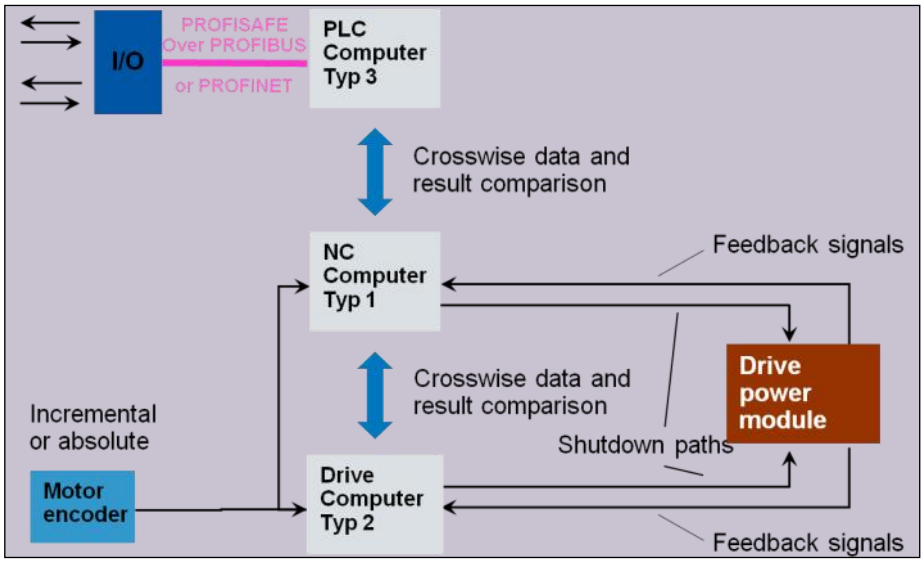
System Based Safety

Today, Safety Technology Moving Towards Software and Electronics.



System based 'Integrated Safety' allows for much faster reaction times of the safety system with 'built in' diagnostics.

The existing CPU's form a two-channel diverse system structure.



With only the addition of ProfiSafe I/O, the standard architecture of the 840D sl & Sinamics S120 drives allow the system based safety integrated features to be implemented. No extra hardware required!

Separate processes in the NC and Drives give redundancy for safe motion monitoring functions. Each process has its own 'Shutdown Path' that can put each drive into a 'Safe State'. Safe functions can be implemented with a single encoder, Absolute or incremental.

Notes :

Overview of the System Based Safety Functions:

SH Safe Standstill (Safe torque off (STO))

The drive pulses are cancelled. The motor and drive power are safety and electronically disconnected.

SBH Safe Operational Stop (Safe operating stop (SOS))

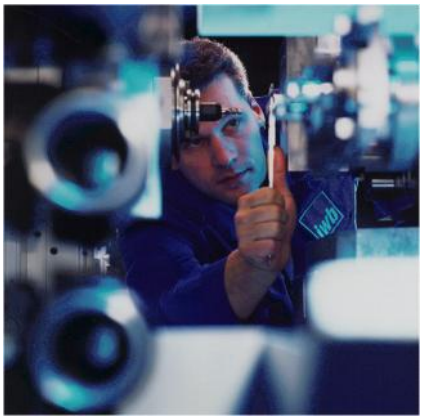
The motor/encoder are safely monitored for standstill. The drive and motor remain fully operational in closed loop control.

Safe working with protective door open to allow….

- Checking
- Measuring
- Cleaning
- Aligning
- Removal of Swarf(Chips)
- Changing of Workpieces
- Changing Tools
- ....

Advantages….

- Fast Restart of machine
- Less brake wear
- No loss of accuracy
- ….



Operational axes….

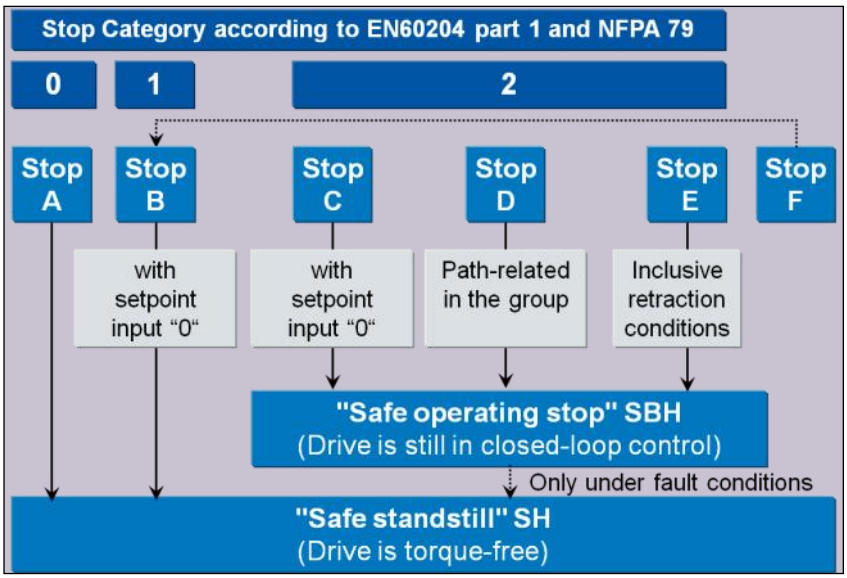
- Vertical Axes
- Non-symmetrical work-pieces
- Axes without friction (linear motors)

Safe pulse disabled axes….

- Spindle can be rotated manually
- Emergency Stop

Safe Stopping Process (Safe External Stops A-E)

When a monitoring function or sensor responds (e.g. Light Barrier), the drives are safely controlled to standstill. The axes can remain operational and are safety monitored for standstill.

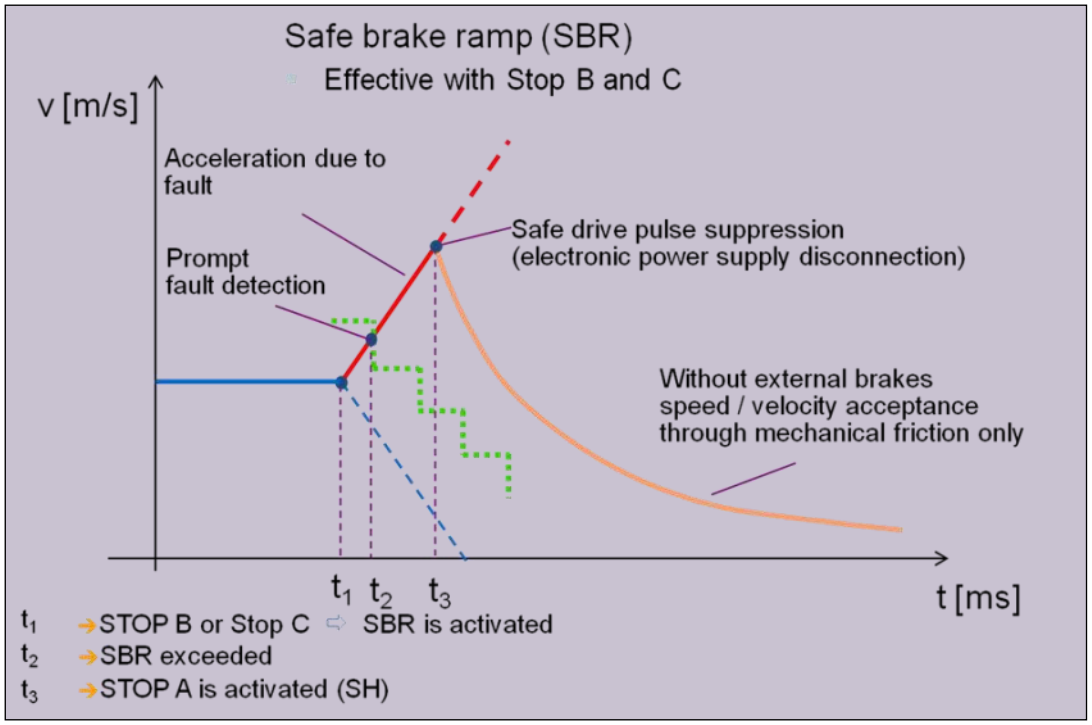


Stopping process can be adapted to suit machine. Axes can be braked along the current acceleration profile or at axis specific current limit. Stop's A, C, D & E can be directly controlled by the safe logic.

Notes :

SBR Safe Braking Ramp

Monitors the speed characteristic. The speed must be reduced after a Stop request has been issued.



SG Safe Reduced Velocity

The axes/spindles are subject to configurable safe limits.
There are four velocity levels per axis.
Levels 2 & 4 can be subject to a 16 position override thus giving upto 34 different 'Safe' velocity limits per axis/ spindle.

Working with the protective door open….

- Single handed operation for jogging axes
- Spindle can be rotated (With enable button)
- Programs tested i.e. measuring cycles (With enable button)

Special Applications….

- Burst/Rupture protection for grinding wheels
- Max. spindle speed (depending on chuck)
- Protective Door only released below safe speed

Advantages….

- Improved Machine Control
- High flexibility, usability, Operator friendly

Notes :

Safe Speed Monitor N < Nx

Filtering:

For actual value fluctuations, e.g. as a result of mechanical vibration of the machine, signal, $n < n_x$ is kept more stable.

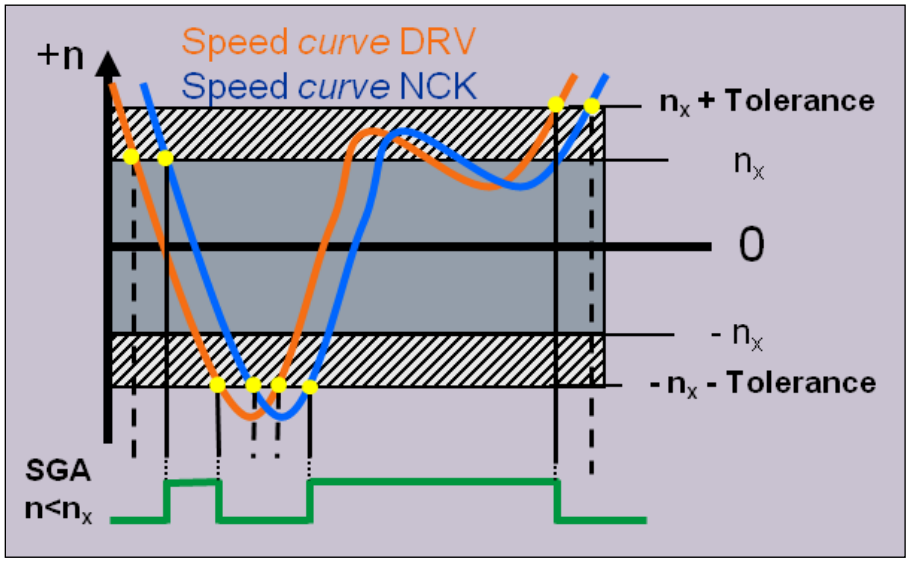
Hysteresis:

Continuous switching is prevented for slight speed fluctuations ("flicker") around the threshold "n_x".

Synchronization:

Definition of a tolerance bandwidth.

Different signal states within the Tolerance bandwidth no longer result in a fault.

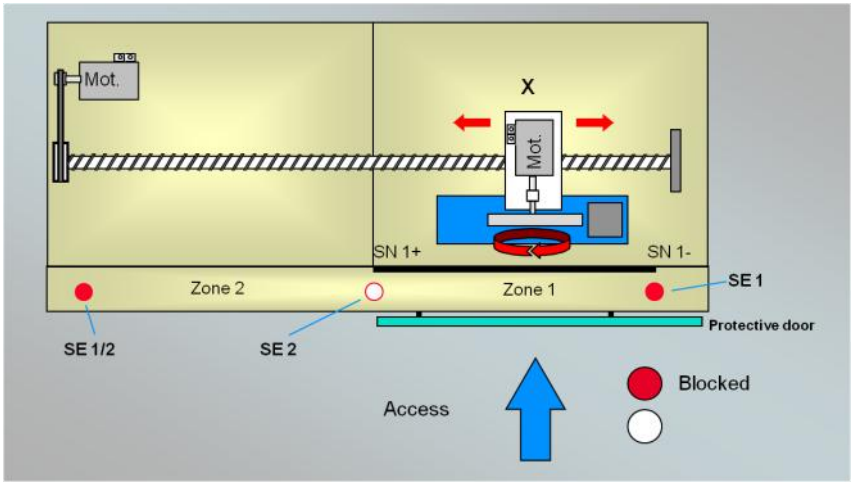


Safe software limit switch (SE)

- Two selectable Safe End-Switch pairs per axis
- A Safe Stop response is configured if the traversing range is violated

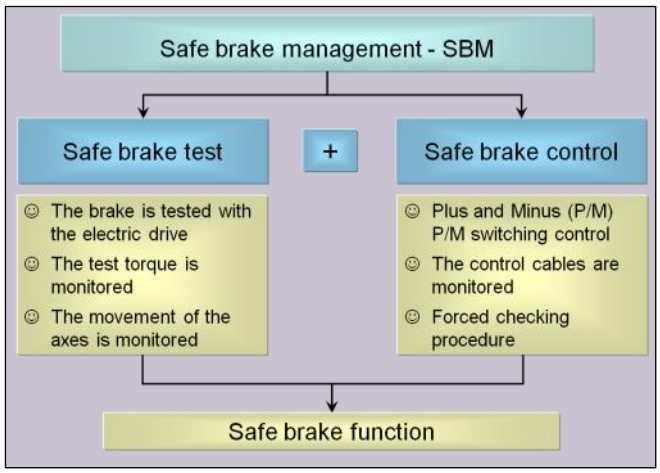
Safe software cams (SN)

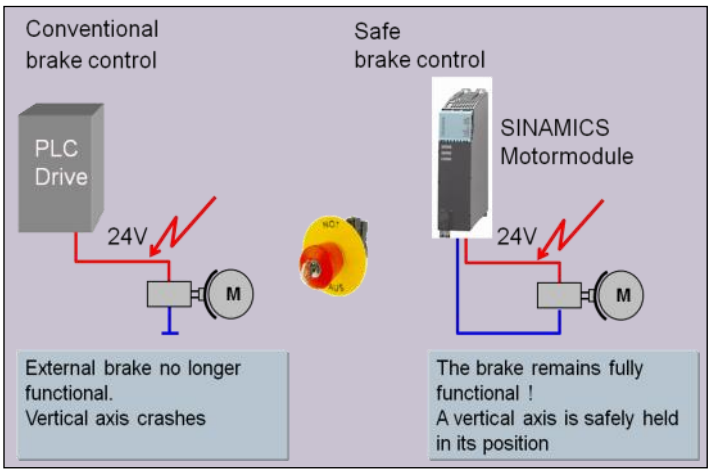
- Variable range detection. 4 Cam Pairs per axis
- Safe Cam track. Upto 30 Cams per axis



Notes :

‘Safe Brake Control - SBC’ and ‘Safe Brake Test’ together form ‘Safe Brake Management’.



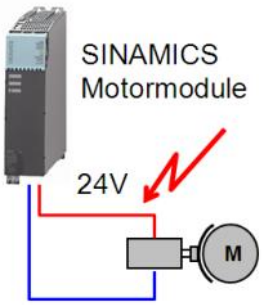


Significantly reduced risk
"Safe brake management" allows the system to safely perform all brake applications within a uniform process.

Less machine damage
The direct, electronic control allows to achieve minimum response times and reduces axis sagging. This prevents production failures following machine crash and increases machine availability.

Reduced brake wear
By using the function "safe operating stop - SBH (SOS)" function, the vertical axes only ever needs to be held by the mechanical brake device in exceptional cases.

The existing safe drive is considered as a fully adequate, safe holding system When using the ‘Safe Operational Stop - SBH (SOS)’ function.



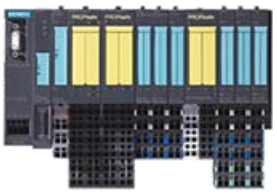
With “safe brake management“, the existing mechanical brake is considered as a fully adequate, safe holding system.

Notes :

Safety Related Inputs and Output Signals (SGE/SGA)

Failsafe I/O systems via ProfiBus or ProfiNet.

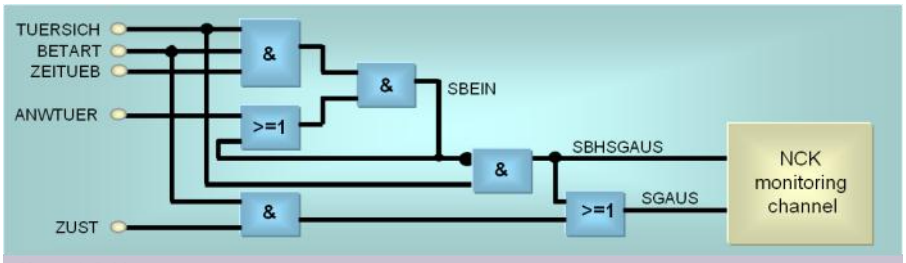
- ET200s
  - 4 F-DI 6ES7 138-4FA0_-0AB0
  - 4 F-DO 6ES7 138-4FB0_-0AB0
  - 4 F-DI / 3 F-DO 6ES7 138-4FC0_-0AB0
  - PME-F 6ES7 138-4CF0_-0AB0
  - F-RO 6ES7 138-4FR0_-0AA0



- ET200eco (ProfiBus Only)
- ET200pro
- ET200M (V4.4 Sp1)
- DP/ASi – F Link
- Safe Communication to F-CPU or other Sinumerik 840D sl over ProfiBus or ProfiNet.

Safe Programmable Logic (SPL)

- Direct connection of safety related devices via Safety Related I/O - SGE/SGA.
- Cyclic execution of the SPL independent of the user program.
- Independent of all NC operating modes and status.
- SPL active at system run-up.




```
;Macro definitions
        DEFINE   TUERSICH  AS    $A_INSE[1]      ; external SPL interface (to the periphery)
        DEFINE   BETART    AS    $A_INSE[2]
        DEFINE   ZEITUEB   AS    $A_INSE[3]
        DEFINE   ANWTUER   AS    $A_INSE[4]
        DEFINE   ZUST      AS    $A_INSE[5]
        DEFINE   SBEIN     AS    $A_MARKERSI[1]  ; internal SPL status flags
        DEFINE   SBHSGAUSAS      $A_OUTSI[1]     ; internal SPL interface to the NCK monitoring channel
        DEFINE   SGAUS     AS    $A_OUTSI[2]
;Logic operations
        IDS=01 DO SBEIN = (TUERSICH AND BETART AND ZEITUEB) AND (ANWTUER OR SBEIN)
        IDS=02 DO SBHSGAUS = TUERSICH AND NOT SBEIN
        IDS=03 DO SGAUS = SBHSGAUS OR (ZUST AND BETART)
;End of program
        M17
```

Notes :

System Based Safety Wiring Diagram Examples

- Basic wiring example
- Detailed wiring example

Notes :

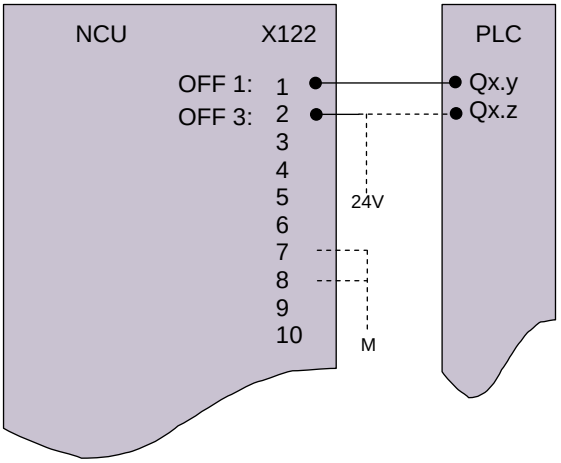
Due to the ability to directly connect Safety Related Devices to the 840D using Safe Inputs and Outputs, the use of conventional safe monitoring relay devices is no longer required.

Guard Door switches, Emergency Stop push-buttons, light barriers...etc can all be directly connected and controlled.

Depending on the application, it may also be possible to remove the power contactor that supplies the drive power.

Hard-wired over-travel or end-limit switches are also no longer required as this function can no be done safely within the system.

Example of the required wiring to the 840D NCU X122.



The control of X122.2 for OFF3 has the following possibilities:

- PLC Output
- Connected directly to 24V
- P849[0] = BICO 1 in each drive

Detailed Wiring Example

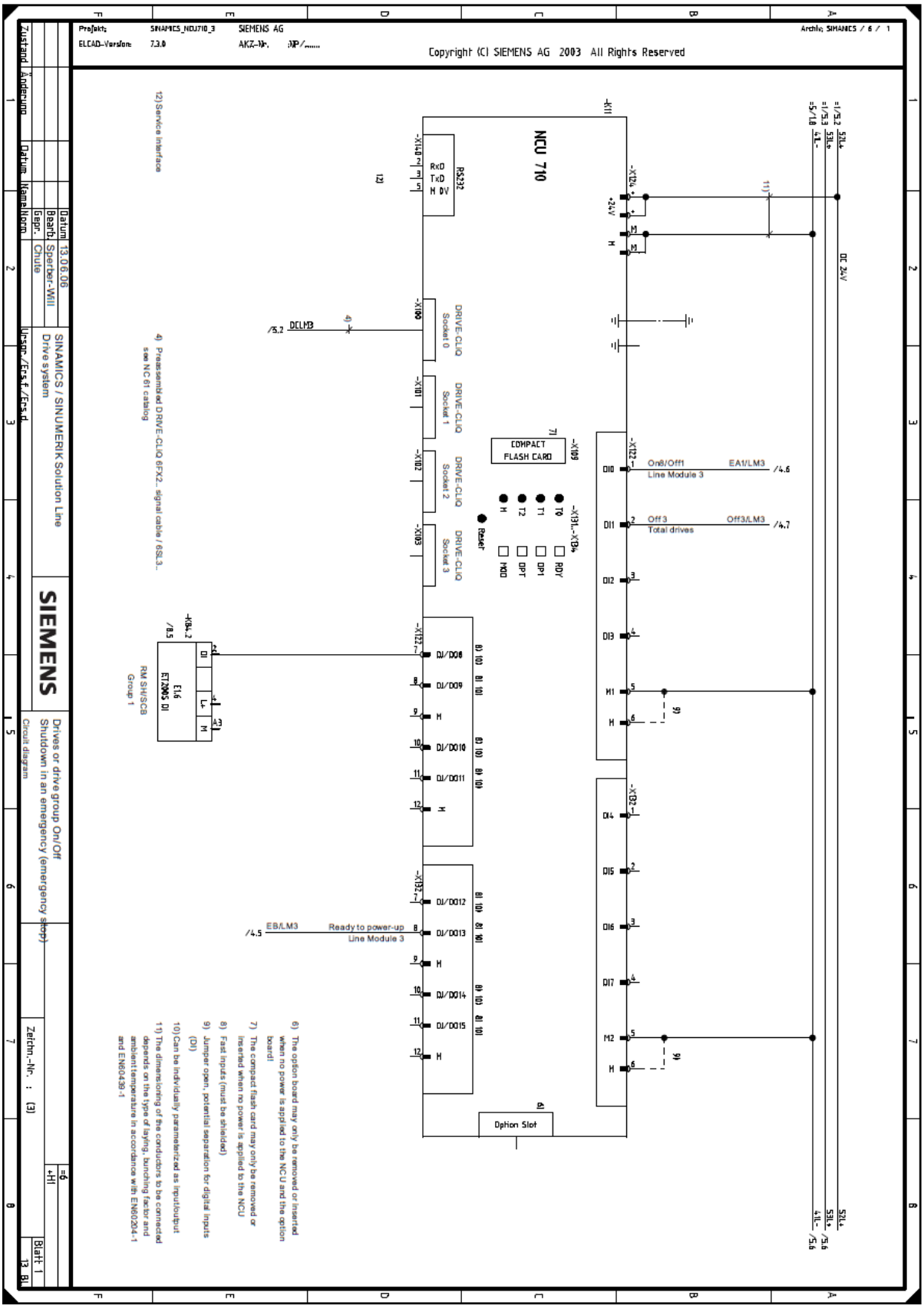
A drive group without main contactor, however, with the system/drive-integrated safety function of the SINUMERIK 840D sl; connection of the safety sensor technology / actuators using ET200S.

- Power supply of the NCU 710 (=6.-K11)
- Control signals to the NCU 710; digital inputs and digital outputs using the data of the standard macros
- DRIVE-CLiQ connection to the power units
- PROFIBUS connection to the ET200S distributed peripherals (=6.-K82)
- Control signals to the ET200S distributed peripherals
- Supply and control of the Line Module, including the integration of the auxiliary contact of the main power switch
- Supply and control of the Double Motor Module (=6.-Q61)
- Motor, holding brake and sensor system (motor-integrated sensor component) connection to the Motor Module
- Constant motor outgoing feeder using ET200S "Standard" motor starter (=6.-Q86.1)
- Constant motor outgoing feeder using ET200S "Fail-safe" motor starter (=6.-Q95.1)
- Safety control with ET200S safe digital input modules (=6.-K92.1, -K93) and output modules (=6.-K94)
- Safe interface level for external signals (=6.-K132, -K134)

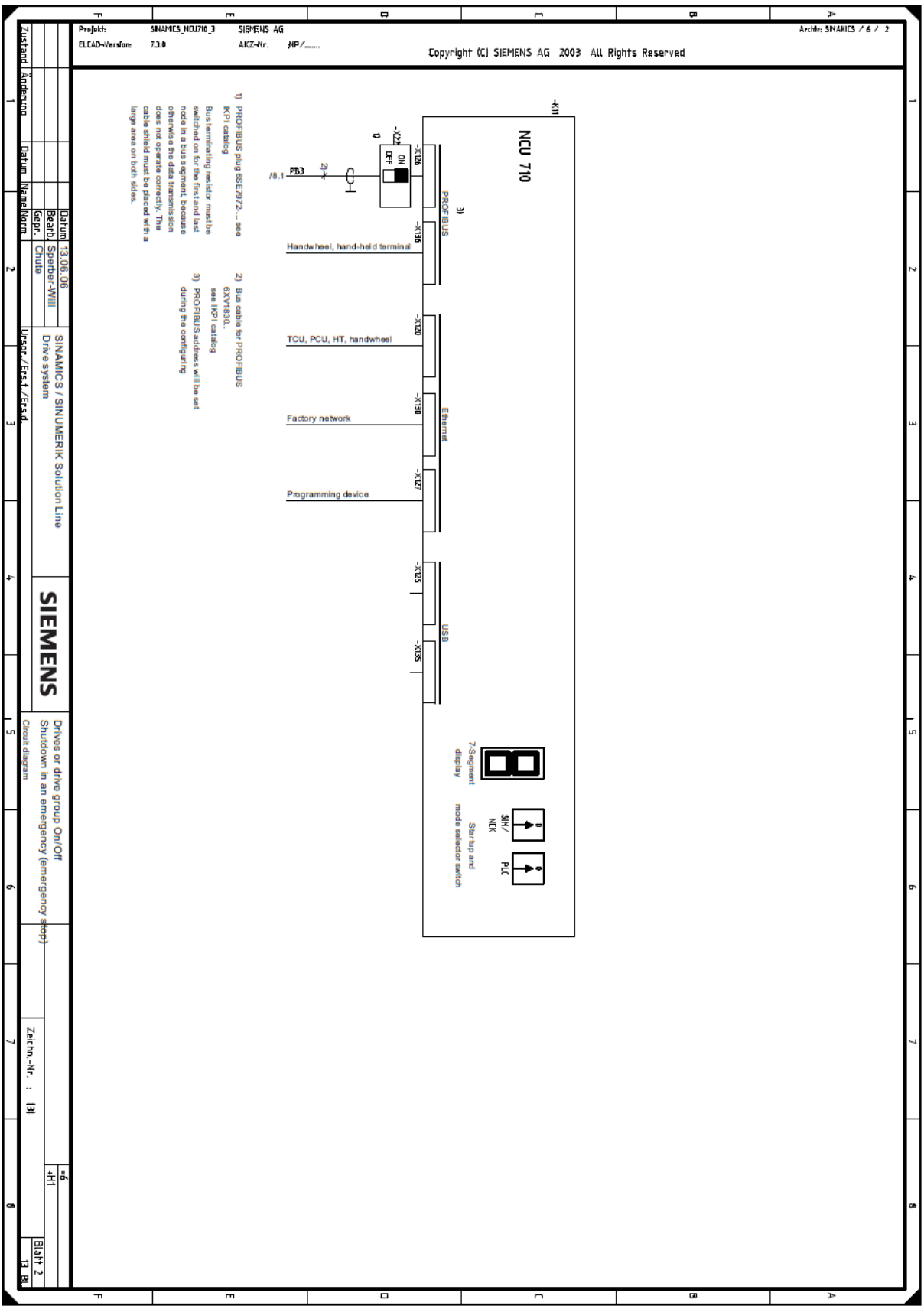
All sensor/actuator signals required for the safety function are connected to the fail-safe modules of the ET200S peripherals (=6.-K92.1, -K93, -K94, -K95). All fail-safe signals are linked using the system/drive-integrated safety function of the SINUMERIK 840D sl. External safety relays or safety controllers are no longer required. An external control of the SH/SBC function using the terminals on the NCU or on the Motor Modules is also not required but is performed internally by the system.

Notice that the OFF1 (X122.1) is now controlled by a standard PLC output and that there is no longer a requirement for a power contactor. OFF3 (X122.2) is shown here also connected to a standard PLC output. However, it could also be connected to +24V or BICO=1 using p849 in each drive.

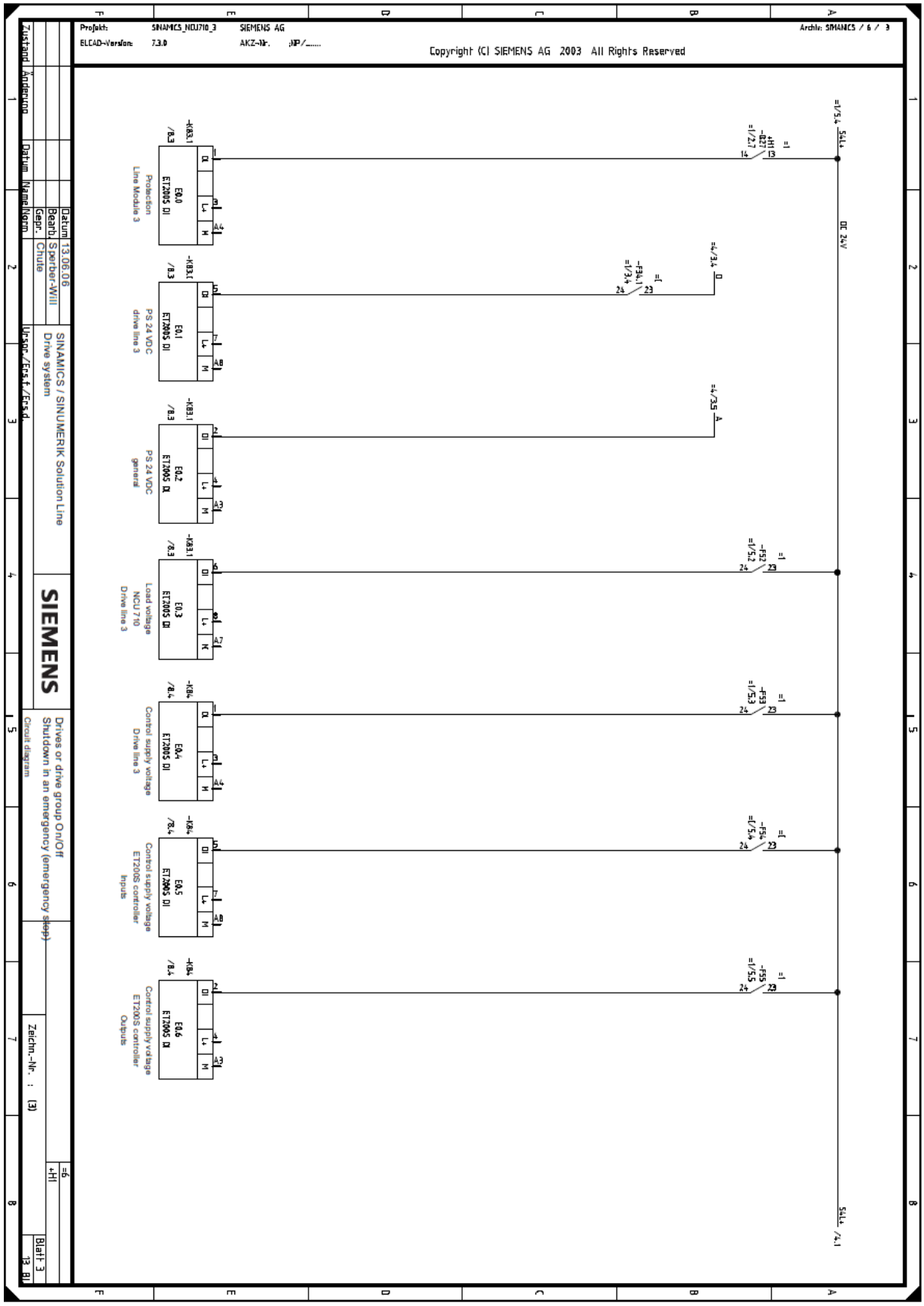
Notes :


NCU 710
COMPACT FLASH CARD
Option Slot
DRIVE-CLiQ Socket 0
DRIVE-CLiQ Socket 1
DRIVE-CLiQ Socket 2
DRIVE-CLiQ Socket 3
RS232
RxD
TxD
M 0V
DC 24V
On8/Off1
Line Module 3
Off 3
Total drives
Ready to power-up
Line Module 3
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram
SIEMENS
Copyright (C) SIEMENS AG 2003 All Rights Reserved

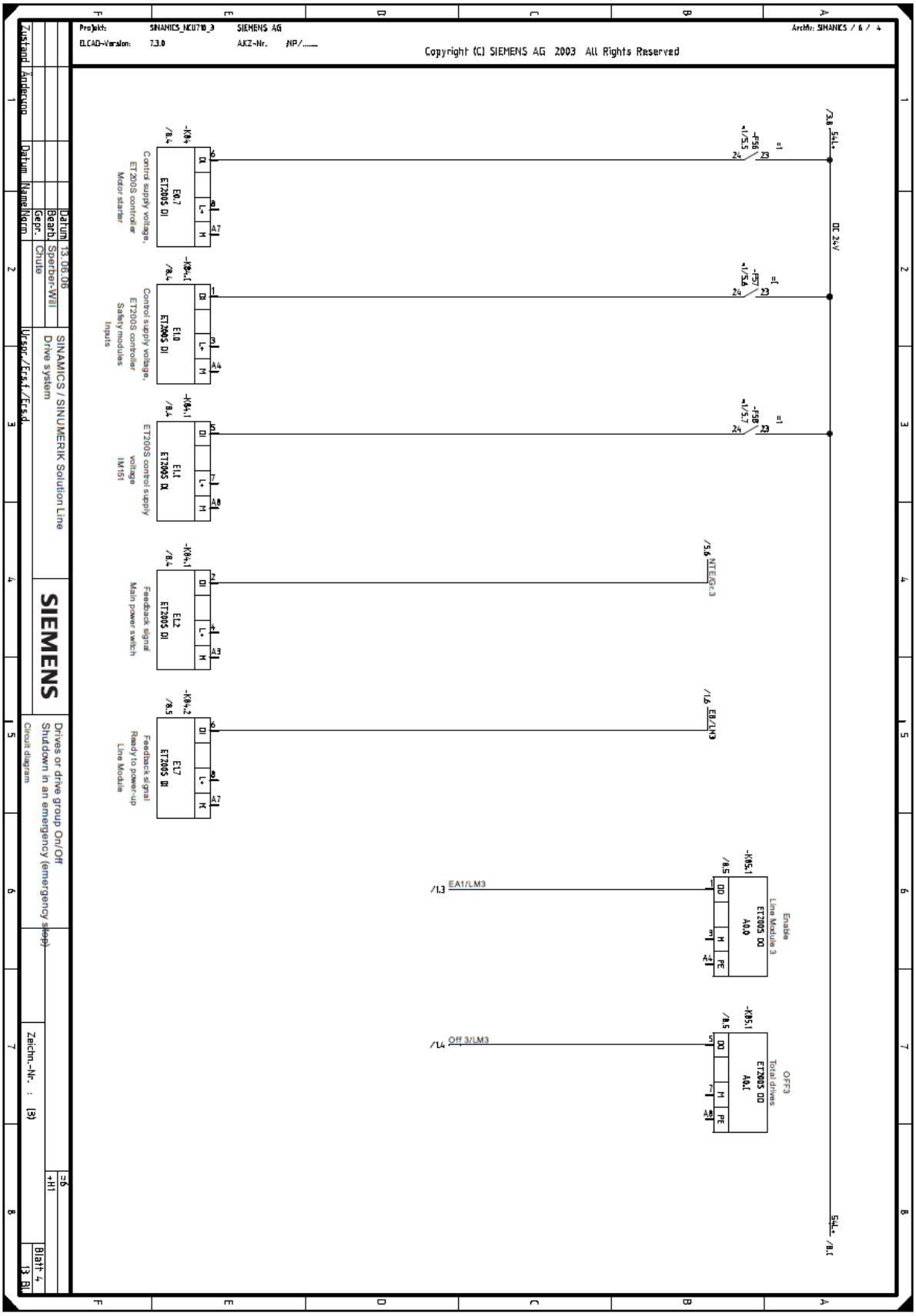
Notes :


Projekt: SINAMICS_NCU710_3
SIEMENS AG
ELCAD-Version: 7.3.0
AKZ-Nr. /NP/_____
Archiv: SINAMICS / 6 / 2
Copyright (C) SIEMENS AG 2003 All Rights Reserved
NCU 710
+K11
-X124
PROFIBUS
3)
-X136
Handwheel, hand-held terminal
-X120
TCU, PCU, HT, handwheel
-X150
Ethernet
Factory network
-X127
Programming device
-X125
USB
-X135
7-Segment display
SIM/ NCK
PLC
Startup and mode selector switch
-X22
ON
OFF
/8.1 PB3
2)
1) PROFIBUS plug 6SE7972... see IKPI catalog
Bus terminating resistor must be switched on for the first and last node in a bus segment, because otherwise the data transmission does not operate correctly. The cable shield must be placed with a large area on both sides.
2) Bus cable for PROFIBUS 6XV1830.. see IKPI catalog
3) PROFIBUS address will be set during the configuring
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram
Datum 13.06.06
Bearb. Sperber-Will
Gepr. Chute
Urspr./Ers.f./Ers.d.
Zeichn.-Nr. : 13
Blatt 2
13 Bl
=6
+H1

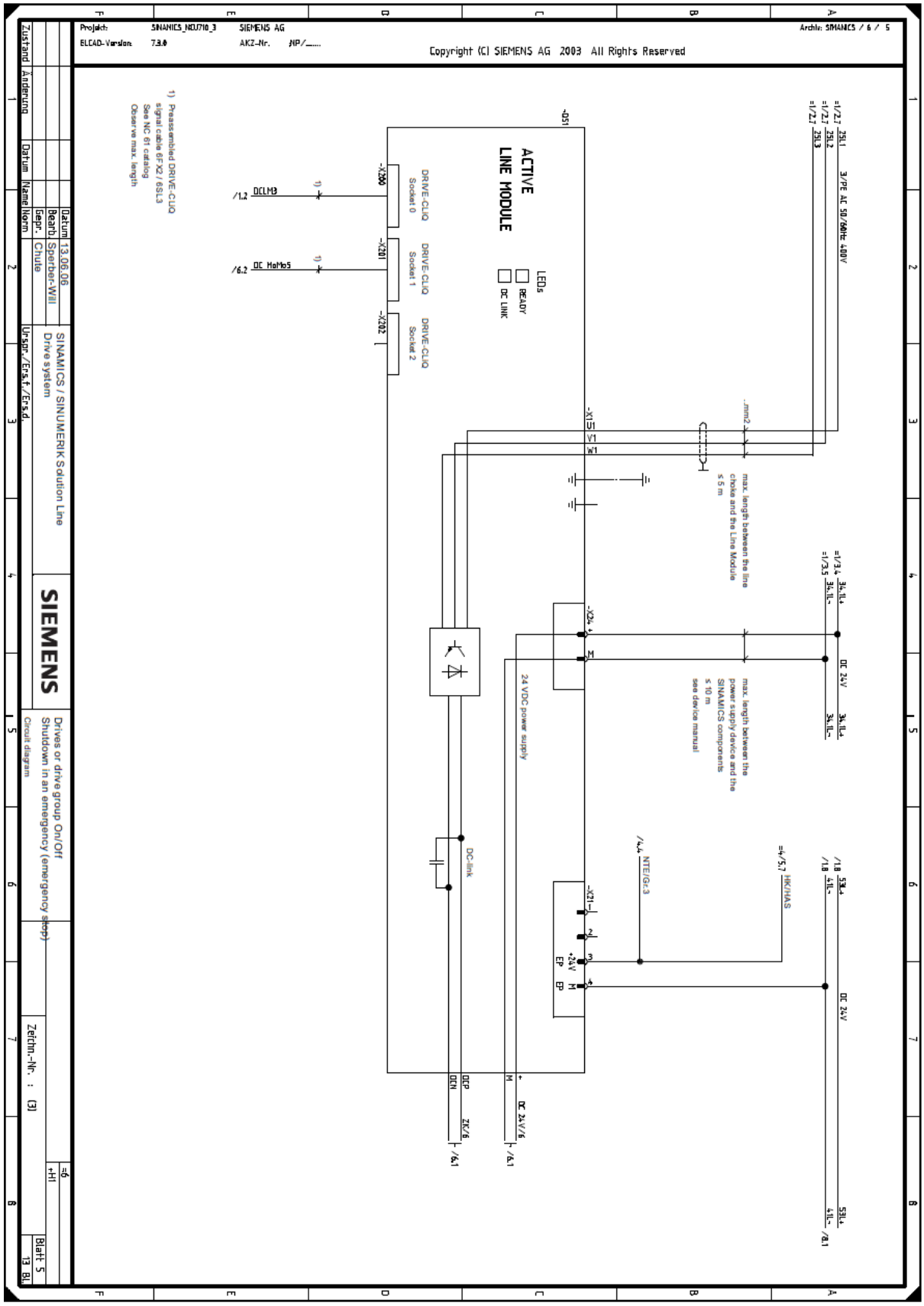
Notes :


Projekt: SINAMICS_NCU710_3
SIEMENS AG
ELCAD-Version: 7.3.0
AKZ-Nr.
Copyright (C) SIEMENS AG 2003 All Rights Reserved
SINAMICS / SINUMERIK Solution Line
Drive system
SIEMENS
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram
Datum 13.06.06
Bearb. Sperber-Willi
Gepr. Chute
Zeichn.-Nr. : (3)
Blatt 3
13_01
DC 24V
Protection Line Module 3
PS 24 VDC drive line 3
PS 24 VDC general
Load voltage NCU 710 Drive line 3
Control supply voltage Drive line 3
Control supply voltage ET200S controller Inputs
Control supply voltage ET200S controller Outputs

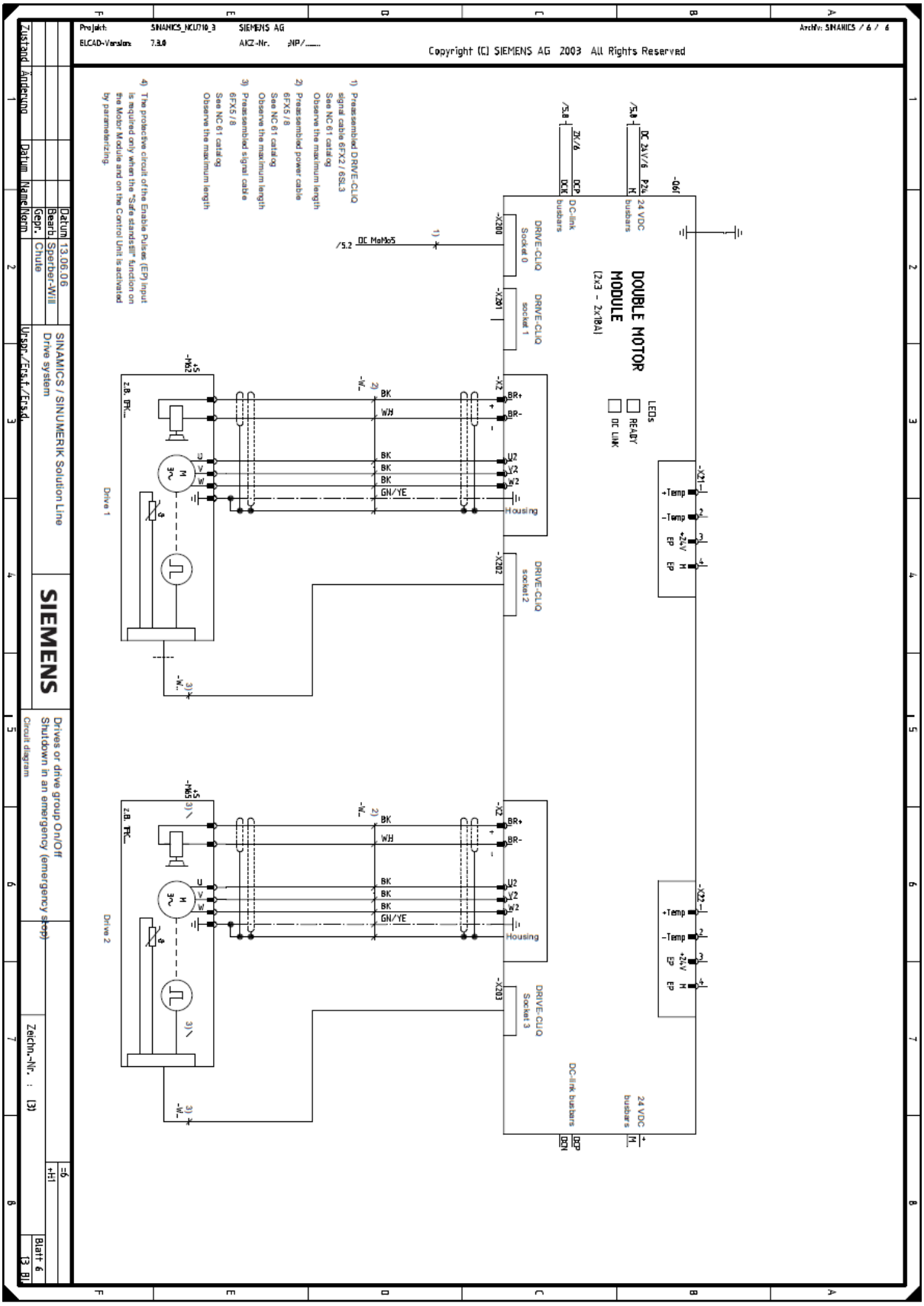
Notes :


Copyright (C) SIEMENS AG 2003 All Rights Reserved
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram
Control supply voltage, ET200S controller Motor starter
Control supply voltage, ET200S controller Safety modules Inputs
ET200S control supply voltage IM151
Feedback signal Main power switch
Feedback signal Ready to power-up Line Module
Enable Line Module 3
OFF3 Total drives
DC 24V
EA1/LM3
Off 3/LM3
NTEAG+3
EB/LM3

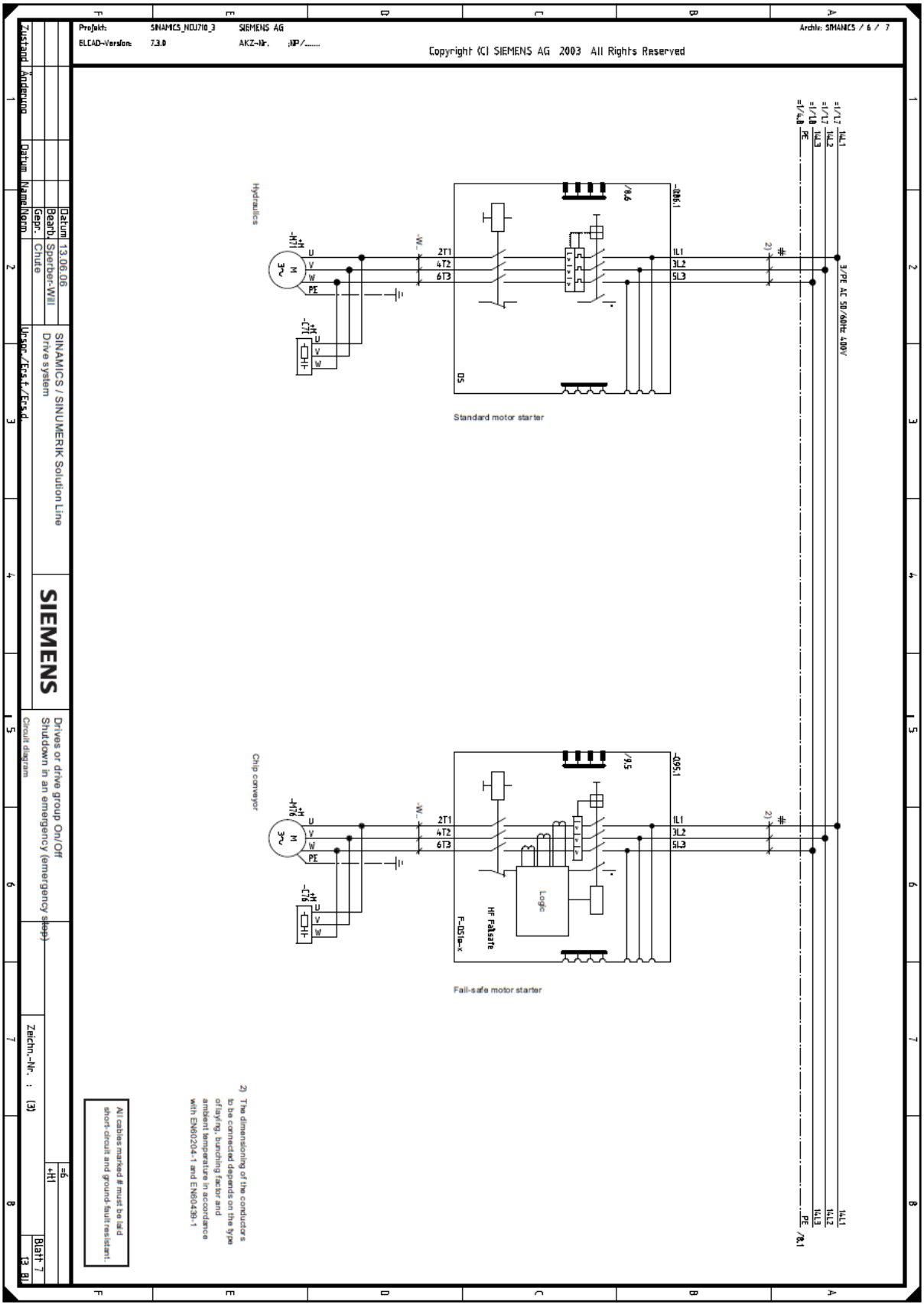
Notes :


Copyright (C) SIEMENS AG 2003 All Rights Reserved
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
ACTIVE
LINE MODULE
LEDs
READY
DC LINK
DRIVE-CLiQ
Socket 0
Socket 1
Socket 2
24 VDC power supply
DC-link
max. length between the line choke and the Line Module ≤ 5 m
max. length between the power supply device and the SINAMICS components ≤ 10 m see device manual
1) Preassembled DRIVE-CLiQ signal cable 6FX2 / 6SL3 See NC 61 catalog Observe max. length

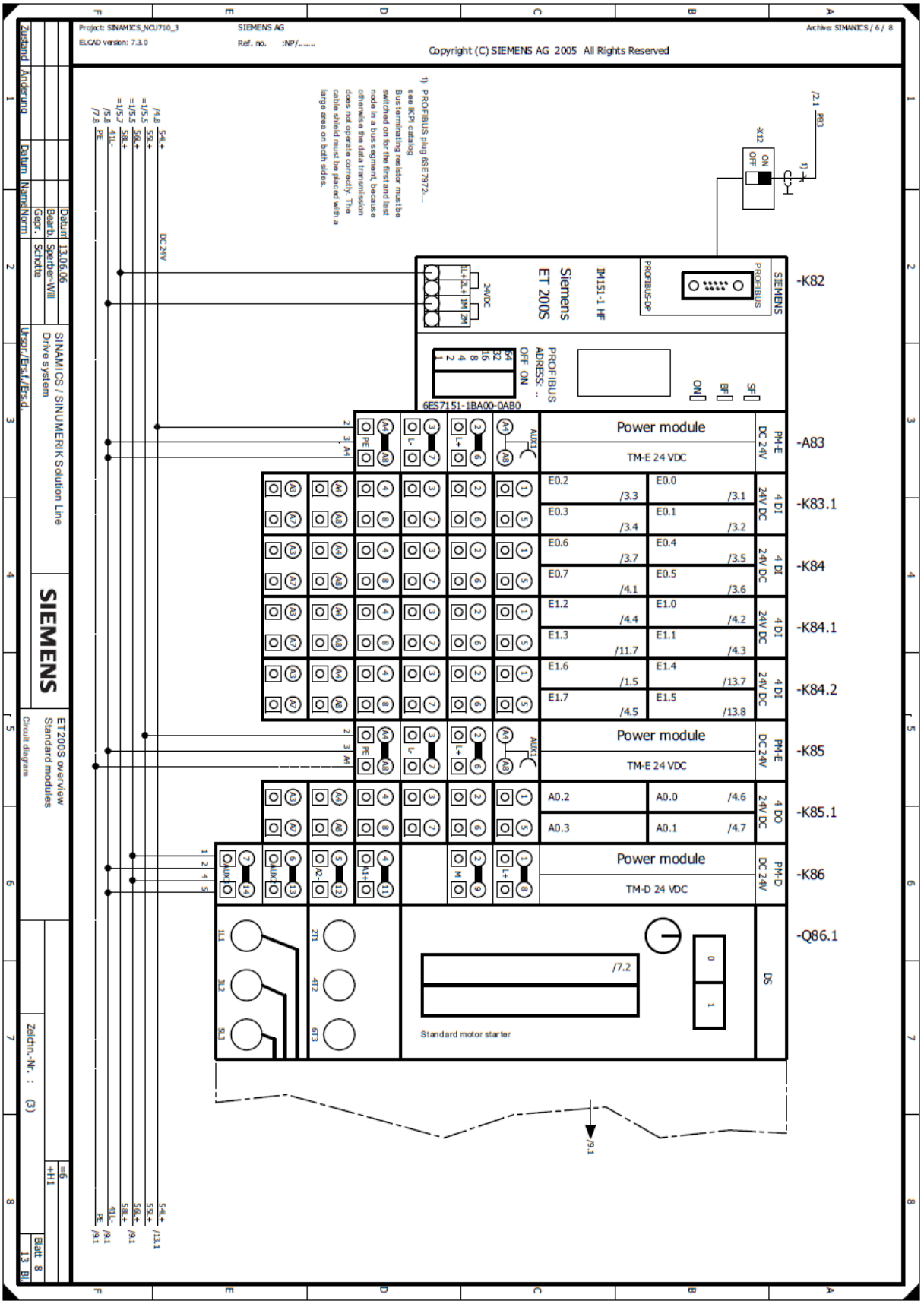
Notes :


Projekt: SINAMICS_NCU710_3
SIEMENS AG
ELCAD-Version: 7.3.0
AKZ-Nr.
Copyright (C) SIEMENS AG 2003 All Rights Reserved
DOUBLE MOTOR MODULE (2x3 - 2x18A)
DRIVE-CLiQ Socket 0
DRIVE-CLiQ socket 1
DRIVE-CLiQ socket 2
DRIVE-CLiQ Socket 3
DC-link busbars
24 VDC busbars
LEDs
READY
DC LINK
Housing
Drive 1
Drive 2
SINAMICS / SINUMERIK Solution Line
Drive system
SIEMENS
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram

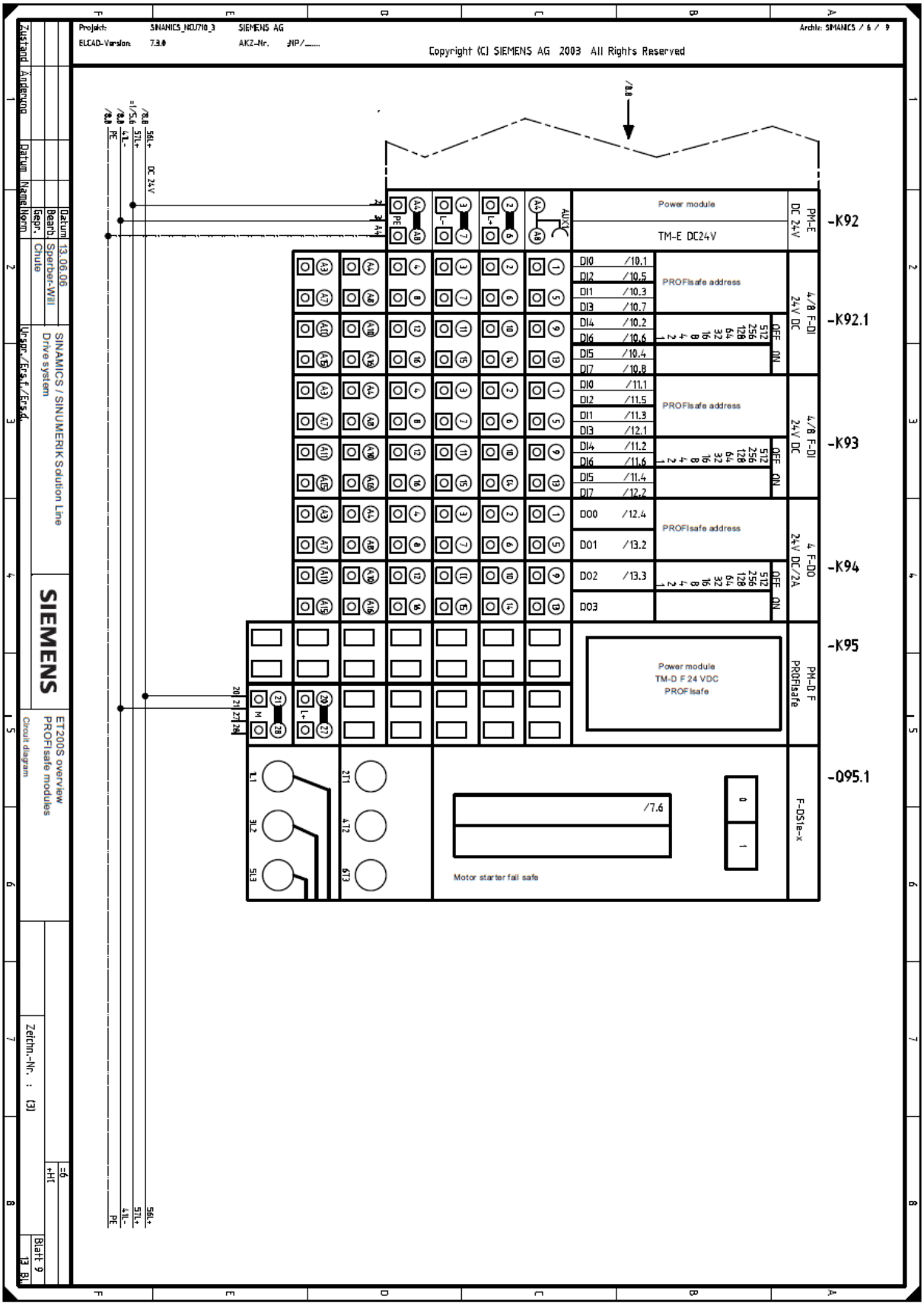
Notes :


Copyright (C) SIEMENS AG 2003 All Rights Reserved
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram
Standard motor starter
Fail-safe motor starter
Hydraulics
Chip conveyor
Logic
HF Failsafe
All cables marked # must be laid short-circuit and ground-fault-resistant.
2) The dimensioning of the conductors to be connected depends on the type of laying, bunching factor and ambient temperature in accordance with EN60204-1 and EN60439-1

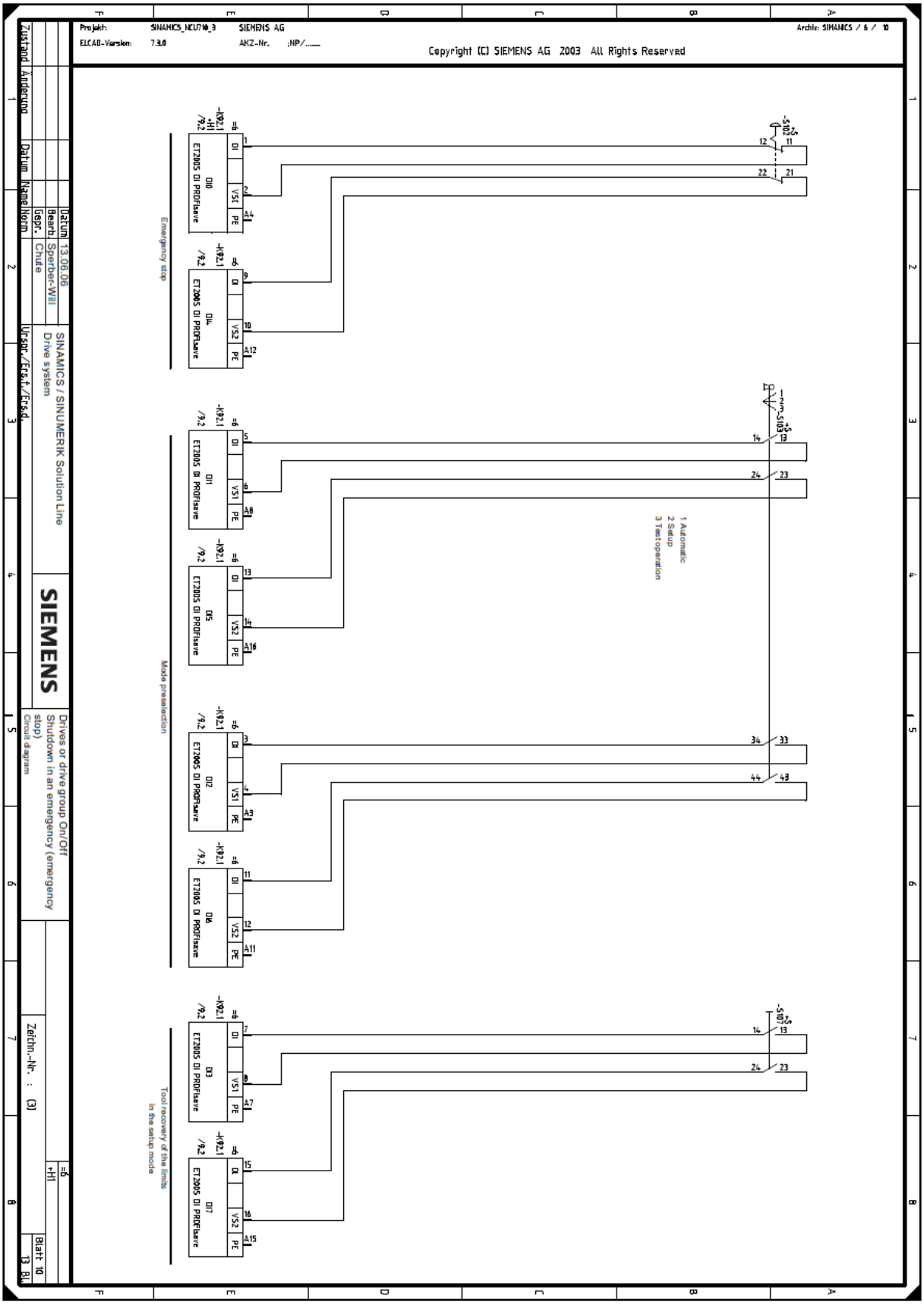
Notes :


Project: SINAMICS_NCU710_3
SIEMENS AG
Archive: SIMANICS / 6 / 8
ELCAD version: 7.3.0
Ref. no. :NP/.....
Copyright (C) SIEMENS AG 2005 All Rights Reserved
1) PROFIBUS plug 6SE7972-...
see IK PI catalog
Bus terminating resistor must be switched on for the first and last node in a bus segment, because otherwise the data transmission does not operate correctly. The cable shield must be placed with a large area on both sides.
-K82
IM151-1 HF
Siemens ET 200S
PROFIBUS-DP
PROFIBUS
6ES7151-1BA00-0AB0
PROFIBUS ADRESS
ON
BF
SF
-A83
Power module
TM-E 24 VDC
PM-E DC 24V
-K83.1
-K84
-K84.1
-K84.2
4 DI 24V DC
-K85
-K85.1
4 DO 24V DC
-K86
Power module
TM-D 24 VDC
PM-D DC 24V
-Q86.1
Standard motor starter
DS
DC 24V
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
ET200S overview
Standard modules
Circuit diagram
Zeichn.-Nr. : (3)
Blatt 8
13 Bl

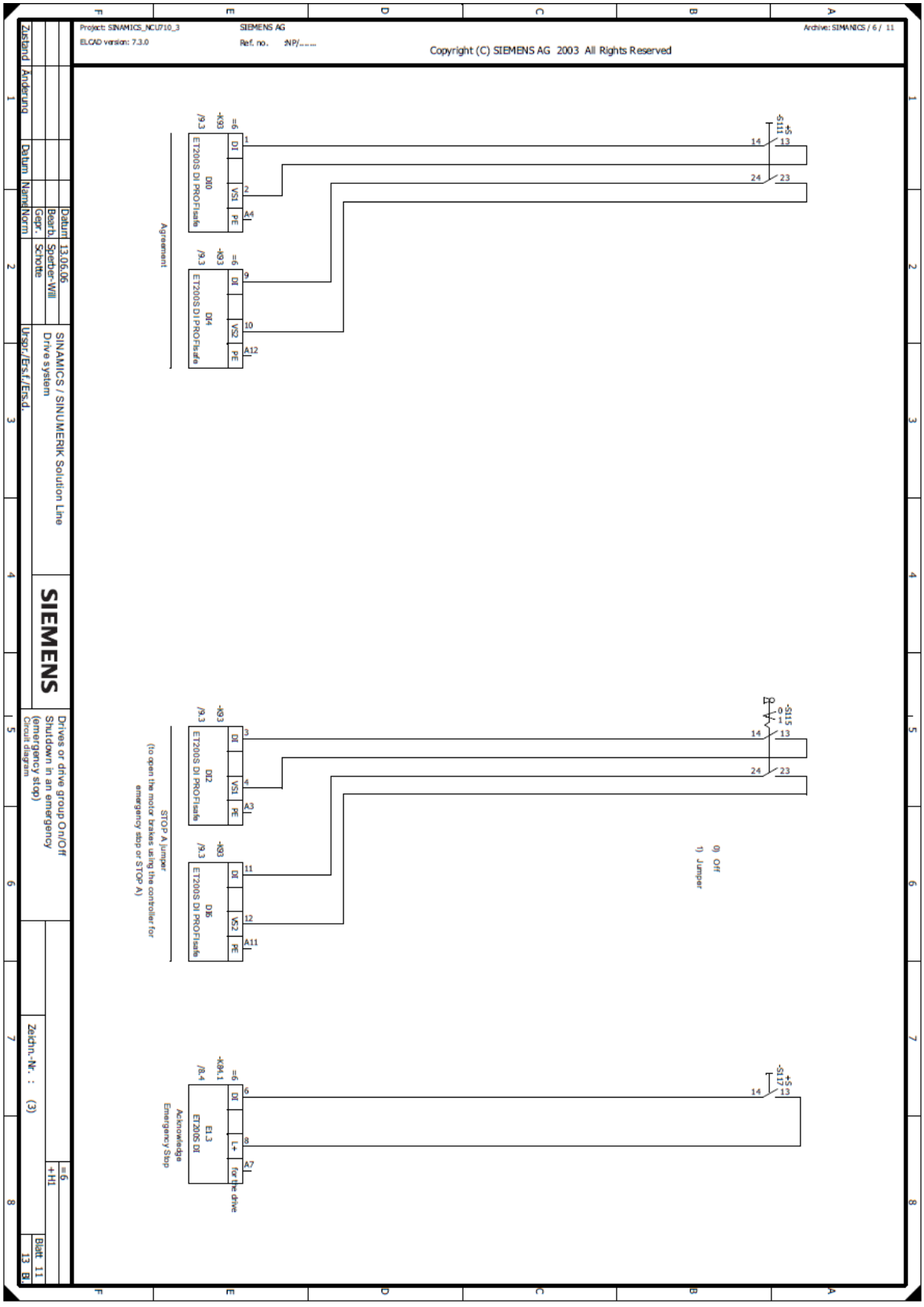
Notes :


Copyright (C) SIEMENS AG 2003 All Rights Reserved
Power module
TM-E DC24V
PM-E DC 24V
-K92
4/8 F-DI 24V DC
-K92.1
-K93
PROFIsafe address
4 F-DO 24V DC/2A
-K94
Power module
TM-D F 24 VDC
PROFIsafe
PM-D F PROFIsafe
-K95
Motor starter fail safe
F-DSte-x
-Q95.1
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
ET200S overview
PROFIsafe modules
Circuit diagram

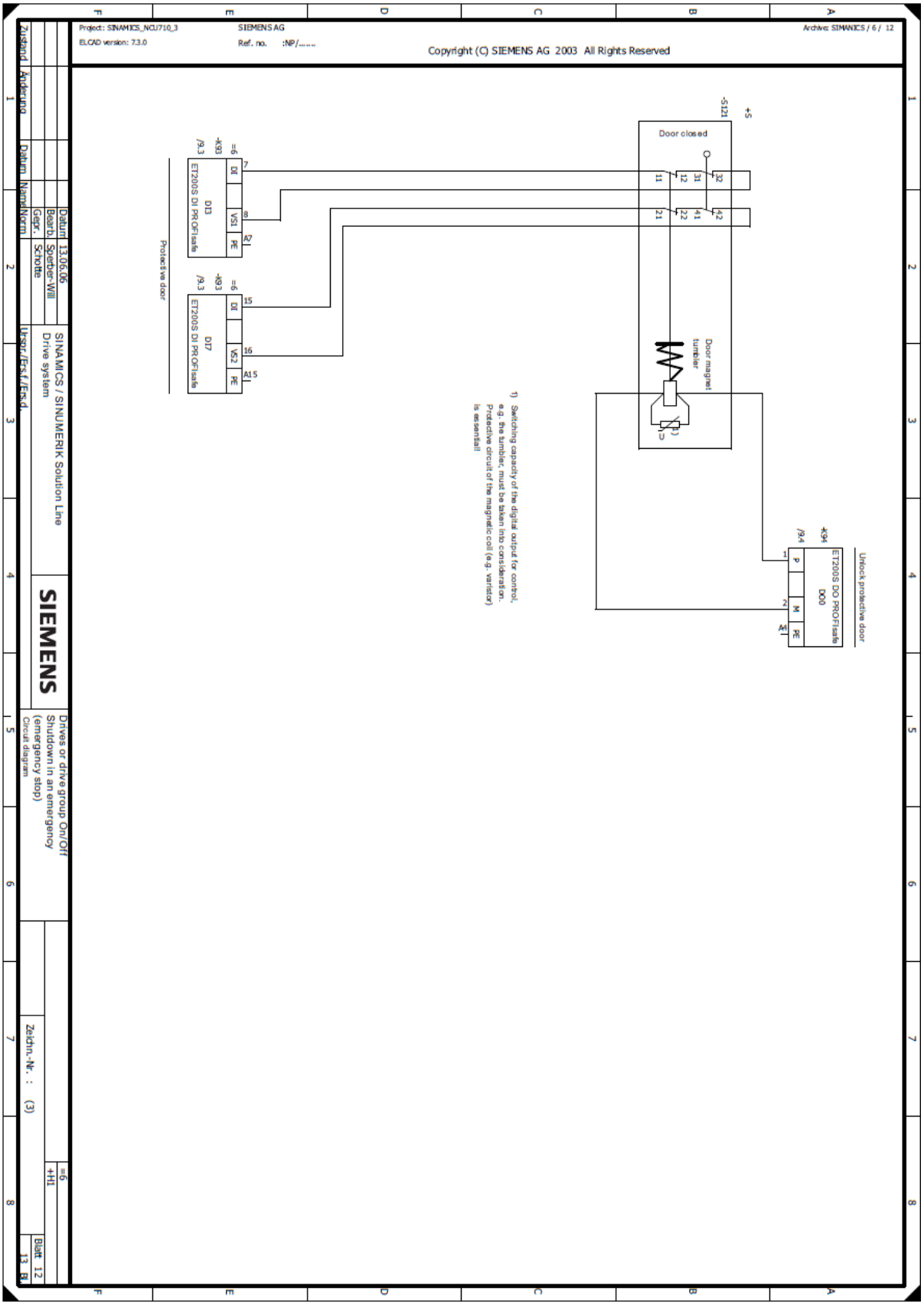
Notes :


Projekt: SINAMICS_NCU710.3 SIEMENS AG
ELCAD-Version: 7.3.0 AKZ-Nr. ;NP/.....
Archiv: SINAMICS / 6 / 10
Copyright (C) SIEMENS AG 2003 All Rights Reserved
Zustand Änderung Datum Name Norm
Datum 13.06.06
Bearb. Sperber-Will
Gepr. Chute
SINAMICS / SINUMERIK Solution Line
Drive system
SIEMENS
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram
Ursp./Ers.f./Ers.d.
Zeichn.-Nr. : (3)
=6 +H1
Blatt 10
13 Bl.
Emergency stop
Mode preselection
Tool recovery of the limits in the setup mode
1 Automatic
2 Setup
3 Testoperation
-S102
-S103
-S107
-K82.1 /9.2
ET200S DI PROFIsave
DI0
DI1
DI2
DI3
DI4
DI5
DI6
DI7
VS1
VS2
PE
A1
A2
A3
A4
A5
A6
A7
A8
A11
A12
A15
A16

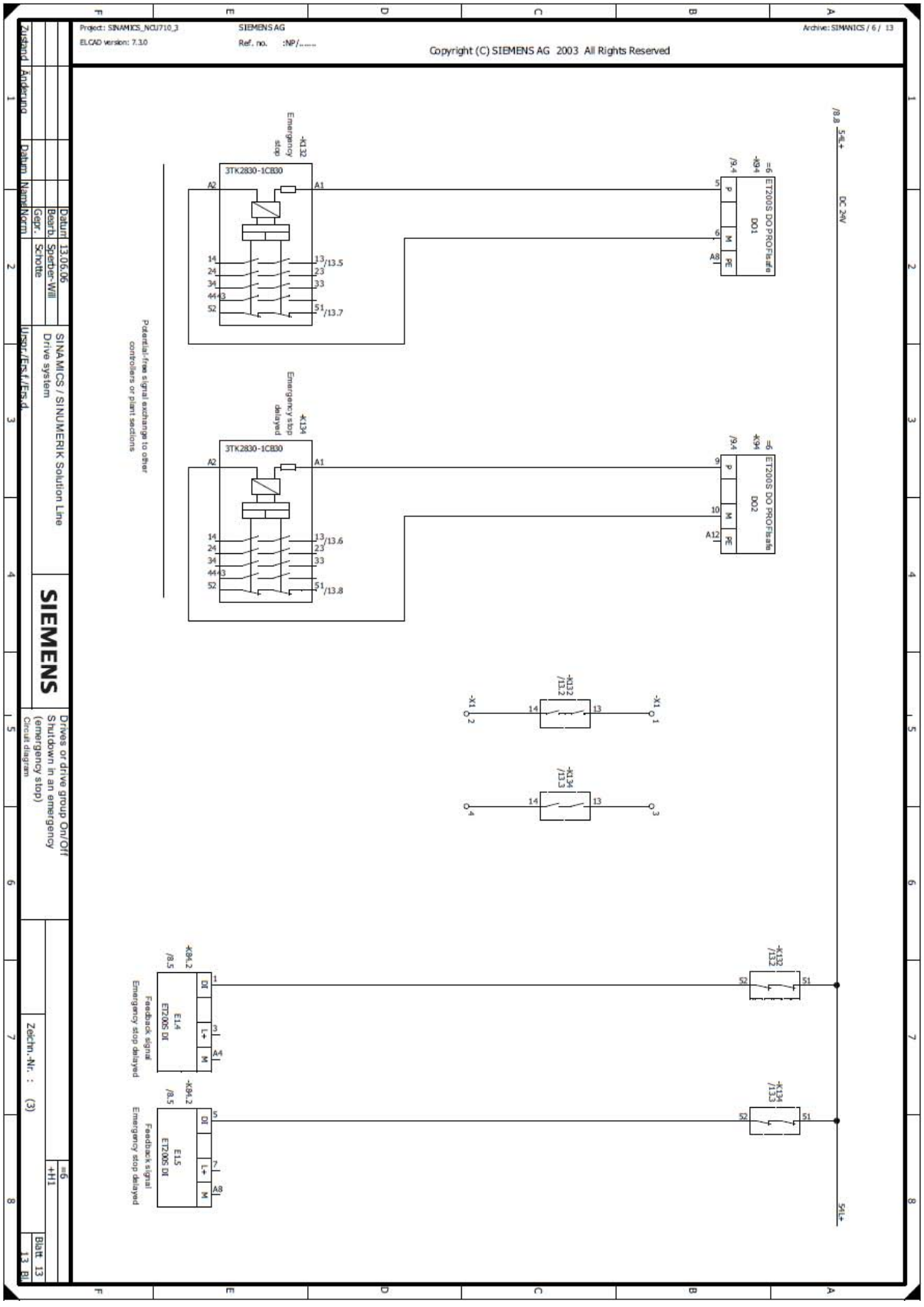
Notes :


Project: SINAMICS_NCU710_3
SIEMENS AG
Archive: SIMANICS / 6 / 11
ELCAD version: 7.3.0
Ref. no. :NP/.....
Copyright (C) SIEMENS AG 2003 All Rights Reserved
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency
(emergency stop)
Circuit diagram
Agreement
STOP A jumper
(to open the motor brakes using the controller for
emergency stop or STOP A)
0) Off
1) Jumper
Acknowledge
Emergency Stop
for the drive
Zeichn.-Nr. : (3)
Blatt 11
13 Bl

Notes :


Project: SINAMICS_NCU710_3
SIEMENS AG
Archive: SINAMICS / 6 / 12
ELCAD version: 7.3.0
Ref. no. :NP/.....
Copyright (C) SIEMENS AG 2003 All Rights Reserved
Door closed
-S121
+S
Door magnet tumbler
-Q3
1)
-K93
/9.3
=6
ET200S DI PROFIsafe
DI3
DI7
-K94
/9.4
ET200S DO PROFIsafe
DO0
Protective door
Unlock protective door
1) Switching capacity of the digital output for control, e.g. the tumbler, must be taken into consideration. Protective circuit of the magnetic coil (e.g. varistor) is essential!
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency
(emergency stop)
Circuit diagram
Zeichn.-Nr. : (3)
+H1
Blatt 12
13

Notes :


Project: SINAMICS_NCU710_3
SIEMENS AG
Archive: SIMANICS / 6 / 13
ELCAD version: 7.3.0
Ref. no. :NP/.....
Copyright (C) SIEMENS AG 2003 All Rights Reserved
Emergency stop
-K132
3TK2830-1CB30
Emergency stop delayed
-K134
Potential-free signal exchange to other controllers or plant sections
SIEMENS
SINAMICS / SINUMERIK Solution Line
Drive system
Drives or drive group On/Off
Shutdown in an emergency (emergency stop)
Circuit diagram
Feedback signal Emergency stop delayed
DC 24V
Blatt 13

Notes :

OEM Documentation (Function Table)

- OEM Documentation
- Function Table example
- Description of Safety Related Devices example

Notes :

OEM Documentation

It is the responsibility of the OEM to provide documentation relating to the functional safety requirements of the machine.

This documentation should describe the interaction of the safety related devices fitted to the machine with the system based safety functions of the Sinumerik 840D.

This should be in the form of a 'Function Table'.

The documentation should also include a description of each safety related device (Emergency Stop Push Buttons, Enable Buttons/Key-switches, Guard Door Switches….etc). It should describe how that device is to be used.

Electrical schematics showing the connection for each safety related device with the safe I/O should also be provided.

Prior to commissioning System Based safety Integrated, the OEM must provide documentation to describe the following;

- Function Table
- Description of Safety Related Devices
- Electrical schematics of safety related device connections to safe I/O.

It is also useful to have a list of the Safe Inputs and Outputs assigned to the F-Modules;

4 F-DI Module

$A_INSE[1] Emergency Stop from HT2
$A_INSE[2] Emergency Stop Reset Push Button
$A_INSE[3] Operator Door Guard Switch
$A_INSE[4] HT2 Enable Button

4 F-DI/3 F-DO Module
Inputs

$A_INSE[9] End Limit Override Keyswitch
$A_INSE[10] Setup Mode Keyswitch
$A_INSE[11] Mode 3 Keyswitch
$A_INSE[12] Spare

Outputs

$A_OUTSE[1] Output for KA1 & KA2 Contactors
$A_OUTSE[2] Spare
$A_OUTSE[3] Spare

Example of a Function Table

This table can be entered in section 2.1 of the SIAT (Safety Integrated Acceptance Test) document.
(This will be covered later in this document)

Operating mode	*Protective doors*	*Axis / spindle*	*Monitoring*
Production	Closed	X1, Y1, A1, C1 SP1	SG4 (SV4) SG4 (SV4)
	Released	X1, Y1, A1, C1 SP1	SBH (SOS) SBH (SOS)
Setup	Closed	X1, Y1, A1, C1 SP1	SG2 (SV2) SG2 (SV2)
	Released	X1, Y1, A1, C1 SP1	SG1 (SV1) SBH (SOS)/ SG1 (SV1) With HT2 Enable

Notes :

Example Description of Safety Related Devices

This table can be entered in section 2.3 of the SIAT (Safety Integrated Acceptance Test) document.
(This will be covered later in this document)

Emergency Stop from HT2 ($A_INSE[1]):

A Stop C for X1, Y1, A1, C1 and Spindle is activated via the SPL using the EMERGENCY STOP signal and the external brakes and the SH (Safe Satndstill) are then activated via STOP A. For an EMERGENCY STOP hydraulic pump, cooling-medium pump, chip conveyor and tool revolver are safely stopped, if safety-relevant via SGAs (Safety Related Outputs) of the SPL.

Emergency Stop Reset Push Button ($A_INSE[2]):

The two-channel button is used to acknowledge the emergency stop and panel monitoring function (power is switched-in).

Operator Door Guard Switch ($A_INSE[3]):

The operator door has a protective door switch. The protective door switch outputs the two-channel signal “doors closed and interlocked”. The safety functions are changed-over and selected in accordance with the previous table.

HT2 Enable Button ($A_INSE[4]):
The Two-Channel enable button is used to allow the spindle to be run at reduced velocity with the operator door open.

End Limit Override Keyswitch ($A_INSE[9]):
The Two-Channel Spring-return Key-switch selects SE2 and allows the axes to be jogged clear of the end limits for SE1.

Setup Mode Keyswitch ($A_INSE[10]):

The “production” and “setting-up” modes are selected using the mode selector switch. The key-operated switch has two contact levels. The safety functions are changed-over and selected in accordance with the previous table.

Output for KA1 & KA2 Contactors ($A_OUTSE[1]):
KA1 and KA2 are non-positively guided relays driven in parallel by safe a output. They are energized as long as the SPL SIRELAY is reset. A normally closed check-back circuit is connected to I36.0 and is monitored by the PLC test stop routine for correct operation of the contactors.

Test Stop:
The test stop routine is executed every 8hours, following complete power-off-on and also manually at any time with MCP User key 1.

Notes :

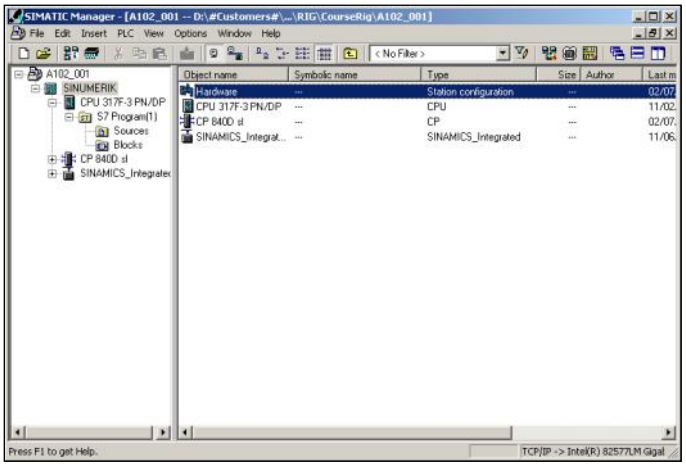
F-Controller Logic: Step7 Hardware Configuration

This section describes how to adapt the hardware configuration of Step7 so as to add the ET200s (IM151-3PN) rack to the existing ProfiNet network and also configure the Safety Related Input & Output F– Modules.

Notes :

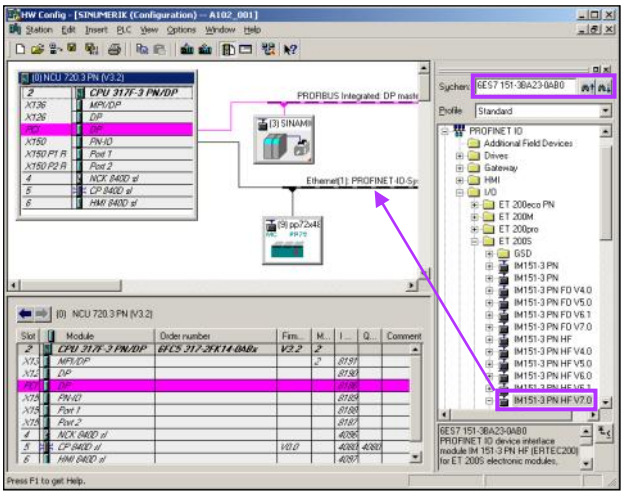
Step7 Hardware Configuration

Using the Step7 project '**A102_001**', open the Hardware Configuration.



IM151-3PN: **6ES7 151-3BA23-0AB0**

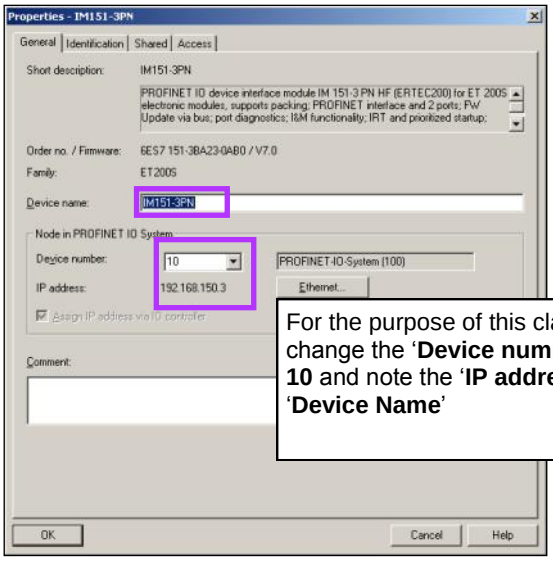
Enter the part number of the IM151-3 PN module and search…..



...or, expand the path '**PROFINET IO**' >> '**I/O**' >> '**ET200S**' and select the corresponding part number and version as stated on the actual hardware.

Once you have identified the correct module, drag the '**IM-151-3 PN HF V?.?**' to the ProfiNet bus.

Once the IM 151-3 module has been placed onto the bus, double click (or Alt+Return) to display the module properties.



For the purpose of this class, change the '**Device number**' to **10** and note the '**IP address**' and '**Device Name**'

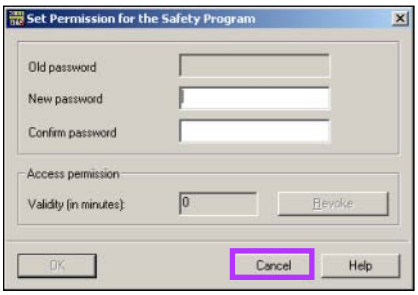
Make a note of the '**Device Name**' (This can be changed if required to suit a drawing reference for example) and IP address. We will need these later.

Click '**OK**' when finished to close the properties window.

<u>The following two pages contain information that applies only to the NCU 7x0.**3** which has 'CPU 317 F' integrated.</u>

The safety Integrated functions known from the SIMATIC F-CPU **cannot** be used in the SINUMERIK environment.

If the software package 'S7 Distributed Safety Programming' is installed within Step 7, you will be requested to enter a password allow set up of a safety program. This will always appear whenever you attempt to display or modify the safety settings in the HW Config.



Click on the 'Cancel' key to acknowledge this dialogue.

<u>Note:</u> You do not need to enter a password when using the Safety Integrated functions of the SINUMERIK 840D sl!

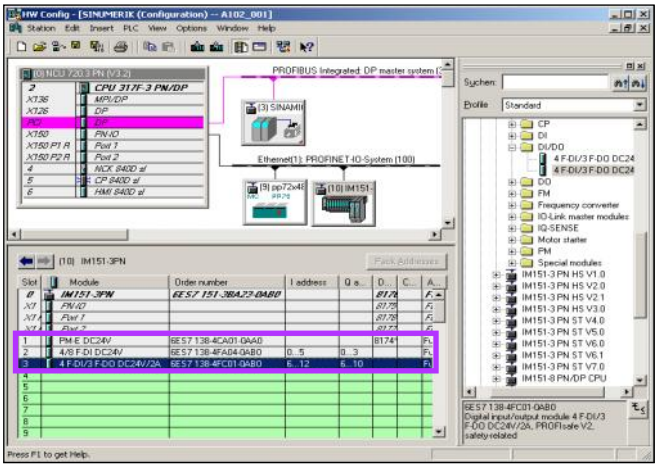
If you do assign a password here, this <u>cannot</u> be undone!

Notes :

If you have assigned a password, you must always enter this password when subsequently modifying the HW Config. Without entering the password, you have only read-only access to the parameter setting of the F-Modules.

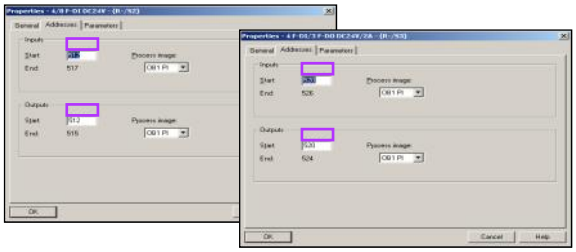
Note: Once a password has been assigned, the Step 7 addon package 'S7 Distributed Safety Programming' is required to modify the HW Config. In this case, the 'S7 F Configuration Pack' no longer suffices. You must be aware of this when sharing the Step 7 project with a colleague or customer.

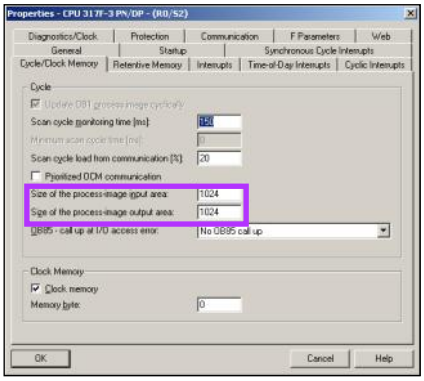
To continue with the hardware configuration. Select the correct sub folder (below the IM151-3) from the hardware catalogue and Insert the PM module into slot 1 of the IM151-3PN rack, the '4/8 F-DI' module to slot 2 and the '4 F-DI/3 F-DO' module to slot 3.



When the F-Modules are inserted, they are automatically assigned the next available I/O start addresses. Sinumerik Safety does not actually access the Safety Related I/O using these addresses. Therefore, It makes sense to re-assign the F-Modules to a higher address range, away from that used by the standard machine I/O.

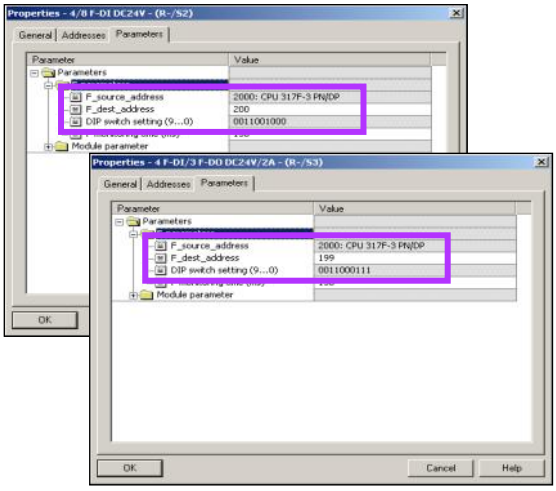
Open the 'Properties' dialog for each F-Module in turn and select the '**Addresses**' tab.
Change the start address of both the Inputs and Outputs for the '4/8F-DI' module and the '4F-DI/3F-DO' module to suitable values. **Both I/O start addresses must be the same on each F-Module.**





Note: Make sure that the selected address range is within the maximum PII & PIO (**P**rocess **I**mage of **I**nputs & **P**rocess **I**mage of **O**utputs) areas as configured within the properties page of the CPU.

Now select the '**Parameter**' tab for each module and note the '**F_source_address'**, '**F_dest_address'** and the '**DIP switch setting(9..0)'**. We will need this information later.



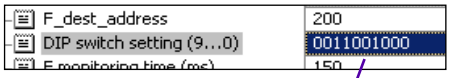
The '**F_source_address**' will be the same for all modules as this is the address of the Profibus/ProfiNet F-Master (CPU 31x).

Step7 allocates a unique '**F_dest_address**' (Destination Address) for each safe module from 200 down. (200, 199, 198...etc).

Notes :

Note: Power must be switched off for this step!

Remove each F-Module from the rack and set the DIP switches of each module in accordance with the **F_dest_address / DIP switch setting(9..0)** value.



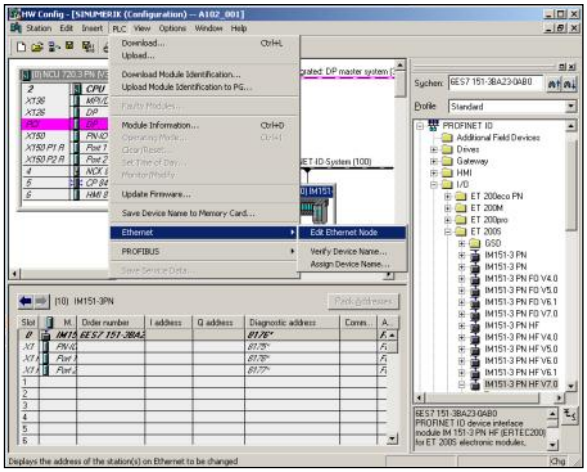
Once the DIP switches have been set correctly and the F-Module inserted back into the rack, power can once again be applied.

The following applies to ProfiNet I/O only.

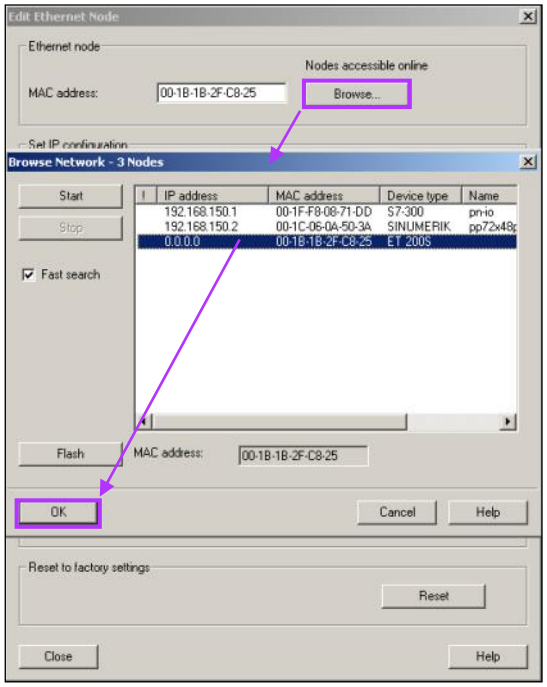
Some ProfiNet devices (including the IM151-3PN) do not have address switches to identify them on the network. These devices need to be programmed with a '**Device Name**' and '**IP Address**'.

To program the IM151-3PN in this case, select the menu '**PLC >> Ethernet >> Edit Ethernet Node**'.

Note: Ensure that the PG/PC is connected to **X150** on the NCU and the PG/PC is set to a DHCP assigned IP address.

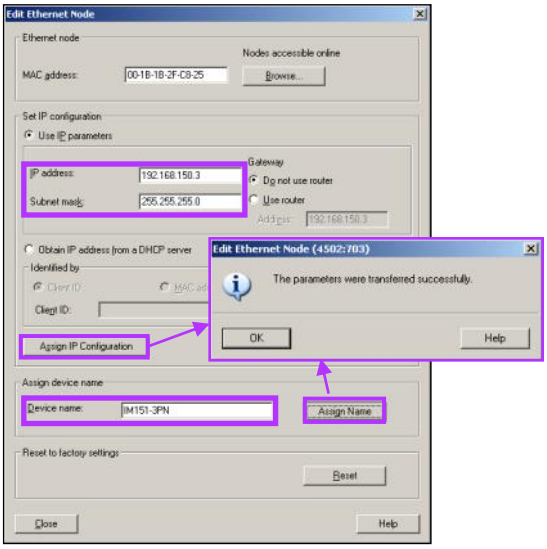


Clicking on the 'Browse…' button will display the devices available on the ProfiNet network.



Select the ET 200S device and click 'OK'.

Enter the '**IP address'**, '**Subnet mask'** and '**Device name'** that we noted earlier.



Now click '**Assign IP Configuration**' then '**Assign Name**'. After each step you should see the dialogue box as shown here.

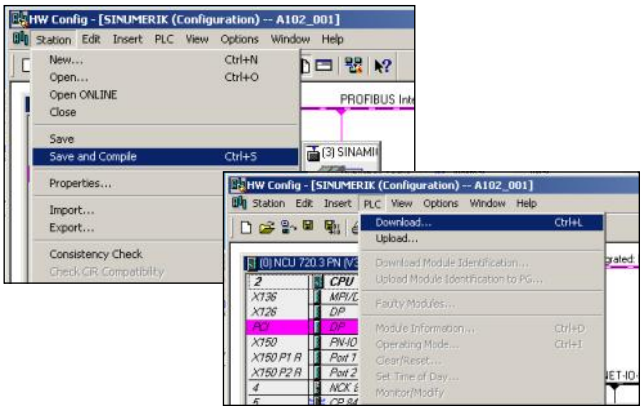
We have now successfully programmed the IM151-3PN device so that it can be recognized on the ProfiNet network. Close the 'Edit Ethernet Node' Dialogue.

Notes :

The hardware configuration can now be compiled and downloaded to the PLC in the normal way.

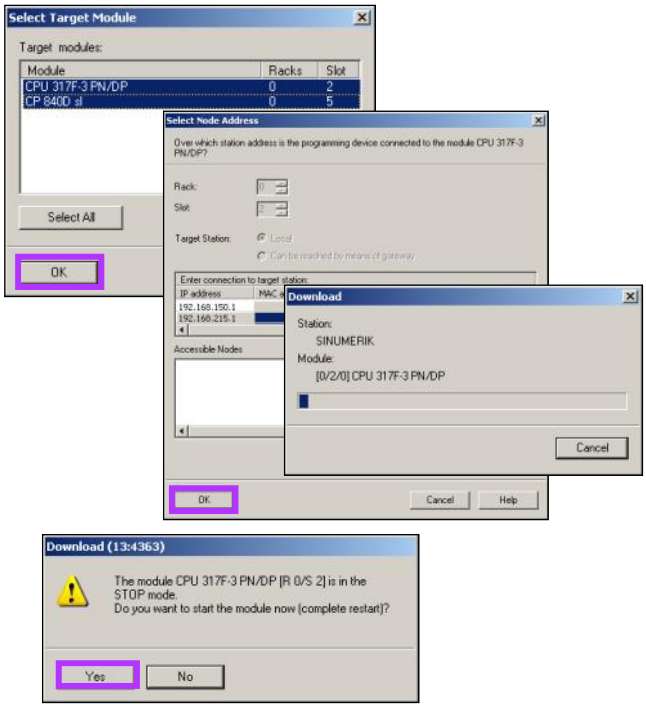
Note: Ensure that the PG/PC is connected to **X127** on the NCU and the PG/PC is set to a DHCP assigned IP address.

Use the menus…….



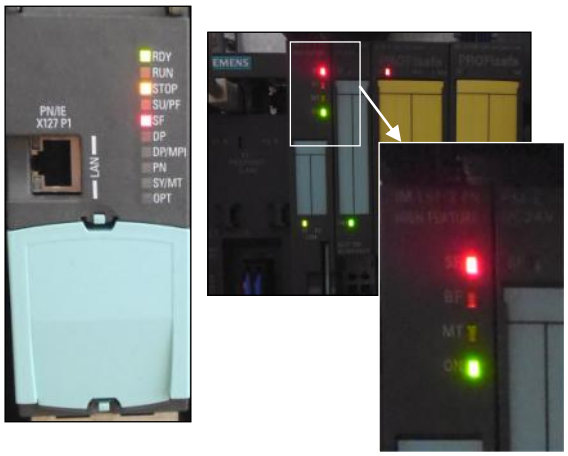
…..or the toolbar Icons.

Click through the usual dialog boxes to complete the download and restart of the PLC-CPU.



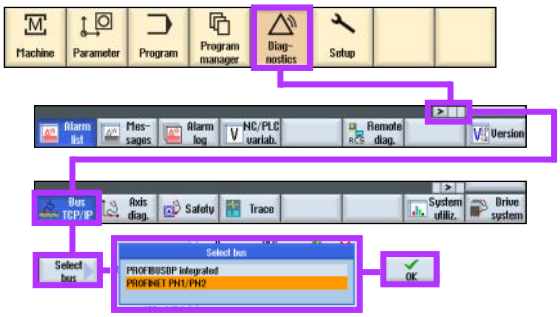
After downloading of the hardware configuration, the PLC should be in STOP mode and the SF LED will be illuminated.

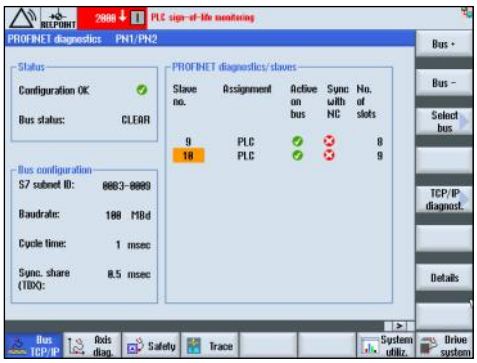
At this stage, the PN and DP LED's on the NCU should **not** be illuminated. Also, the BF LED on the IM151-3 will also be off. This means that the IM151-3 is correctly configured and recognized on the bus.



Note the 'SF' LED on the NCU, IM151-3 & 4/8 F-DI Module.

HMI Operate offers some very useful diagnostic screens for showing the status of ProfiBus & ProfiNet devices….



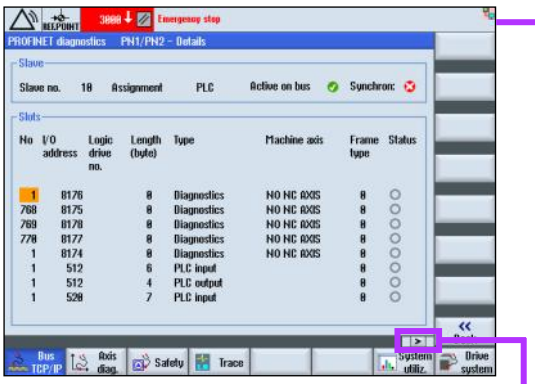


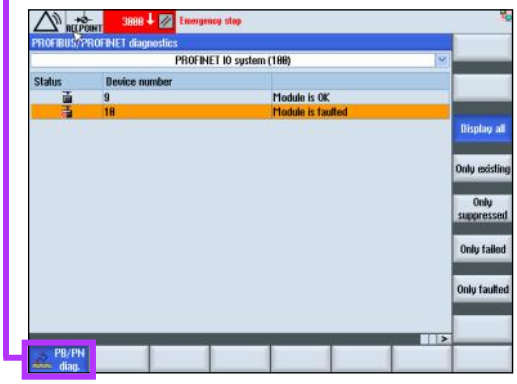
Here you can see that the two ProfiNet devices are 'Active on Bus'.

Slave 9 is the PP72/48 and **Slave 10** is the newly configured IM151-3 PN.

Notes :

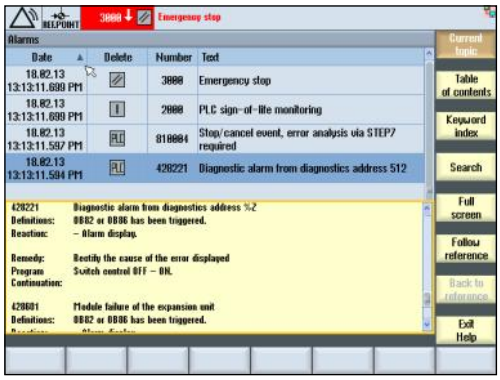
Further useful information can be seen in the details screen…..





….and the ‘PB/PN Diag’ screen.

As you can see from the previous screenshot (and the SF LED’s on the NCU, IM151-3 & 4/8 F-DI Module), there is a problem with ProfiNet device number 10. This is the ‘IM151-3 PN’ rack.
The ‘Alarm list’ will also show that there is a diagnostic error from address 512 (PLC Error 428221).

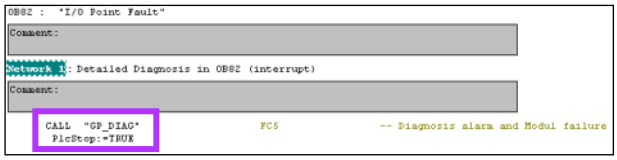


The diagnostic error has triggered a call to OB82 or OB86.

Address 512 is the I/O start address of the F-Modules that we configured earlier.

OB82 & OB86 both call FC5 (GP_DIAG) which actually generates the PLC error 428221.

Within OB82 & OB86, FC5 (GP_DIAG) is called. The default parameter value for PlcStop := TRUE.



This is the reason that the PLC is in STOP.

This can be changed to FALSE so that the PLC will no longer go into STOP mode when the system detects a diagnostic interrupt from a PN/DP device. This applies to all diagnostic interrupts and not just the interrupts from safety related PN/DP devices.

Error 428221 will still be displayed when PlcStop := FALSE.

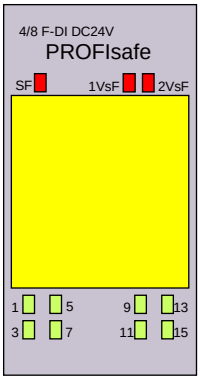
NOTE: When setting this parameter to FALSE, you must decide if it’s necessary to interrogate the header data area’s of each OB (e.g. FB125-DP_DIAG / FB126-PN_DIAG) for further information on the faulty module. An appropriate course of action can then be taken by the user PLC program.

From all of the diagnostic information we now have. It is clear that there is a problem with the 4/8 F-DI module (address 512).

The default configuration for the 4/8 F-DI module is for ‘4 Dual Channel’ inputs. The module expect both channels to be ‘Equivalent’ i.e. both channels must have the same state at all times, both channels on or both channels off.

The 4 inputs are connected to the following terminals;

Input 1——— 1 & 9
Input 2——— 5 & 13
Input 3——— 3 & 11
Input 4——— 7 & 15



In the default configuration, terminals 1, 5, 3 & 7 must be supplied by Vs1 (terminals 2,6,4 & 8). Terminals 9,13,11 & 15 must be supplied by Vs2 (terminals 10,14,12 & 16).

The status of each channel can be easily observed via the LED’s.

If you look at LED’s 5 & 13 for Input 2, you can see that only LED 13 is illuminated (Non-Equivalent). Due to the default settings of the module, this is the reason for the SF LED’s, the system call to OB82 / OB86, PLC Error 428221 and the PLC in STOP Mode due to FC5.PlcStop := TRUE.

Notes :

Press the '**Emergency Stop Reset**' button on the front of the training rig and observe the LED's 5 & 13 of the 4/8 F-DI Module. You can see them swap state so that LED 5 is now on and LED 13 is off. This is a '**Non-Equivalent**' Input.

The Safety Related Devices of the training rig are connected as follows;

4 F-DI Module

Channels 1 & 9	Emergency Stop from HT2
Channels 5 & 13	Emergency Stop Reset Push Button
Channels 3 & 11	Operator Door Guard Switch
Channels 7 & 15	HT2 Enable Button

4 F-DI/3 F-DO Module

Inputs

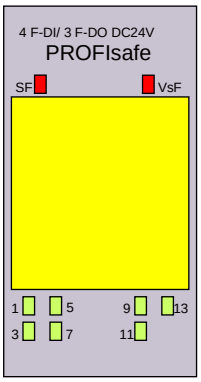
Channels 1 & 2	End Limit Override Keyswitch
Channels 5 & 6	Setup Mode Keyswitch
Channels 9 & 10	Mode 3 Keyswitch
Channels 13 & 14	Spare

Outputs

Channels 3 & 4	Output for KA1 & KA2 Contactors
Channels 7 & 8	Spare
Channels 11 & 12	Spare

Now operate each of the Safety Related Devices in turn and observe the LED's of the two safety modules.

All of the Safety Related Input Devices are 'Two Channel Equivalent'.

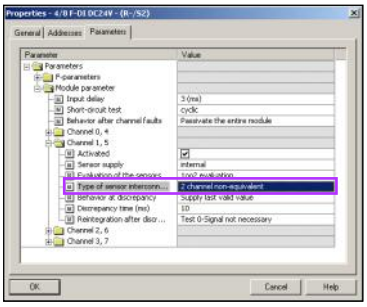


Note: The 4 F-DI/3 F-DO module has a different terminal number assignment. Also, there is only 1 VS supply (terminals 15 & 16) for both channels of each input.

For more information on this and other F-Modules, see the '**ET200S F Operating Manual**'.

We must now change the second input of the '4/8 F-DI module (Channels 1 & 5) so that the 'Type of Sensor Interconnection' is set to '2 Channel Non-Equivalent' and not the default setting of '2 Channel Equivalent'.

We can do this in the hardware configuration by opening the properties dialogue for the 4/8 F-DI module.



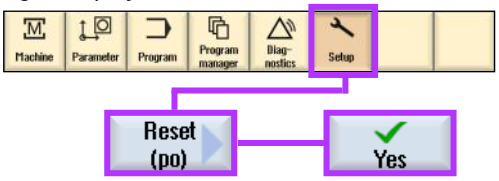
Select the 'Parameters' tab, now expand the folders 'Module Parameter' and 'Channel 1, 5'.

Now change the 'Type of Sensor Interconnection' to 'non-equivalent'.

Close the properties dialogue with 'OK' (Acknowledge the warning with 'Close' >> 'OK'), 'Save & Compile' then download the hardware configuration again.

The system will no longer report an SF error on the NCU, IM151-3 PN or the 4/8 F-DI Module.

Carry Out an NCK Reset and the PLC will RUN with no error messages displayed.



However, operating the 'Safety Related Input Devices' (Slowly) may still cause a 'Diagnostic Error' to be triggered. This will be indicated by the SF LED, PLC Error '428221 Diagnostics alarm from diagnostics address xxx' and possibly a PLC in STOP mode due to OB82 calling FC5 (GP_DIAG) if PlcStop := TRUE.

This is due to the default 'Discrepancy time' of 10ms being exceeded for a particular input channel.

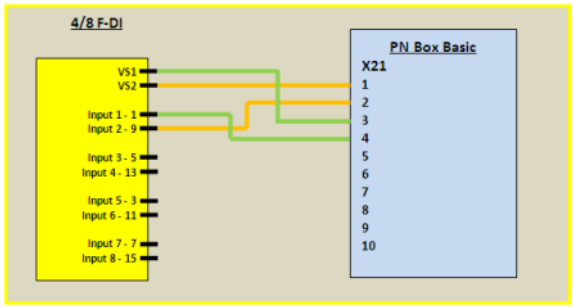
This can also be changed in the properties dialogue of the F-Module within the hardware configuration.

E.g. change the '**Discrepancy Time**' from **10ms** to **500ms** for the channel that is causing the error. This will stop the fault from appearing under normal operation.

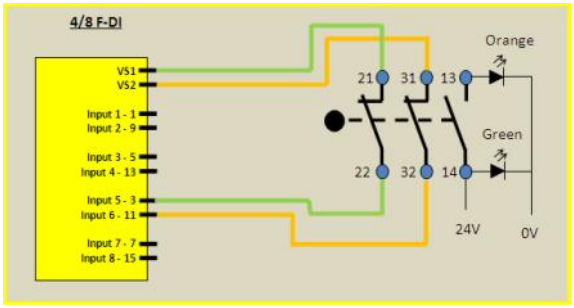
Notes :

Circuit Diagrams for the Simulator Safety Related Devices:

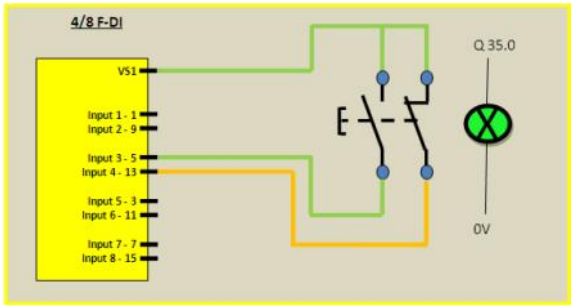
Channels 1 & 9 Emergency Stop from HT2



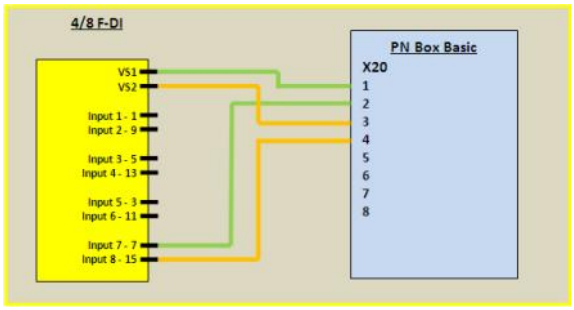
Channels 3 & 11 Operator Door Guard Switch



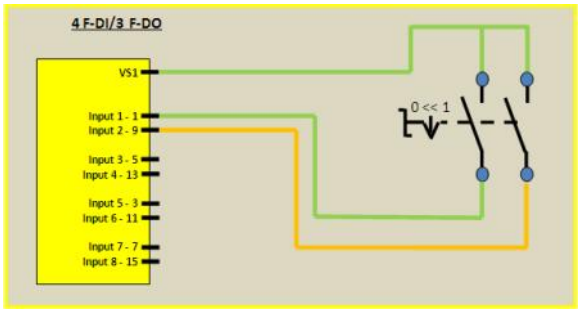
Channels 5 & 13 Emergency Stop Reset Push Button



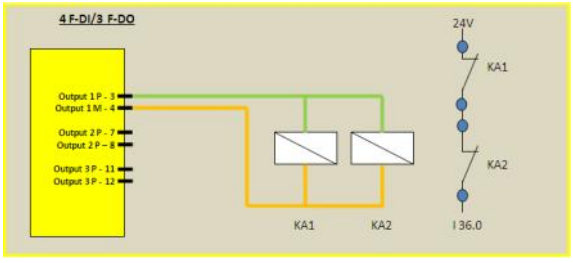
Channels 7 & 15 HT2 Enable Button



Channels 1 & 2 End Limit Override Keyswitch

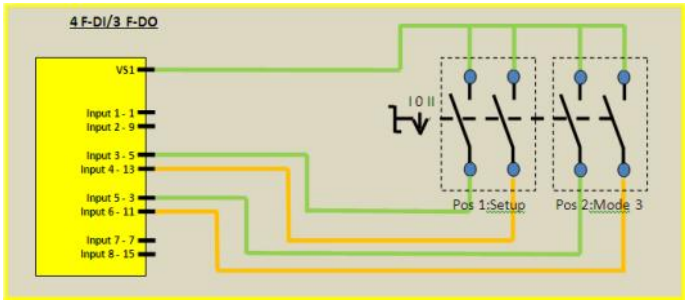


Channel 3 & 4 Output for KA1 & KA2 Contactors



Channels 7 & 8 Not Connected
Channels 11 & 12 Not Connected

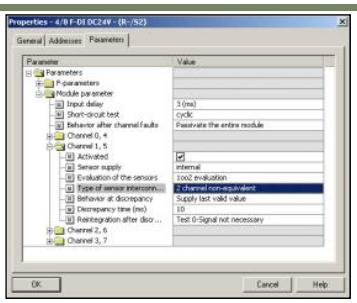
Channels 5 & 6 Setup Mode Keyswitch
Channels 9 & 10 Mode 3 Keyswitch



Channels 7 & 15 Not Connected

Notes :

It is important to understand the module and channel parameters of the F-Modules.



Input Delay:
This can be used to suppress interference for all input channels.

Short Circuit Test:
This parameter allows you to activate the short-circuit detection depending on the F-module used:

- For the entire F-module, or...
- For the channels of the F-module for which "internal sensor power supply" is set.

The short-circuit test is only useful if you are using simple switches that do not have their own power supply.

The short-circuit detection deactivates the sensor supply temporarily. The length of the deactivation period is equivalent to the configured input delay. The input delay is permanently set to 3 ms for the 4 F-DI/3 F-DO DC24V PROFIsafe electronic module and the F-switch.

If a short circuit is detected, the F-module triggers a diagnostic interrupt in the CPU.

Behavior after Channel Faults:

This parameter is used to specify whether the entire F-module is passivated or just the faulty channel(s) in the event of channel faults.

Channel Specific Parameters

Activated:
If you select this check box, the corresponding channel or channel pair is enabled for the signal processing in the safety program. If you clear the check box, an unused channel or channel pair is disabled.

Sensor Supply:
Here, you can specify whether or not the sensors are supplied via the F-module (internal sensor power supply Vs1 & Vs2). The 'short-circuit test' requires this parameter to be set to 'Internal'.

Evaluation of the Sensors:

1oo1 evaluation
In 1oo1 evaluation, there is one sensor, and it is connected to the F-module via a single channel.

1oo2 evaluation
In 1oo2 evaluation, two input channels are occupied by:

- A single-channel sensor
- A non-equivalent sensor
- A two-channel sensor
- Two single-channel sensors

The input signals are compared internally for equivalence or non-equivalence.

Note that in 1oo2 evaluation, two channels are combined into a channel pair. The number of available input channels of the F-module is reduced accordingly.

Type of Sensor Interconnection:

If "1oo2 evaluation" is specified for the sensor evaluation, you can use this parameter to select the type of sensor interconnection for each input channel:

- "2-channel equivalent" (with equivalent signals): one two-channel sensor or two single-channel sensors are connected via two channels to two opposing input channels
- "2-channel non-equivalent" (with non-equivalent signals): one non-equivalent sensor or two single-channel sensors are connected via two channels in a non-equivalent way to two opposing input channels
- "1-channel": one sensor is connected via one channel to two opposing input channels

For "1oo1 evaluation", the type of sensor interconnection is set permanently to "1-channel".

Notes :

Behavior at Discrepancy:

For the "Behavior at discrepancy", you assign the value that is provided to the safety program in the F-CPU during a discrepancy between two relevant input channels, i.e., during the ongoing discrepancy time. You can assign the behavior at discrepancy as follows:

- "Provide last valid value" or
- "Provide value 0"

Note: Only applicable if 'Sensor evaluation' is set to '1oo2 evaluation'.

Provide last valid value
As soon as a discrepancy between the signals of two relevant input channels is detected, the last valid value (old value) prior to the discrepancy is made available to the safety program in the F-CPU. This value is made available until the discrepancy disappears or until the discrepancy time expires and a discrepancy error is detected. Accordingly, the sensor-actuator response time is extended by this time.

As a result, the discrepancy time of sensors connected via 2 channels has to be tuned for fast reactions to short response times. Thus, it makes no sense, for example, if sensors connected via 2 channels with a discrepancy time of 500 ms trigger a time-critical shutdown. In the absolute worst case, the sensor -actuator response time is extended by an amount approximately equal to the discrepancy time:

- You should therefore arrange the sensors in the process with as little discrepancy as possible.
- You should then select the shortest possible discrepancy time that, at the same time, has sufficient cushion against false tripping of discrepancy errors.

Provide value 0
As soon as a discrepancy between the signals of two relevant input channels is detected, the value "0" is made available to the safety program in the F-CPU.

If you have assigned "Provide value 0", the sensor-actuator response time will not be affected by the discrepancy time.

Discrepancy Time:
Here, you can specify the discrepancy time for each channel pair. The entered value is rounded to multiples of 10 ms. When you have assigned parameters to the type of sensor interconnection as "1-channel", the time discrepancy is permanently set to 10 ms.

Requirements
You have assigned the following parameters:

- Sensor evaluation: "1oo2 evaluation" and
- Type of sensor interconnection: "2-channel equivalent" or "2-channel non-equivalent"

Discrepancy Analysis and Discrepancy Time
If you use a two-channel sensor, a non-equivalent sensor, or two single-channel sensors that measure the same physical process variables, the sensors will have a delayed response, for example, due to the limited precision of their arrangement to one another.

The discrepancy analysis for equivalence/nonequivalence is utilized with fail-safe inputs to identify errors based on the timing of the two signals of the same functionality. Discrepancy analysis is initiated when different levels are detected for two associated input signals (for nonequivalence testing: when the same levels are detected). A test is conducted to determine whether the difference in levels (when testing for non-equivalence: agreement) has disappeared after a configurable period of time. If not, this means that a discrepancy error exists. In most cases, the discrepancy time starts but does not expire because the signal differences disappear after a short time.

Select such a large discrepancy time, that when no errors are present, the difference between both signals (for non-equivalence testing: the agreement of the signals) has disappeared in every case before the discrepancy time has expired.

Behavior while Discrepancy Time is Running
While the assigned discrepancy time is running, either the last valid value or "0" is made available from the relevant input channels to the safety program in the F-CPU according to the assigned behavior at discrepancy.

Behavior After Discrepancy Time Elapses
If the input signals do not agree once the assigned discrepancy time expires, e.g., due to a break in a sensor wire, a discrepancy error is detected and the "Discrepancy error" diagnostic message containing information on which channels are faulty is generated in the diagnostic buffer of the F-module.

Notes :

Reintegration after discrepancy error:
This parameter specifies the criteria for when a discrepancy error is regarded as corrected, thus enabling reintegration of the relevant input channels.

The following parameter assignment options are available:

- "Test 0 signal required" or
- "Test 0 signal not required"

Requirements
You have assigned the following parameters:

- Sensor evaluation: "1oo2 evaluation"

Test 0 signal required
If you have assigned "Test 0 signal required," a discrepancy error is not regarded as corrected until a 0 signal is present at both of the relevant input channels.

If you are using non-equivalent sensors, i.e., "Type of sensor interconnection" is set to "2-channel non-equivalent", a 0 signal must be present at the channel supplying the useful signal.

For information on which channels of the F-module supply useful signals, refer to the manual for the F-module you are using.

Test 0 signal not required
If you have assigned "Test 0 signal not required," a discrepancy error is regarded as corrected when a discrepancy no longer exists at the two relevant input channels.

F-modules in SIMATIC S7 for which the "Reintegration after discrepancy error" parameter cannot be assigned exhibit this behavior.

Based on the information in this section, we should: **(for the purposes of this training class only)**

- Set a 'Discrepancy time of 500ms for each input channel on both F-Modules.
- Disable any unused channels (4 F-DI/3 F-DO Input Channel 13, 14 & Outputs 7, 8 & 11, 12)
- Change the FC5(GP_DIAG) parameter PlcStop := FALSE.

This completes the Step7 hardware configuration of the Safety Related F-Modules.

Notes :

F-Controller Logic: Enabling Safety in the NC

This section describes the following;

- Enable Safety Integrated within the NC
- Explanation of the Safety Related System Variables $A_INSE & $A_OUTSE
- Assign the S7 Safety I/O (previously configured) to Sinumerik Safety Related System Variables $A_INSE & $A_OUTSE
- Description of NCK-SPL and PLC-SPL
- Running the NCK-SPL & PLC-SPL at system start-up

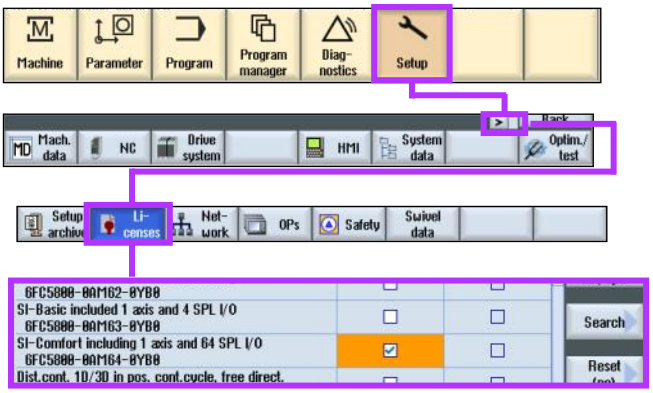
Notes :

Enable Safety Integrated within NC

Depending on the number of External Safety Related Inputs and Outputs are to be configured, the corresponding Safety Integrated option must be set.

'SI Basic'	Maximum of 4 SPL I/O
'SI Comfort'	Maximum of 64 SPL I/O
'SI High Feature'	Maximum of 192 SPL I/O (Requires Software >= 4.5)

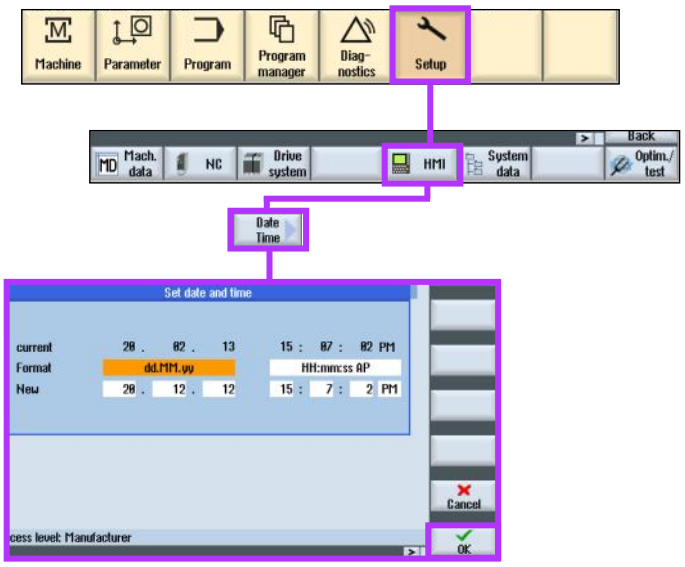
For this module, enable the '**SI Comfort'** Option.



Execute an 'Reset (PO)' to activate the option once set.

Setting the Date & Time

During the commissioning, the Checksums of Safety Related parameters are stored within the control. The information stored also includes a time & date stamp. For this reason (and good practice at any other time!) check that the time and date is set correctly within the system.



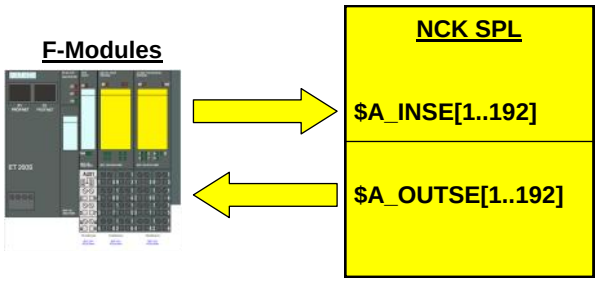
Assignment of $A_INSE & $A_OUTSE system variables to hardware

The safe input and output channels of the F-Modules configured within Step7 must be assigned to safety related system variables $A_INSE[..] & $A_OUTSE[..].

$A_INS**E**[n]	Safety Related **EXTERNAL** Input
$A_OUTS**E**[n]	Safety Related **EXTERNAL** Output

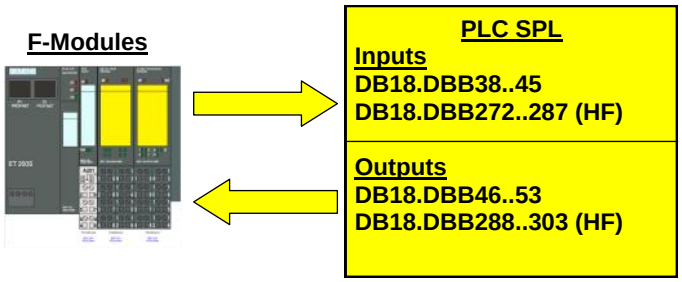
n=Number of input or output. Range is 1 to 192.

Once assigned, the 'NCK Safe Programmable Logic' (SPL) can then read from and write to the I/O of the F-Modules using the $A_INSE & $A_OUTSE system variables.



The PLC SPL will also require access to the inputs and outputs of the F-Modules.

When the F-Module I/O is assigned to the NCK System Variables, the PLC SPL can then read from and write to DB18 within the PLC.



The system handles the transfer of signals between the F-Modules and DB18 automatically.

For every 'Safety Related NCK System Variable', there is a corresponding bit within DB18. This allows for 'Redundancy' in the monitoring of the 'Safety Related System Variables'. Both the NCK Safety Related System Variable (e.g. $A_OUTSE[n]) and the corresponding bit in DB18, must always have the same equivalence. If not, a 'Crosscheck' error will be displayed.
E.g. **27090 Error in Crosswise Data Comparision NCK-PLC'.**

Notes :

In order to assign the F-Module Input and Output channels to the system variables $A_INSE & $A_OUTSE, we must set the following General Machine Data's:

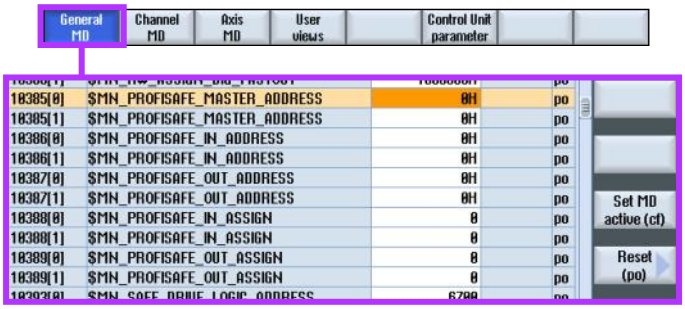
10385[0] $MN_PROFISAFE_MASTER_ADDRESS

10386[0] $MN_PROFISAFE_IN_ADDRESS
10386[1] $MN_PROFISAFE_IN_ADDRESS

10387[0] $MN_PROFISAFE_OUT_ADDRESS

10388[0] $MN_PROFISAFE_IN_ASSIGN
10388[1] $MN_PROFISAFE_IN_ASSIGN

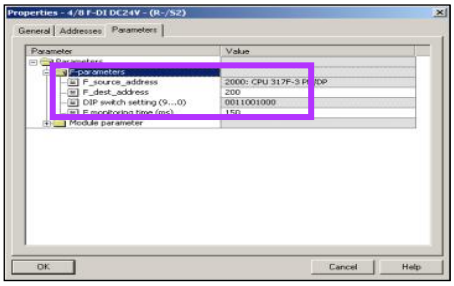
10389[0] $MN_PROFISAFE_OUT_ASSIGN
10389[1] $MN_PROFISAFE_OUT_ASSIGN



10385[0] $MN_PROFISAFE_MASTER_ADDRESS

Defines the PROFIsafe address for the F master NCK/PLC. This is used to uniquely assign an F master to an F slave. This parameter must be entered in accordance with the "F_source_address" parameter set in S7--ES for the F slaves. An attempt to establish communications is only made for F slaves that have entered this address.

Format: 0**s** 00 **aaaa**
s: Bus segment (5 = I/O connection on the PLC side)
aaaa: **Hexadecimal** PROFIsafe address of the F master.



Earlier, we noted the '**F_source_address**' from the Step7 hardware configuration. We can see in this case it is set to '$\mathbf{2000_{10}}$'. **F_source_address = 2000_{10} = $7D0_{16}$**

Therefore, we must set **10385[0]** = 0**5**00**07D0**

10386[0..1] $MN_PROFISAFE_IN_ADDRESS

Definition of the PROFIsafe target address of a PROFIsafe input module. As specified in the 'F_dest_address'.
Format: 0**s** 00 **aaaa**
s: Bus segment (5 = I/O connection on the PLC side)
aaaa: **Hexadecimal** PROFIsafe address of the F module.

Again, we noted that the '**F_dest_address**' for the first F-Module (4/8 FDI) is **200_{10}** = **$C8_{16}$** and **199_{10}** = **$C7_{16}$** for the second F-Module (4 F-DI/3 F-DO). This is also the DIP switch setting value of each F-Module.

Therefore, we must set;
10386[0] = 0**5**00**00C8** for the '4/8 F-DI' Module
10386[1] = 0**5**00**00C7** for the '4 F-DI/3 F-DO' Module

10387[0] $MN_PROFISAFE_OUT_ADDRESS

Definition of the PROFIsafe target address of a PROFIsafe output module. As specified in the 'F_dest_address'.
Format: 0**s** 00 **aaaa**
s: Bus segment (5 = I/O connection on the PLC side)
aaaa: **Hexadecimal** PROFIsafe address of the F-module.

The '**F_dest_address**' for the first output F-Module (second Module in the rack - 4 F-DI/3 F-DO) is **199_{10}** = **$C7_{16}$**

Therefore, we must set;
10387[0] = 0**5**00**00C7** for the '4 F-DI/3 F-DO' Module

Note: As the second F-Module is both an Input & Output device, the 'F_dest_address' is the same for….

10386[1] $MN_PROFISAFE_IN_ADDRESS

….and

10387[0] $MN_PROFISAFE_OUT_ADDRESS

Notes :

10388[0..1] $MN_PROFISAFE_IN_ASSIGN

The SPL area data is specified in the decimal notation in the following format: **aaa bbb**
aaa = area limit 1, SPL signal $A_INSE[**aaa**]
bbb = area limit 2, SPL signal $A_INSE[**bbb**]

Example:
PROFISAFE_IN_ASSIGN[0] = **4001** or alternatively **1004**:
The system variable area $A_INSE[**1**] to $A_INSE[**4**] is supplied with the state of the input terminals of the PROFIsafe module, which were parameterized using MD PROFISAFE_IN_ADDRESS[0] and were selected using MD PROFISAFE_IN_FILTER[0].

For this class, we will assign the following Safety Related External Inputs $A_INSE[1..4] to the first F-Module (4/8 F-DI) and $A_INSE[9..12] to the second F-Module (4 F-DI/3 F-DO).

10388[0] $MN_PROFISAFE_IN_ASSIGN = 001004
10388[1] $MN_PROFISAFE_IN_ASSIGN = 009012

Note: It is not necessary to leave a gap in the assignment of the $A_INSE/$A_OUTSE variables.

10389[0] $MN_PROFISAFE_OUT_ASSIGN

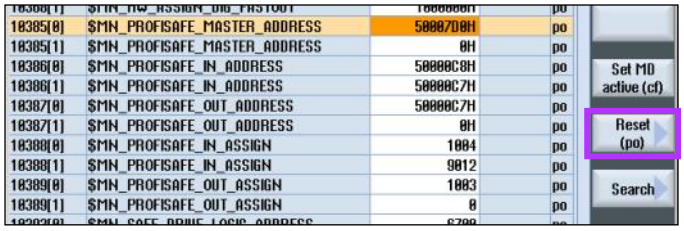
The SPL area data is specified in the decimal notation in the following format: **aaa bbb**
aaa = area limit 1, SPL signal $A_OUTSE[**aaa**]
bbb = area limit 2, SPL signal $A_OUTSE[**bbb**]

Example:
PROFISAFE_OUT_ASSIGN[0] = **4001** or alternatively **1004**:
The system variable area $A_OUTSE[**1**] to $A_OUTSE[**4**] is supplied with the state of the input terminals of the PROFIsafe module, which were parameterized using MD PROFISAFE_OUT_ADDRESS[0] and were selected using MD PROFISAFE_OUT_FILTER[0].

For this class, we will assign the following Safety Related External Outputs $A_OUTSE[1..3] to the second F-Module (4 F-DI/3 F-DO).

10389[0] $MN_PROFISAFE_IN_ASSIGN = 001003

Once the machine data's have been entered, execute a 'Reset (PO)'.

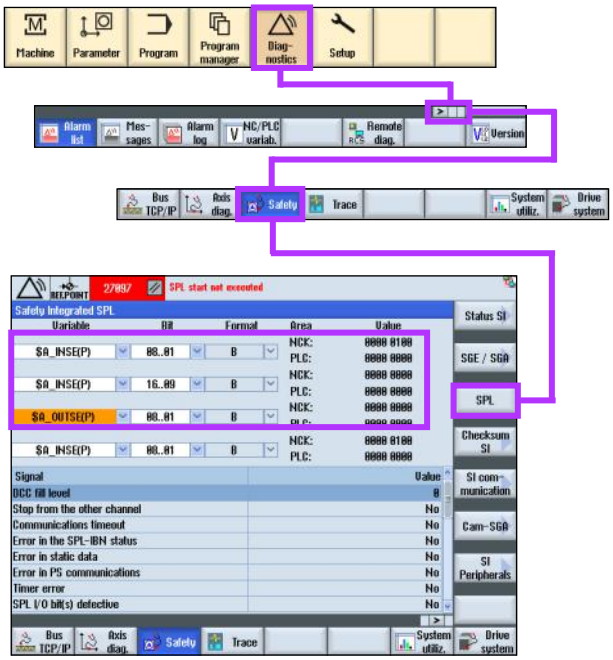


Following the NCK reset….

….'27097 SPL start not executed' will appear.

We can ignore this error for now.

It is now possible to see the status of the External Safety Inputs and Outputs $A_INSE & $A_OUTSE.



Select the variables as shown and operate the Safety devices (Keyswitches, Operator Door...etc) and notice that the status bits change in the NCK only, and not the in PLC.

Notes :

The Safety Related Devices of the training rig are now assigned as follows;

4 F-DI Module

$A_INSE[1] Emergency Stop from HT2
$A_INSE[2] Emergency Stop Reset Push Button
$A_INSE[3] Operator Door Guard Switch
$A_INSE[4] HT2 Enable Button

4 F-DI/3 F-DO Module
Inputs

$A_INSE[9] End Limit Override Keyswitch
$A_INSE[10] Setup Mode Keyswitch
$A_INSE[11] Mode 3 Keyswitch
$A_INSE[12] Spare

Outputs

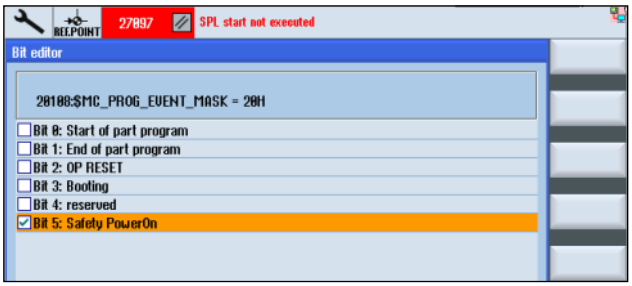
$A_OUTSE[1] Output for KA1 & KA2 Contactors
$A_OUTSE[2] Spare
$A_OUTSE[3] Spare

We can now address the NC Error '27097 SPL start not executed'.

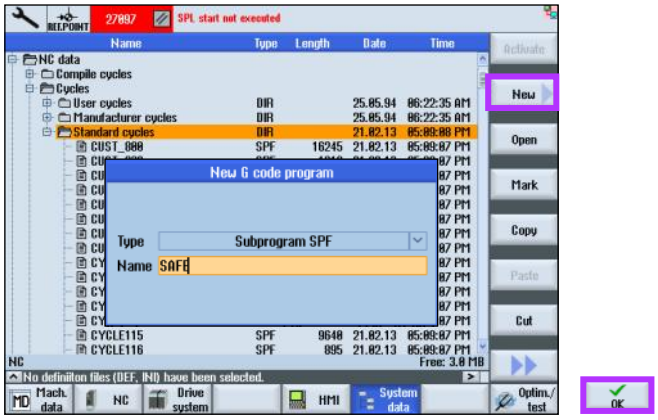
'Sinumerik System Based Safety Integrated' requires 'Safe Programmable Logic' (SPL) to run in both the NC and PLC. The above error informs us that the system has not executed the file '/CST/SAFE.SPF'. By default, the system expects to see the SAFE.SPF file run within 20s of power on (MD13310 $MN_SAFE_SPL_START_TIMEOUT), if not, then the error 27097 is displayed.

SAFE.SPF is the NCK SPL, it will eventually contain the user program that will define how the various safety functions will be selected. The logic within the NCK SPL will be created using Static Synchronous Actions (IDS=…).

Executing the SAFE.SPF at system power on is enabled by MD20108 $MC_PROG_EVENT_MASK bit 5.

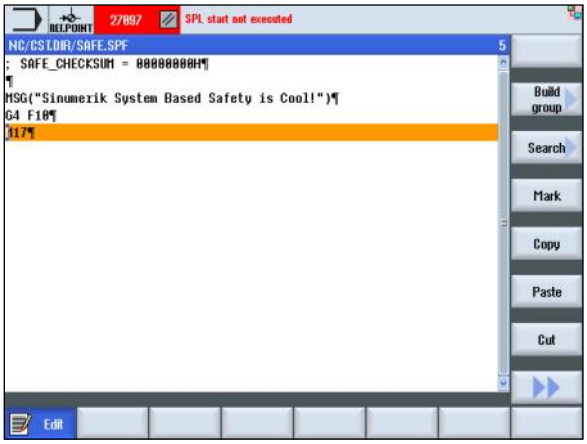


The file '/CST/SAFE.SPF' must be created using the 'System Data' screen within the 'Setup' Area.



Note: The SAFE.SPF file MUST exist within the 'Standard Cycles'(CST.DIR) folder in the NCK.

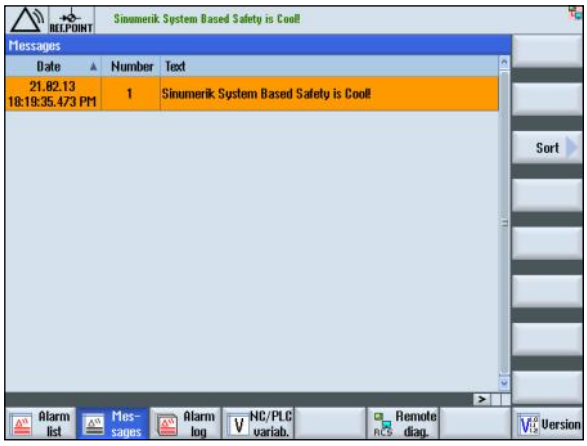
The editor will open and the system will automatically insert the first for the SAFE_CHECKSUM value.



Enter some test code as shown and include the M17 to indicate the end of the file. Close the editor and execute an NCK reset.

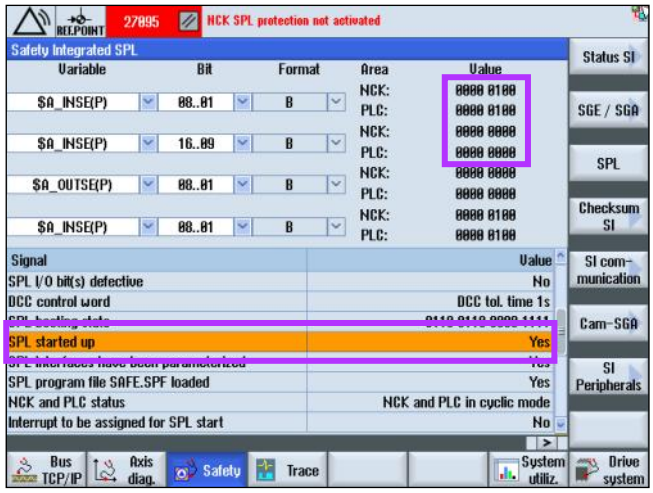
Notes :

Immediately after power on, the test message will appear…



….and after the SAFE.SPF has been executed, the alarm '**28095 NCK SPL protection not activated**' will be displayed. This alarm is normally displayed throughout the commissioning of System Based Safety. It can be cleared with reset and will re-appear periodically and after power on until the commissioning is complete.

The 'Safety' screen within the 'Diagnostics' area will now show that the SPL has started….



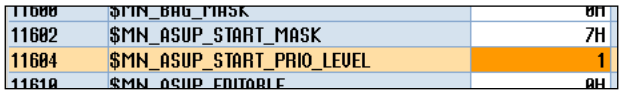
….and you will also now see that **both** NCK & PLC status bits are displayed when the safety devices are operated.

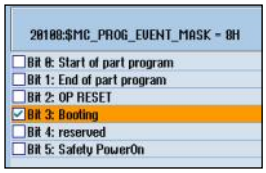
The NCK SPL (CST.DRI/SAFE.SPF) can also be started using the PROG_EVENT mechanism.

Using the PROG_EVENT mechanism will also require the option for 'Multiple Mode Actions' 6FC5800-0AM43-0YB0.

Prior to S/W 2.6 SP1, the M43 option was not required. The ability to execute 'CST.DRI/PROG_EVENT.SPF' at power-on. It was included within the Safety Integrated license M63 / M64.

Additional Md's to be set for the PROG_EVENT mechanism;
MD11602 = 7
MD11604 = 1





Setting bit 3 of MD 20108 $MC_PROG_EVENT_MASK will cause 'CST.DIR/PROG_EVENT.SPF' to be executed at power on.

The standard PROG_EVENT.SPF routine will call 'CMA.DIR/CYCPE_MA.SPF' if it exists.

The system variable $P_PROG_EVENT can be interrogated to determine the event.

Example of 'CMA.DIR/CYCPE_MA.SPF';

```
;=========================================
;$P_PROG_EVENT == 0 : Implicit Call of PROG_EVENT.SPF
;$P_PROG_EVENT == 1 : Start of Program
;$P_PROG_EVENT == 2 : End of Program
;$P_PROG_EVENT == 3 : Operator Panel Reset
;$P_PROG_EVENT == 4 : System Power-on
;$P_PROG_EVENT == 5 : After Block Search
;=========================================
IF ($P_PROG_EVENT == 4)
  CALL "/_N_CST_DIR/_N_SAFE_SPF"
ENDIF
M17
```

Another method for executing the NCK-SPL ('CST.DIR/SAFE.SPF') is to use the PI-Services block within the PLC FB4 "PI_SERV" in conjunction with FC9 "ASUP". The M43 Option may also be required for this method.

Notes :

Regardless of how the NCK-SPL ('CST.DIR/SAFE.SPF') is executed at start-up, useful status information can be interrogated within the PLC using DB18 "SPL".

DB18.DBX136.0	SPL_STATUS[1]	NCK-SPL Interfaces parameterized
DB18.DBX136.1	SPL_STATUS[2]	NCK-SPL Program file exists
DB18.DBX136.2	SPL_STATUS[3]	NCK Waits for the PLC to boot
DB18.DBX136.3	SPL_STATUS[4]	NCK and PLC in cyclic operation
DB18.DBX136.4	SPL_STATUS[5]	Call FB4 processing for SPL
DB18.DBX136.5	SPL_STATUS[6]	Exit FB4 processing for SPL
DB18.DBX136.6	SPL_STATUS[7]	Call FC9 processing for SPL
DB18.DBX136.7	SPL_STATUS[8]	Exit FC9 processing for SPL
DB18.DBX137.0	SPL_STATUS[9]	SPL start implemented using PROG_EVENT mechanism
DB18.DBX137.1	SPL_STATUS[10]	Crosswise data comparison started, NCK
DB18.DBX137.2	SPL_STATUS[11]	Crosswise data comparison started, PLC
DB18.DBX137.3	SPL_STATUS[12]	NCK-SPL checksum checking active
DB18.DBX137.4	SPL_STATUS[13]	All SPL protective mechanisms active
DB18.DBX137.5	SPL_STATUS[14]	End of SPL program reached
DB18.DBX137.6	SPL_STATUS[15]	SPL start via Safety-PowerOn
DB18.DBX137.7	SPL_STATUS[16]	Not Assigned

Now that the NCK-SPL is being executed at Power-on as an ASUP using the 'Safety PowerOn', 'PROG_EVENT' or PLC controlled PI service (FB4/FC9) method. We also require a PLC -SPL to be called in the PLC user program once the NCK-SPL has completed.

The PLC-SPL block type (FB, FC...etc) and number are freely assignable.

For the purpose of this class, FC91 "fcSI:PLC-SPL" will be used for the actual SPL.

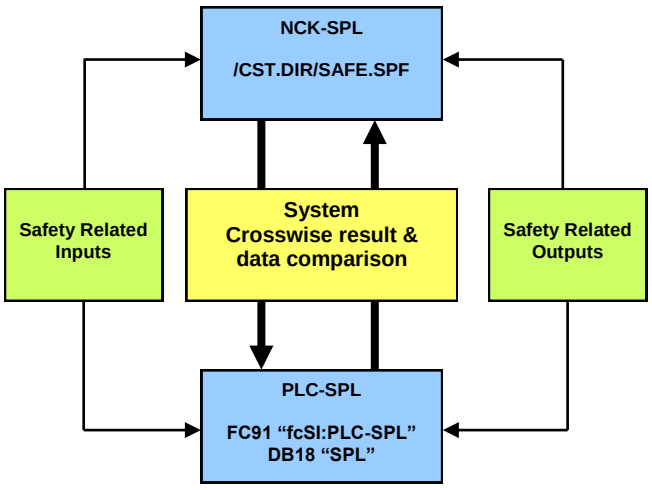
FC91 will be called from FC90 "fcSI" (FC90 is called cyclically from OB1).

FC91 "fcSI:PLC-SPL" will be called conditionally and only when the NCK-SPL has been executed. DB18.DBX137.5 can be used to detect the status of the NCK-SPL.

```
Network 4 : Call PLC-SPL When NCK-SPL has been executed
Comment:

      A     "SPL".SPL_STATUS[14]        DB18.DBX137.5
      CC    "fcSI:PLC-SPL"              FC91              -- Safety Integrated - PLC SPL

Network 5 : Title:
```

The programming language (LAD, STL, FBD...etc) for the PLC-SPL is open and remains the preference of the OEM.

The SPL will contain the Safety Logic for the machine. The logic will manipulate Safety Related System variables and the status of these variables will be critically monitored and evaluated for discrepancy between the NCK-SPL and PLC-SPL.

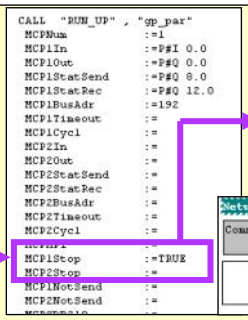


The functionality of the logic in both the NCK-SPL and PLC-SPL must be identical. Only a short time (1s) is allowed for any difference in the state of a Safety Related System Variable.

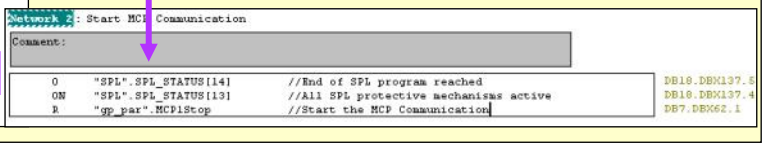
Notes :

Tip: In order to ensure that the NCK-SPL (CST.DIR/SAFE.SPF) is executed without restriction during startup of the control, any or all of the following steps could be implemented:

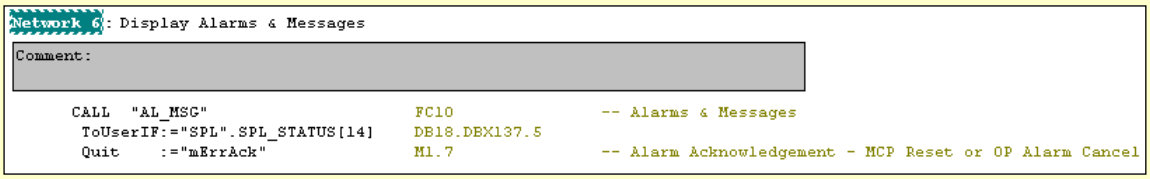
1) Within OB100 set the FB1 “RUN_UP” parameter ‘MCP1Stop’ to TRUE to Stop the MCP communication…..



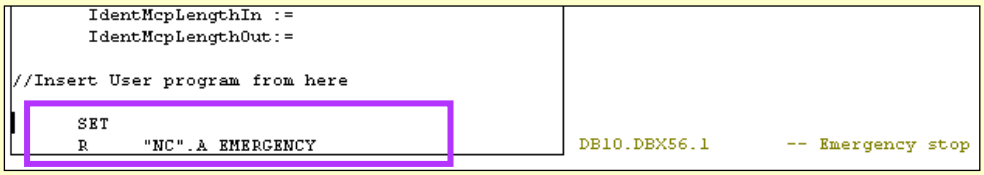
….then start the MCP communication from a cyclically called block (e.g. FC90 called from OB1) once the NCK-SPL has been executed or the NCK-SPL protection is not activated (MD11500[0..1] & DB18.DBX36.0).



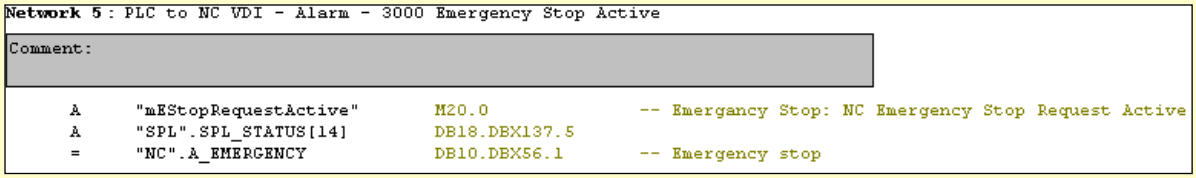
2) If you’re alarm concept uses the FC10 “AL_MSG” parameter ‘ToUserIF := TRUE’, then wait until the NCK-SPL has been executed before setting the parameter TRUE. This will allow the NCK-SPL to execute even if alarms and messages that generate ‘NC-Start-Disable’ or ‘Read-in-Disable’ are active.



3) Within OB100, reset the PLC to NC VDI Signal for ‘Emergency Stop’ so that NC Alarm 3000 does not appear….



….until there is an active request for emergency stop from the user program and the NCK-SPL has completed.



This completes the section ‘Enabling Safety in the NC’.

The Safety I/O is now assigned to ‘Safety Related System Variables’ $A_INSE and $A_OUTSE and the SAFE.SPF file is now started automatically at power on. The PLC-SPL (FC91) is also called once the NCK-SPL has completed.

Notes :

F-Controller Logic: Writing the SPL for the Emergency Stop (SIRELAY)

This section describes the following;

- Assigning friendly names to the safety related system variables in NCK-SPL(DEFINE) and PLC-SPL(UDT 18)
- Use of the Safe Software Relay SIRELAY
- Description of the safety related system variables $A_MARKERSI, $A_PLCSIIN, $A_PLCSIOUT

NOTE: Any NC/PLC code shown in this document is to be treated as an example and for guidance purposes only!

Notes :

In order to help with the structure and to make the SPL easier to view, it is advisable to assign symbolic names to the safety related system variables in both the NCK-SPL and the PLC-SPL.

To assign Symbolic Names to the NCK-SPL, the 'DEFINE' instruction is used within the SAFE.SPF file.

```
; SAFE_CHECKSUM = 00010264H
;+--------------------------------------+
;|  Safe Programmable Logic (NCK-SPL)   |
;+--------------------------------------+
; File:              SAFE.SPF
; Company:           SIEMENS IA & DT MC MT
; Project:           A102
; Author:
; Creation Date:
;--------------------------------------
;
;---------- Cycle Definition -----------
;Suppress Single Block, Display
PROC SAFE SBLOF DISPLOF

;---------- Declaration ----------------
;Define Symbolic Names for SPL Variables
;======================================
;External Inputs - $A_INSE
DEFINE IE1_EStop         AS $A_INSE[1]
DEFINE IE2_EStop_Reset   AS $A_INSE[2]
DEFINE IE3_OP_Door       AS $A_INSE[3]
DEFINE IE4_HT2_Enable    AS $A_INSE[4]

DEFINE IE9_OT_Override   AS $A_INSE[9]
DEFINE IE10_Setup        AS $A_INSE[10]
DEFINE IE11_Mode3        AS $A_INSE[11]
DEFINE IE12_Spare        AS $A_INSE[12]

;External Outputs - $A_OUTSE
DEFINE OE1_KA1_KA2       AS $A_OUTSE[1]
DEFINE OE2_Spare         AS $A_OUTSE[2]
DEFINE OE3_Spare         AS $A_OUTSE[3]

M17
```

Within the 'Archives/ Manufacturer/A102' folder, the archive '**A102_SAFE_001.ARC**' can be loaded to save time.

To assign symbolic names for to the PLC-SPL, compile '**A102_UDT18_001**' within the source folder of the Step7 project.

```
STOP_MODE : BOOL ;
SPL_DATA : STRUCT
 IE1_EStop           : BOOL ;    //INSEP1
 IE2_EStop_Reset     : BOOL ;    //INSEP2
 IE3_OP_Door         : BOOL ;    //INSEP3
 IE4_HT2_Enable      : BOOL ;    //INSEP4
 A_INSE5 : BOOL ;    //INSEP5
 A_INSE6 : BOOL ;    //INSEP6
 A_INSE7 : BOOL ;    //INSEP7
 A_INSE8 : BOOL ;    //INSEP8
 IE9_OT_Override     : BOOL ;    //INSEP9
 IE10_Setup          : BOOL ;    //INSEP10
 IE11_Mode3          : BOOL ;    //INSEP11
 IE12_Spare          : BOOL ;    //INSEP12
 A_INSE13 : BOOL ;   //INSEP13
 A_INSE14 : BOOL ;   //INSEP14
```

…..

```
 A_INSE64 : BOOL ;   //INSEP64
 OE_KA1_KA2   : BOOL ;     //OUTSEP1
 OE2_Spare    : BOOL ;     //OUTSEP2
 OE3_Spare    : BOOL ;     //OUTSEP3
 A_OUTSE4 : BOOL ;   //OUTSEP4
 A_OUTSE5 : BOOL ;   //OUTSEP5
```

Check that UDT 18 is assigned to DB18 within the Symbols Table.

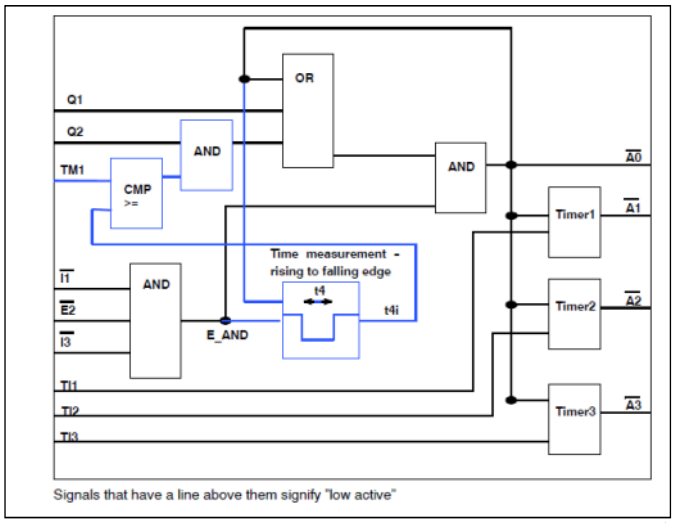
PI	DB	16	DB	16	Datainterface PI-Services
SPL	DB	18	UDT	18	SPL dataarea
MMC	DB	19	UDT	19	Interface MMC-Signals

Symbolic names for the Safety Related System Variables should be added to the NCK & PLC SPL as new variables are used.

Safe software relay

Before we write the SPL code for the emergency stop circuit, we must understand the use of the 'Safety Software Relay" **SIRELAY**. Up to four SIRELAY's can be configured.

The standard SPL block "safety software relay" is designed to meet the requirements of an Emergency Stop function with safe programmable logic. However, it can also be used to implement other similar safety functions, e.g. to control a protective door.



Three shutdown inputs E1, E2 & E3

If one of these inputs is set to 0, direct output A0 is set to 0. Outputs A1, A2 & A3 switch off with the delay of TI1, TI2 & TI3. If one of these inputs is not used, then it is internally set to "1" as static signal. If required, one of these inputs must also be used to initiate test operation of the safety relay (forced checking procedure).

Q1 & Q2 Acknowledge Inputs

Q1 must be supplied with the signal from the real acknowledge button.

Q2 is only used for the forced checking procedure. The software relay itself does not have to be subject to a forced checking procedure. However, if the Emergency Stop function is executed and if external actuators have to be subject to a forced checking procedure, if the relay drops--out during the Emergency Stop test, then it can be acknowledged using Q2 (in a defined time window, refer to TM1). This input must also be connected with a safety system variable (even if the signal is not used) preferably with a $A_MARKERSI in order to detect that this acknowledge signal is available as steady state signal in the crosswise data comparison with the PLC. The associated comparison data in the PLC must have a steady state 0 signal level.

Notes :

Three timer initialization values TI1, TI2 & TI3
The times after which outputs A1, A2 & A3 are switched to 0 at a negative edge in output signal A0 are defined here.

One timer limit value TM1
Defines the maximum time that the shutdown inputs E1, E2 & E3 may have been at a 0 signal level so that they can still be acknowledged using Q2. Q2 should only be used for the internal safety relay test. It is not permissible that Q2 is used to acknowledge a ”real” shutdown.

Four output values A0 to A3
A0 supplies the result of **AND**ing E1, E2 & E3 without any delay. Outputs A1, A2 & A3 supply the same result for positive edges of A0; for negative edges, the results are delayed by the timer initialization values TI1, TI2 & TI3(switch-off delay).
A0 to A3 do not produce a result after booting until an acknowledgment has been received via Q1.

Initialization in the SAFE.SPF program
The connections for the function block are defined when initialized. The input and output quantities of the function block are assigned to the required system variables ($A_MARKERSI, $A_INSE, $A_OUTSE,...). The following functions must be called:

(The status value contains the number of the first incorrect parameter; a value of 0 indicates that the parameter assignment is correct.)

SIRELIN:
This language command assigns the input quantities Q1, Q2, E1, E2 and E3 to the safety relay x (x = 1..4).

Syntax: **SIRELIN(x,status,”Q1”,”Q2”,”E1”,”E2”,”E3”)**

The transfer parameters Q1 to E3 are strings and must therefore be entered in quotation marks (” ”). The following system variables are permissible as input quantities:
$A_MARKERSI[ ]
$A_INSE[ ]
$A_INSI[ ]
$A_OUTSE[ ]
$A_OUTSI[ ]
E2 and E3 are optional. If these parameters are not entered, the relevant inputs are set to ”1” (static signal).

SIRELOUT:
This language command assigns the output quantities A0, A1, A2 and A3 to safety relay x (x = 1..4).

Syntax: **SIRELOUT(x,status,”A0”,”A1”,”A2”,A3”)**

The transfer parameters A0 to A3 are strings and must therefore be entered in quotation marks (” ”). The following system variables are permissible as output quantities:

$A_MARKERSI[ ]
$A_OUTSE[ ]
$A_OUTSI[ ]
$A_PLCSIOUT[ ]

A1 to A3 are optional. If these parameters are not specified, then the corresponding outputs are not supplied. However, if A1 is specified, the initialization value for timer 1 (TI1) must also be parameterized via SIRELTIME. The same applies for A2 and timer 2 (TI2) and A3 and timer 3 (TI3).

SIRELTIME:
This language command assigns the times for the timers required to safety relay x (x = 1..4). These include the timer limit value TM1 and the timer initialization values TI1, TI2 and TI3.

Syntax: **SIRELTIME(x,status,TM1,TI1,TI2,TI3)**

Transfer parameters TM1 & TI1 to TI3 are REAL numbers (timers in seconds). TI1 to TI3 are optional.
If these parameters are not specified, the corresponding outputs A1 to A3 are not supplied. However, if TI1 is specified, output A1 must also be parameterized via SIRELOUT. The same applies for TI2 and A2 and TI3 and A3.

Notes :

SIRELAY:
Cyclic Monitoring of the configured SIRELAY is required for operation. The cyclic section must be integrated in the synchronized actions of the SPL. Only the SIRELAY number (1 to 4) is passed to the SIRELAY function.

A ‘Status’ value other than 0 will indicate a fault with the parameterization of the SIRELIN, SIRELOUT or SIRELTIME function calls.

Syntax: **Status = SIRELAY(x)**

The ”Status” variable must be defined as integer to correctly map the possible return values of the function block.

The following values are possible for status:

Status Value	Significance
0	SIRELAY parameterized correctly
1	The input quantities of the safety relay are either not parameterized or incorrectly parameterized. Remedy: Call SIRELIN with the correct parameterization.
2	The output quantities of the safety relay are either not parameterized or incorrectly parameterized. Remedy: Call SIRELOUT with the correct parameterization.
3	The input and output quantities of the safety relay are either not parameterized or not correctly parameterized. Remedy: Call SIRELIN and SIRELOUT with the correct Parameterization.
4	The timers of the safety relay are either not parameterized or not correctly parameterized. Remedy: Call SIRELTIME with the correct parameterization
5	The input quantities and timers of the safety relay are either not parameterized or not correctly parameterized. Remedy: Call SIRELIN and SIRELTIME with the correct parameterization.
6	The output quantities of the safety relay are either not parameterized or not correctly parameterized. Remedy: Call SIRELOUT and SIRELTIME with the correct parameterization
7	The initialization of the safety relay was either not carried out or not correctly carried out. Remedy: Call SIRELIN, SIRELOUT and SIRELTIME with the correct parameterization

Example:
Example code for the emergency stop relay function within the NCK-SPL (SAFE.SPF) using the SIRELAY.

```
DEF INT R1_IN, R1_OUT, R1_TIME

DEFINE IE1_ESTOP            AS $A_INSE[1]
DEFINE IE2_ESTOP_RESET      AS $A_INSE[2]
DEFINE M1_QUIT2             AS $A_MARKERSI[1]
DEFINE M2_A0                AS $A_MARKERSI[2]
DEFINE M3_A1                AS $A_MARKERSI[3]
DEFINE M4_A2                AS $A_MARKERSI[4]
DEFINE M5_A3                AS $A_MARKERSI[5]
DEFINE R1_STATUS            AS $AC_MARKER[1]

SIRELIN(1, R1_IN, ”IE2_ESTOP_RESET ”, “M2_QUIT2”, “IE1_ESTOP ”)

SIRELOUT(1, R1_OUT, “M2_A0”, “M3_A1”, “M4_A2”, “M5_A3”)

SIRELTIME(1, R1_TIME, 0.4, 1.0, 2.0, 3.0)

IDS = 200 DO R1_STATUS = SIRELAY(1)

;Error Handling
IDS = 201 EVERY (R1_STATUS <> 0) DO …..
```

Example:
FB10 “SI_Relais” within the PLC can be called within the PLC-SPL to compliment the NCK-SPL SIRELAY function.

FB10 should be called once for each SIRELAY(1..4) with a separate instance data block.

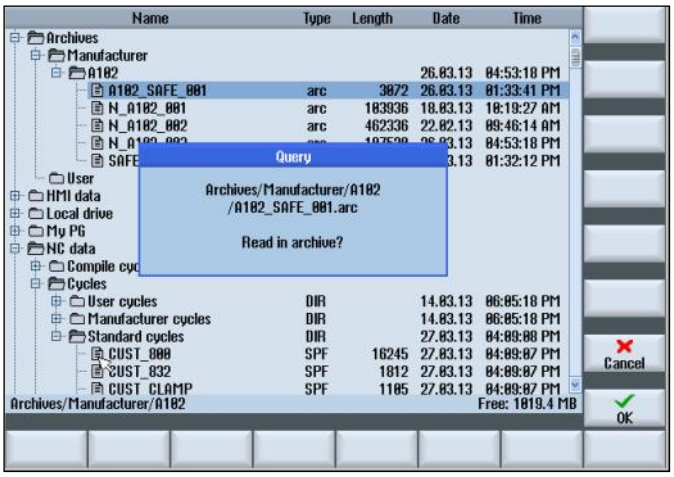
```
FUNCTION_BLOCK FB 10
```

Declaration of the function

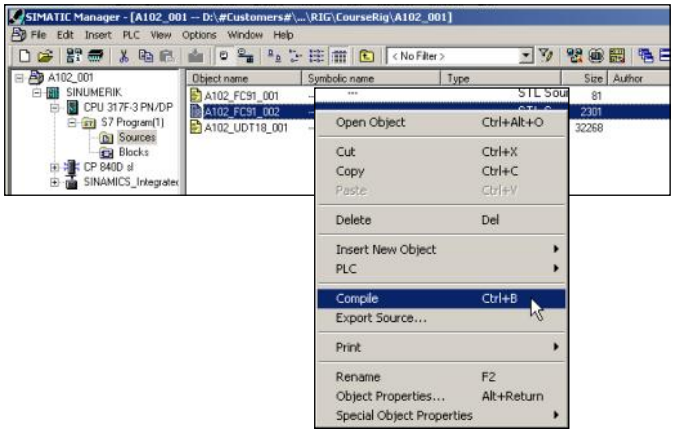
```
VAR_INPUT
        In1 : BOOL := True ; // Input 1
        In2 : BOOL := True ; // Input 2
        In3 : BOOL := True ; // Input 3
        Ackn1 : BOOL; // Ackn1 signal
        Ackn2 : BOOL; // Ackn2 signal
        TimeValue1 : TIME := T#0ms ; // TimeValue for Output 1
        TimeValue2 : TIME := T#0ms ; // TimeValue for Output 2
        TimeValue3 : TIME := T#0ms ; // TimeValue for Output 3
END_VAR
VAR_OUTPUT
        Out0 : BOOL; // Output without Delay
        Out1 : BOOL; // Delayed Output to False by Timer 1
        Out2 : BOOL; // Delayed Output to False by Timer 2
        Out3 : BOOL; // Delayed Output to False by Timer 3
END_VAR
VAR_INOUT
        FirstRun: BOOL; // True by User after 1st start of SPL
END_VAR
```

Notes :

Within the ‘Archives/Manufacturer/A102’ folder, load the archive ‘**A102_SAFE_001.ARC**’. This archive contains a SAFE.SPF with the SIRELAY configured for the simulator.



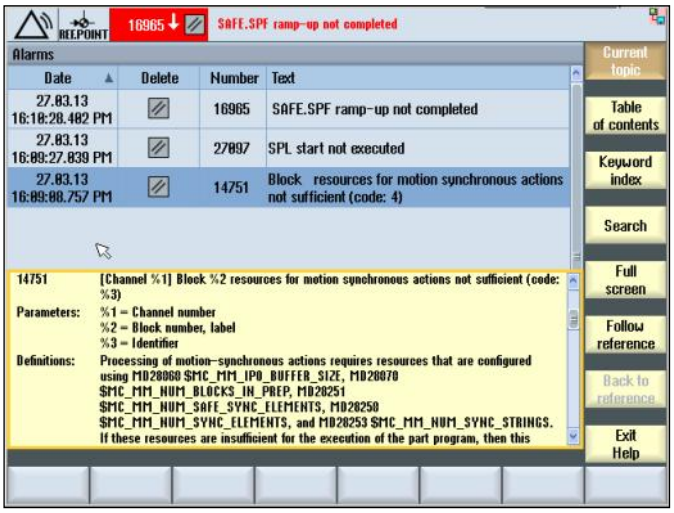
Within Step7, compile the source file ‘**A102_FC91_001**’….



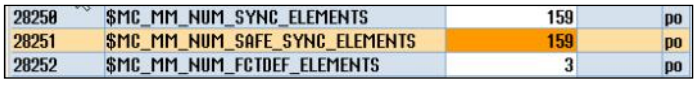
...now download the newly complied FC91 “fcSI:PLC-SPL” to the PLC from within the blocks folder.

Perform an NCK Reset to activate the new version of NCK-SPL ‘SAFE.SPF’.

Following the NCK reset, The NCK error 14751 will appear.



This error is due to the default value of MD28251 $MC_MM_NUM_SAFE_SYNC_ELEMENTS = 0 Change this value to 159.



NCK Reset to activate new value.

We can now look at the SPL code that has been loaded to the NCK(SAFE.SPF) and the PLC (FC90 “fcSI” & FC91”fcSI:PLC-SPL”).

You will see new safety variables used, namely:
$A_MARKERSI and $A_PLCSIIN

$A_MARKERSI are internal Boolean markers that the SPL can use to store intermediate states. There are 64 ($A_MARKERSI[1..64]) in total and all can be freely assigned within the SPL.

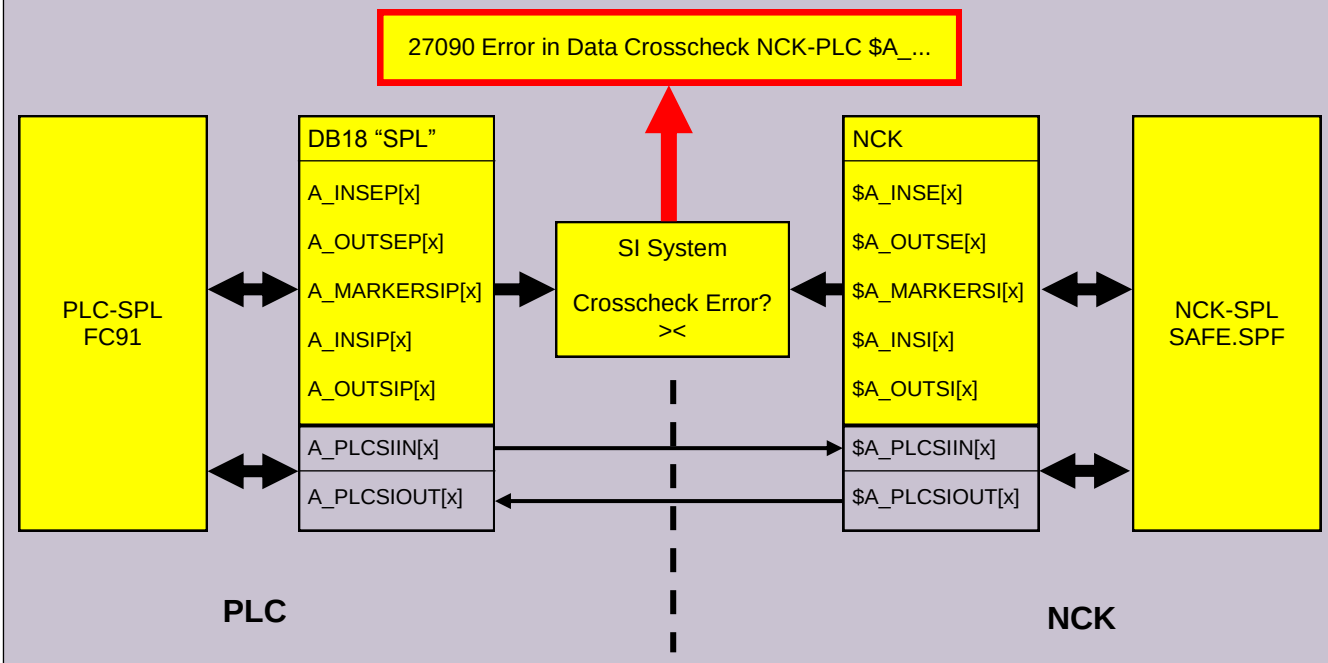
For each Safety Related System Variable (e.g. $A_INSE, $A_OUTSE, $A_MARKERSI...etc) within the NC, there is a corresponding bit within DB18 (“SPL)” of the PLC.

To avoid errors (e.g. 27090 Error in Data Crosscheck…..), the PLC-SPL must manipulate the appropriate bits within DB18 in the same way that the NCK-SPL manipulates the system variables $A_....etc.

One exception to this rule are the Boolean system variables $A_PLCSIIN[1..32 <96 for HF Option>] and $A_PLCSIOUT[1..32 <96 for HF Option>].
These variables are ‘Single Channel Variables’ and can be used to communicate none ’SAFE’ signals to and from the SPL.

Notes :

NC Fault '27090 Error in Data Crosscheck NCK-PLC $A_... NCK ?' will be posted if the SI system detects a difference of state between the PLC(DB18) and the NCK (Saftey Related System Variables).
PLCSIIN and PLCSIOUT communication variables are 'Single Channel and are not subject to 'Crosscheck' monitoring. They are passes directly to/from the NCK for use within both PLC & NCK.



FC90 "fcSI" is called cyclically from OB1. This is the 'Support' block for PLC-SPL FC91 "fcSI:PLC-SPL". FC90 prepares and reacts to the state of the safety related variables within the PLC-SPL. Throughout this course, FC90, along with FC91 will be expanded as we continue to enable and commission System Based Safety Integrated.

```
FC90 : Safety Integrated
Network 1: Start MCP Communication
      O     "SPL".SPL_STATUS14          //End of SPL program reached
      ON    "SPL".SPL_STATUS13          //All SPL protective mechanisms active
      R     "gp_par".MCP1Stop           //Start the MCP Communication

Network 2 : Initialise SIRELAY FB10 within PLC-SPL
      A     DB8.DBX   45.0              //OB1 Scan 1
      S     "mSI:SIRELAY_1_FirstRun"

Network 3 : No PLC-EStop request active to SPL
      AN    "mPLCEStopRequestActive"
      =     "SPL".IP1_PLC_READY

Network 4 : Call PLC-SPL When NCK-SPL has been executed
      A     "SPL".SPL_STATUS14          //End of SPL program reached
      CC    "fcSI:PLC-SPL"

Network 5 : Generate Emergency Stop request when SIRELAY is off.
      AN    "SPL".SPL_DATA.M6_SIREL1_OUT0
      AN    "NC".E_EMSTOP
      =     "idbEStop".ReqEStop[2]

Network 6 : Emergency Stop Reset PB LED
      A     "SPL".SPL_STATUS14          //End of SPL program reached
      AN    "NC".A_EMERGENCY
      A     "SPL".SPL_DATA.IE1_EStop
      A     "m0.5Hz"
      O     "SPL".SPL_DATA.M6_SIREL1_OUT0
      =     "qPP72x48:X222.39:35.0"
```

Here, we pass the status of any active Emergency Stop requests from the PLC (FB100 "fbEStop") to the SPL using the single channel variable $A_PLCSIIN[1]. (FB100 "fbEStop" checks for emergency stop requests from within the PLC i.e. Circuit Breaker Tripped, No Hydraulic Pressure...etc. It also controls the VDI signals for NCK Alarm '3000 Emergency Stop Active'.

Call FC91 "fcSI:PLC-SPL" once the NCK-SPL has completed.

Create a PLC emergency stop request for FB100 "fbEStop" if the SPL SI Relay is off. $A_MARKERSI[6] == FALSE.

Notes :

FC91 “fcSI:PLC-SPL” is the PLC equivalent of the NCK-SPL “SAFE.SPF”. The functionality of both NCK-SPL& PLC-SPL must be identical, otherwise error “27090 Error in Data Crosscheck ...” will occur.

PLC-SPL FC91 “fcSI:PLC-SPL”

```
FC91 : Safety Integrated: PLC-SPL
Network 1: IDS=200: Assign EStop Input A_INSEP[1] to A_MARKERSIP[1]
      A     "SPL".SPL_DATA.IE1_EStop
      =     "SPL".SPL_DATA.M1_SIREL1_IN1

Network 2 : IDS=201: Assign PLC Estop Input A_PLCSIIN[1] to A_MARKERSIP[2]
      A     "SPL".IP1_PLC_READY
      =     "SPL".SPL_DATA.M2_SIREL1_IN2

Network 3 : IDS=202: Assign A_MARKERSIP[3] Statically TRUE
      A     "mTRUE"
      =     "SPL".SPL_DATA.M3_SIREL1_IN3

Network 4 : IDS=203: Assign Estop Reset Input A_INSEP[2] to A_MARKERSIP[4]
      A     "SPL".SPL_DATA.IE2_EStop_Reset
      =     "SPL".SPL_DATA.M4_SIREL1_QUIT1

Network 5 : IDS=203: Assign A_MARKERSIP[5] Statically FALSE
      A     "mFALSE"
      =     "SPL".SPL_DATA.M5_SIREL1_QUIT2

Network 6 : Call SIRELAY(1) for EStop
      A     "SPL".SPL_DATA.M1_SIREL1_IN1
      =     L       0.0
      BLD   103
      A     "SPL".SPL_DATA.M2_SIREL1_IN2
      =     L       0.1
      BLD   103
      A     "SPL".SPL_DATA.M3_SIREL1_IN3
      =     L       0.2
      BLD   103
      A     "SPL".SPL_DATA.M4_SIREL1_QUIT1
      =     L       0.3
      BLD   103
      A     "SPL".SPL_DATA.M5_SIREL1_QUIT2
      =     L       0.4
      BLD   103
      CALL  "SI_Relais" , "idbSI:SIRELAY_1:EStop"
       In1        :=L0.0
       In2        :=L0.1
       In3        :=L0.2
       Quit1      :=L0.3
       Quit2      :=L0.4
       TimeValue1:=T#1S
       TimeValue2:=T#2S
       TimeValue3:=T#3S
       Out0       :="SPL".SPL_DATA.M6_SIREL1_OUT0
       Out1       :="SPL".SPL_DATA.M7_SIREL1_OUT1
       Out2       :="SPL".SPL_DATA.M8_SIREL1_OUT2
       Out3       :="SPL".SPL_DATA.M9_SIREL1_OUT3
       FirstRun   :="mSI:SIRELAY_1_FirstRun"
      NOP   0
```

NCK-SPL “/CST.DIR/SAFE.SPF”

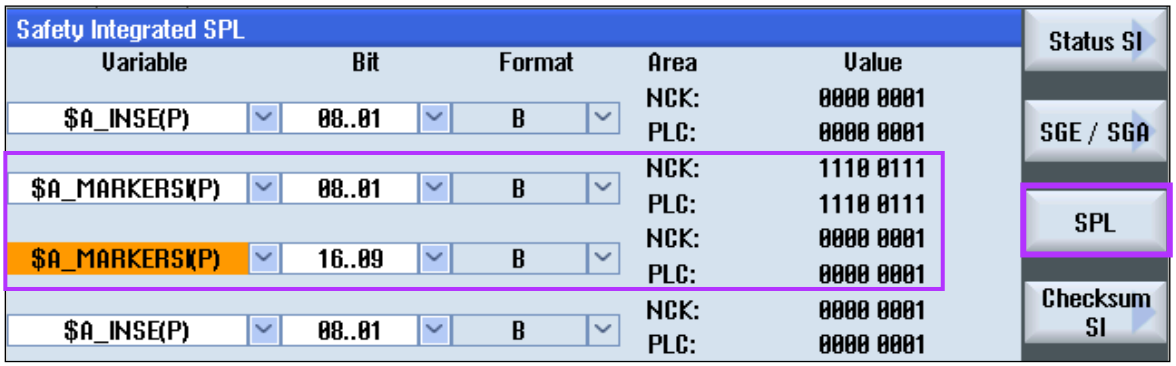
```
;Net 1:Assign EStop Input $A_INSE[1] to $A_MARKERSI[1]
N10120 IDS=200 DO MI1_SIREL1_IN1 = IE1_ESTOP

;Net 2:Assign PLC Estop Input $A_PLCSIIN[1] to $A_MARKERSI[2]
N10125 IDS=201 DO MI2_SIREL1_IN2 = IP1_PLC_READY

;Net 3:Assign $A_MARKERSI[3] Statically TRUE
N10130 IDS=202 DO MI3_SIREL1_IN3 = TRUE

;Net 4:Assign EStop Reset Input $A_INSE[2] to $A_MARKERSI[4]
N10135 IDS=203 DO MI4_SIREL1_QUIT1 = IE2_EStop_Reset

;Net 5:Assign $A_MARKERSI[5] Statically FALSE
N10140 IDS=204 DO MI5_SIREL1_QUIT2 = FALSE

;Net 6:Call SIRELAY(1) for EStop
N10145 SIRELIN(1, SIREL1_IN, "MI4_SIREL1_QUIT1", "MI5_SIREL1_QUIT2",
"MI1_SIREL1_IN1", "MI2_SIREL1_IN2", "MI3_SIREL1_IN3")
N10150 SIRELOUT(1, SIREL1_OUT, "MI6_SIREL1_OUT0", "MI7_SIREL1_OUT1",
"MI8_SIREL1_OUT2", "MI9_SIREL1_OUT3")
N10155 SIRELTIME(1,SIREL1_TIME, 0.4, 1.0, 2.0, 3.0)
N10160 IDS=205 DO M_STATUS_1 = SIRELAY(1)

N10165 M17
```

Apart from some parameters and syntax for the SIRELAY, Notice how the functionality of the code in both the NCK-SPL and the PLC-SPL is the same.

Notice how the network and IDS numbers are used in the comments of each SPL area. This helps to quickly associate the equivalent code in each area.

Using HMI Operate, Go to the ‘Diagnostics’ >> ‘Safety’ >> ‘SPL’ screen, enable the drives and press the ’Emergency Stop Reset’ button.

Observe the safety related system variables ($A_MARKERSI[1..9]) that we have used and assigned to the SIRELAY(1) within the SPL. Notice $A_MARKERSI[6..9] changing state when the Emergency Stop button on the HT2 is pressed.



Notes :

Useful Addresses for DB18

External Inputs:		External Outputs:		Internal Inputs:	
$A_INSE[1..8]	DB18.DBX38.0 - 7	$A_OUTSE[1..8]	DB18.DBX46.0 - 7	$A_INSI[1..8]	DB18.DBX54.0 - 7
$A_INSE[9..16]	DB18.DBX39.0 - 7	$A_OUTSE[9..16]	DB18.DBX47.0 - 7	$A_INSI[9..16]	DB18.DBX55.0 - 7
$A_INSE[17..24]	DB18.DBX40.0 - 7	$A_OUTSE[17..24]	DB18.DBX48.0 - 7	$A_INSI[17..24]	DB18.DBX56.0 - 7
$A_INSE[25..32]	DB18.DBX41.0 - 7	$A_OUTSE[25..32]	DB18.DBX49.0 - 7	$A_INSI[25..32]	DB18.DBX57.0 - 7
$A_INSE[33..40]	DB18.DBX42.0 - 7	$A_OUTSE[33..40]	DB18.DBX50.0 - 7	$A_INSI[33..40]	DB18.DBX58.0 - 7
$A_INSE[41..48]	DB18.DBX43.0 - 7	$A_OUTSE[41..48]	DB18.DBX51.0 - 7	$A_INSI[41..48]	DB18.DBX59.0 - 7
$A_INSE[49..56]	DB18.DBX44.0 - 7	$A_OUTSE[49..56]	DB18.DBX52.0 - 7	$A_INSI[49..56]	DB18.DBX60.0 - 7
$A_INSE[57..64]	DB18.DBX45.0 - 7	$A_OUTSE[57..64]	DB18.DBX53.0 - 7	$A_INSI[57..64]	DB18.DBX61.0 - 7
(High Feature Option)		(High Feature Option)		(High Feature Option)	
$A_INSE[65..192]	DB18.DBB272-287	$A_OUTSE[65..192]	DB18.DBB288-303	$A_INSI[65..192]	DB18.DBB304-319

Internal Outputs:		Markers:	
$A_OUTSI[1..8]	DB18.DBX62.0 - 7	$A_MARKERSI[1..8]	DB18.DBX70.0 - 7
$A_OUTSI[9..16]	DB18.DBX63.0 - 7	$A_MARKERSI[9..16]	DB18.DBX71.0 - 7
$A_OUTSI[17..24]	DB18.DBX64.0 - 7	$A_MARKERSI[17..24]	DB18.DBX72.0 - 7
$A_OUTSI[25..32]	DB18.DBX65.0 - 7	$A_MARKERSI[25..32]	DB18.DBX73.0 - 7
$A_OUTSI[33..40]	DB18.DBX66.0 - 7	$A_MARKERSI[33..40]	DB18.DBX74.0 - 7
$A_OUTSI[41..48]	DB18.DBX67.0 - 7	$A_MARKERSI[41..48]	DB18.DBX75.0 - 7
$A_OUTSI[49..56]	DB18.DBX68.0 - 7	$A_MARKERSI[49..56]	DB18.DBX76.0 - 7
$A_OUTSI[57..64]	DB18.DBX69.0 - 7	$A_MARKERSI[57..64]	DB18.DBX77.0 - 7
(High Feature Option)		(High Feature Option)	
$A_OUTSI[65..192]	DB18.DBB320-335	$A_MARKERSI[65..192]	DB18.DBB336-351

We can now modify the SPL to control Safety Related Output 1 ($A_OUTSE[1]). $A_OUTSE[1] is for the 2 relays within the simulator = KA1 & KA2, in this case, $A_OUTSE[1] will be assigned to $A_MARKERSI[6].
$A_MARKERSI[6] is connected to the OUT0 channel of the SIRELAY(1). OUT0 is the instant (non-delayed) channel.

Add a new network (7) to FC91 “fcSI:PLC-SPL” with the following code……
Note: The absolute addresses for the variables are;

$A_MARKERSI[6] = DB18.DBX 70.5
$A_OUTSE[1] = DB18.DBX 46.0

```
Network 7 : IDS=206: Assign MarkerSIP[6] to KA1_KA2 Contactor Output
      A     "SPL".SPL_DATA.M6_SIREL1_OUT0     DB18.DBX70.5       -- MARKERSIP6
      =     "SPL".SPL_DATA.OE1_KA1_KA2        DB18.DBX46.0       -- OUTSEP1
```

….Now modify the NCK-SPL ‘CST.DIR/SAFE.SPF’ and add the following block N10165 for IDS=206.

```
;Net 7:Assign MarkerSIP[6] to KA1_KA2 Contactor Output
IDS=206 DO OE1_KA1_KA2 = M16_SIREL1_OUT0
M17
```

Edit | Cut

Once the SAFE.SPF and FC91 have been modified

- Load FC91 to the PLC
- NCK reset to activate the modified NCK-SPL (the relays KA1 & KA2 will now be energized when the emergency stop circuit is reset.)
- Observe the status of $A_OUTSE[1] in the SPL Safety Diagnostics screen.

The relays KA1 & KA2 should now energize when the emergency stop is reset.

Notes :

F-Motion Monitoring:

Description of Safety Related Axis and Drive Machine Data

This section describes the following;

- Brief overview of ‘Safe Motion Monitoring’ machine data
- Explanation of the Safe Checksum monitoring
- Copy and Confirm of SI Data

NOTE: Any NC/PLC code shown in this document is to be treated as an example and for guidance purposes only!

Notes :

This section will describe how the various system based safety integrated functions are applied to the axes/spindle.

The OEM must decide what safe functions are to be used for each axis/spindle.

Available System Based Safety Functions;

-SH (STO) Safe Standstill [1)]

-SBH (SOS) Safe Operational Stop [1)]

-Safe Stopping Process (External Stops) [1)]

-SBR Safe Braking Ramp [1)]

-SG (SLS) Safe Reduced/Limited Speed - 4 Levels

-Safe Speed Limit/Monitor ($n<n_x$) (SSM)

-SE (SLP) Safe Software Limit Switches

-SN (SCA) Safe Software Cams

-SBC Safe Brake Control

[1)] Functions are mandatory for each axis/spindle

Axis and drive specific 'safe' machine data are used to enable and configure the safety functions.

Safety Machine data's exist in both 'Axis' and 'Drive' machine data areas.

All Safety related machine data for the axes begin with '**$MA_SAFE_.....'.**

Safe Axis machine data range:
MD 36901 $MA_SAFE_FUNCTION_ENABLE
….
….
MD 36999[0..2] $MA_SAFE_DES_CHECKSUM

Within the Drive, all Safety related machine data begins with '**SI Motion...**' from **p9500** onwards.

e.g. **p9500 SI Motion monitoring clock cycle (Control Unit)**

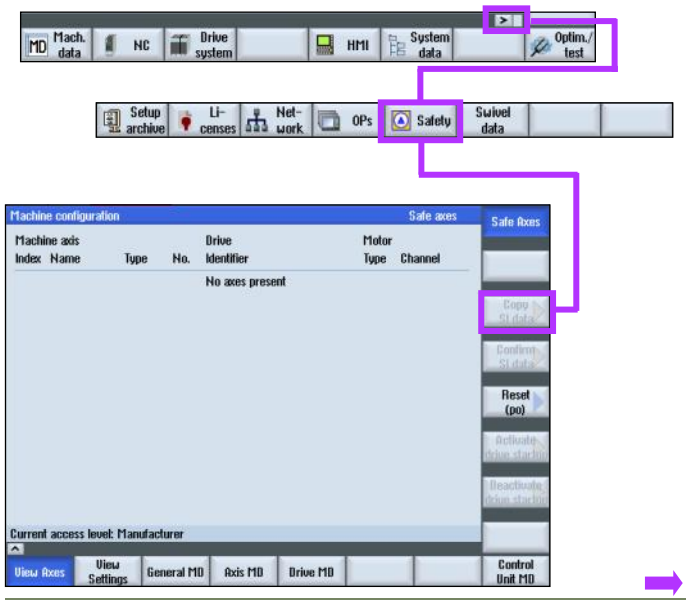
For almost every Axis specific 'Safety Related' machine data, there exists an equivalent Drive specific 'Safety Related' machine data.
e.g.
MD 36901 $MA_SAFE_FUNCTION_ENABLE

...is equivalent to...

p9501 SI Motion enable safety functions (Control Unit)

With a small number of exceptions, the value of each and every safety related machine data within the NC must match the value entered in the equivalent 'Drive' safety machine data.

However, it is not necessary to enter the values by hand in both the NC **AND** the Drive. Using the 'Safety' commissioning section of the 'Setup' area, you can copy safety related data from the 'NC' to the 'Drive'.



The following pages show a list of Safety related NC machine data with it's Drive equivalent parameters.

This is not a complete listing of the 'Axis' safety related data, but a list that shows the 'Axis' MD that has an equivalent safety related 'Drive' MD.

With the exception of the highlighted parameters, all of the values from the NC data are copied to the respective drive data when the afore mentioned 'Copy SI Data' function is used within the 'Setup' >> 'Safety' commissioning area.

The encoder related data in MD's 36916 to 36922 (Highlighted), are not automatically copied and must be entered manually in both the Axis and Drive MD. This is necessary in the event of a two encoder system, where the motor encoder is monitored and used by the drive safety, whilst the direct (e.g. Linear scale) will be used by the NC safety.

Notes :

List of NC Safety parameters that have equivalent drive Safety parameters that are copied automatically with the 'Copy SI Data' softkey in the 'Setup' >> 'Safety' commissioning area.

Note: Highlighted parameters 36916/p9516 to 36922/p9522 must be entered manually within NC and drive.

NC MD	Name	Drive MD	Name
10089	$MN_SAFE_PULSE_DIS_TIME_BUSFAIL	p9580	SI motion pulse suppression delay bus failure
10090	$MN_SAFETY_SYSCLOCK_TIME_RATIO	p9500	SI motion monitoring clock cycle
36901	$MA_SAFE_FUNCTION_ENABLE	p9501	SI motion enable safety functions
36902	$MA_SAFE_IS_ROT_AX	p9502	SI motion axis type
36903	$MA_SAFE_CAM_ENABLE	p9503	SI motion SCA (SN) enable
36905	$MA_SAFE_MODULO_RANGE	p9505	SI motion SCA (SN) modulo value
36916	$MA_SAFE_ENC_IS_LINEAR	p9516	SI motion, motor encoder configuration, safety--related functions
36917	$MA_SAFE_ENC_GRID_POINT_DIST	p9517	SI motion linear scale, grid division
36918	$MA_SAFE_ENC_RESOL	p9518	SI motion encoder pulses per revolution
36919	$MA_SAFE_ENC_PULSE_SHIFT	p9519	SI motion fine resolution G1_XIST1
36920	$MA_SAFE_ENC_GEAR_PITCH	p9520	SI motion spindle pitch
36921	$MA_SAFE_ENC_GEAR_DENOM[n]	p9521	SI motion, gearbox, encoder/load, denominator
36922	$MA_SAFE_ENC_GEAR_NUMERA[n]	p9522	SI motion, gearbox, encoder/load, numerator
36930	$MA_SAFE_STANDSTILL_TOL	p9530	SI motion standstill tolerance
36931	$MA_SAFE_VELO_LIMIT[n]	p9531	SI motion SLS (SG) limit values
36932	SAFE_VELO_OVR_FACTOR[n]	p9532	SI motion SLS (SG) override factor
36934	MA_SAFE_POS_LIMIT_PLUS[n]	p9534	SI motion SLP (SE) upper limit values
36935	$MA_SAFE_POS_LIMIT_MINUS[n]	p9535	SI motion SLP (SE) lower limit values
36936	$MA_SAFE_CAM_POS_PLUS[n]	p9536	SI motion SCA (SN) plus cam position
36937	$MA_SAFE_CAM_POS_MINUS[n]	p9537	SI motion, SCA (SN) minus cam position
36938	$MA_SAFE_CAM_TRACK_ASSIGN	p9538	SI motion SCA (SN) cam track assignment
36940	$MA_SAFE_CAM_TOL	p9540	SI motion SCA (SN) tolerance
36942	$MA_SAFE_POS_TOL	p9542	SI motion actual value comparison tolerance (crosswise)
36944	$MA_SAFE_REFP_POS_TOL	p9544	SI motion actual value comparison tolerance (referencing)
36945	$MA_SAFE_VELO_X_FILTER_TIME	p9545	SI Motion SSM (SGA n < nx) filter time
36946	$MA_SAFE_VELO_X	p9546	SI Motion SSM (SGA n < nx) velocity limit n_x
36947	$MA_SAFE_VELO_X_HYSTERESIS	p9547	SI motion SSM (SGA n < nx) velocity hysteresis
36948	$MA_SAFE_STOP_VELO_TOL	p9548	SI motion SBR actual speed tolerance
36949	$MA_SAFE_SLIP_VELO_TOL	p9549	SI motion slip speed tolerance
36950	$MA_SAFE_MODE_SWITCH_TIME	p9550	SI motion SGE changeover tolerance time
36951	$MA_SAFE_VELO_SWITCH_DELAY	p9551	SI motion SGE changeover delay time
36952	$MA_SAFE_STOP_SWITCH_TIME_C	p9552	SI motion transition time STOP C to SOS (SBH)
36953	$MA_SAFE_STOP_SWITCH_TIME_D	p9553	SI motion transition time STOP D to SOS (SBH)
36954	$MA_SAFE_STOP_SWITCH_TIME_E	p9554	SI motion transition time STOP E to SOS (SBH)
36955	$MA_SAFE_STOP_SWITCH_TIME_F	p9555	SI motion transition time STOP F to STOP B
36956	MA_SAFE_PULSE_DISABLE_DELAY	p9556	SI motion pulse cancelation delay time
.....	.....	.....	.....

Notes :

NC MD	Name	Drive MD	Name
……..	……..	……..	……..
36957	$MA_SAFE_PULSE_DIS_CHECK_TIME	p9557	SI motion pulse cancellation checking time
36958	$MA_SAFE_ACCEPTANCE_TST_TIMEOUT	p9558	SI motion acceptance test time limit
36960	MA_SAFE_STANDSTILL_VELO_TOL	p9560	SI motion pulse cancelation shutdown speed
36961	$MA_SAFE_VELO_STOP_MODE	p9561	SI motion SLS (SG) stop response
36962	$MA_SAFE_POS_STOP_MODE	p9562	SI motion SLP (SE) stop response
36963	$MA_SAFE_VELO_STOP_REACTION[n]	p9563	SI motion SLS (SG)--specific stop response
36997	MA_SAFE_ACKN	r9727	SI motion, internal drive user agreement
36998	$MA_SAFE_ACT_CHECKSUM	r9728	SI motion, actual checksum, SI parameters
36999	$MA_SAFE_DES_CHECKSUM	p9729	SI motion, reference checksum, SI parameters
36969	$MN_SAFE_BRAKETEST_TORQUE_NORM	p2003	Reference torque

The parameters highlighted above are not copied from the NC to the Drive with the 'Setup' >> 'Safety' >> 'Copy SI Data' function.

MD 36977/r9727 and 36969/p2003 will be described later.

'Actual' checksum values of all safety related data values are created on power up of the control or following an NCK reset.

Checksum values are stored in 'General MD', 'Axis MD' and also within 'Drive parameters'.

General MD

MD13318[0..3] contains **'ACTUAL'** checksum values for all 'Global' (non-axis/drive specific) safety related data.

13318[0]	$MN_SAFE_GLOB_ACT_CHECKSUM	76A2AC89H
13318[1]	$MN_SAFE_GLOB_ACT_CHECKSUM	E6F5AA52H
13318[2]	$MN_SAFE_GLOB_ACT_CHECKSUM	F03D44ADH
13318[3]	$MN_SAFE_GLOB_ACT_CHECKSUM	11ADBH
13319[0]	$MN_SAFE_GLOB_DES_CHECKSUM	0H
13319[1]	$MN_SAFE_GLOB_DES_CHECKSUM	0H
13319[2]	$MN_SAFE_GLOB_DES_CHECKSUM	0H
13319[3]	$MN_SAFE_GLOB_DES_CHECKSUM	0H

Index [0] of MD 13318 contains the checksum values for MD's relating to 'Safe communication', 'SPL I/O connection', 'SPL functionality'….etc.
E.g.

10385[n] $MN_PROFISAFE_MASTER_ADDRESS
10386[n] $MN_PROFISAFE_IN_ADDRESS
10387[n] $MN_PROFISAFE_OUT_ADDRESS
10388[n] $MN_PROFISAFE_IN_ASSIGN
10389[n] $MN_PROFISAFE_OUT_ASSIGN

Index [1] of MD 13318 contains the checksum values for MD's relating to 'SPL User Data'.
E.g.

13312 $MN_SAFE_SPL_USER_DATA

Index [2] of MD 13318 contains the checksum values for MD's relating to 'Profisafe I/O Connection Enable Data'.
E.g.

13302 $MN_PROFISAFE_IN_ENABLE_MASK
13303 $MN_PROFISAFE_OUT_ENABLE_MASK

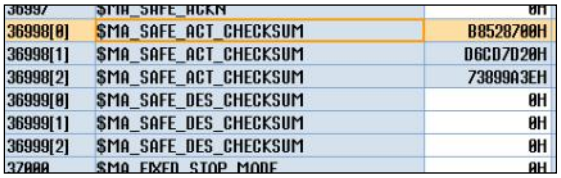
Index [3] of MD 13318 contains the checksum values for MD's relating to the Step7 Profisafe hardware configuration.
E.g.

Data provided by the Step7 F-Configuration add-on package.

Notes :

Axis MD

MD36998[0..2] contains **'ACTUAL'** checksum values for all 'Axis' specific safety related data.



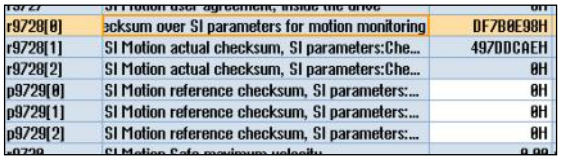
Index [0] of MD 36998 contains the checksum values for MD's relating to 'Safe motion monitoring'
E.g.
36901 $MA_SAFE_FUNCTION_ENABLE
36902 $MA_SAFE_IS_ROT_AX
...etc

Index [1] of MD 36998 contains the checksum values for MD's relating to 'Sinamics Hardware'
E.g.
36927 $MA_SAFE_ENC_MOD_TYPE
...etc

Index [2] of MD 36998 contains the checksum values for MD's relating to 'Sinamics Coupling'
E.g.
36906 $MA_SAFE_CTRLOUT_MODULE_NO
36907 $MA_SAFE_DRIVE_PS_ADDRESS
36912 $MA_SAFE_ENC_INPUT_NO

Drive MD

p9728[0..2] contains **'ACTUAL'** checksum values for all 'Drive' specific safety related data.

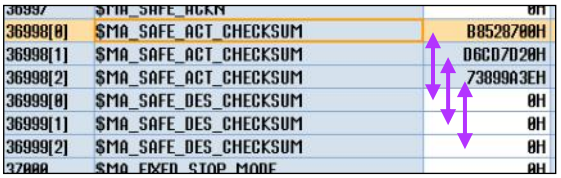


Index [0] of p9728 contains the checksum values for MD's relating to 'Safe motion monitoring'
E.g.
p9501 SI motion enable safety functions
p9502 SI motion axis type
...etc

Index [1] of p9728 contains the checksum values for MD's relating to 'Safe gearing data' (Leadscrew pitch, ENC resolution..etc
E.g.
p9516 to p9522 (i.e. All drive related data not included in the 'Copy SI' function.

Index [2] of p9728 contains the checksum values for MD's relating to 'Safe encoder actual value configuration' (Hardware)
E.g.
p9515 SI motion coarse position value configuration

The Safe Actual Checksum Data's are updated at power-on and following an NCK reset. When 'System Based Safety' is enabled, the system will compare the values of the 'Actual' checksum data against the 'Desired' checksum data values.



If any safety related data is changed, the actual checksum value will then differ from the 'Desired' checksum value and the following errors will appear:

Global Checksum Data:

MD 13318[0] <> MD 13319[0]
27070 "Checksum error, SPL parameter assignment, and SPL interfaces. Confirmation and acceptance test are required!"

MD 13318[1] <> MD 13319[1]
27071 "Checksum error, safe SPL parameterization. Confirmation and acceptance test are required"

MD 13318[2] <> MD 13319[2]
27072 "Checksum error, enabling safe communication. Confirmation and acceptance test required"

MD 13318[3] <> MD 13319[3]
27073 "Checksum error, S7 PROFIsafe configuration. Confirmation and acceptance test required.

Axis Specific Checksum Data:

MD 36998[0] <> MD 36999[0]
27032 "Axis %1 checksum error of safe monitoring. Confirmation and acceptance test are required!"

MD 36998[1] <> MD 36999[1]
27035 "Axis %1 new HW component, Acknowledgement and function test required"

MD 36998[2] <> MD 36999[2]
27060 "Axis %1 checksum error of drive assignment. Confirmation and acceptance test required"

Notes :

Drive Specific Checksum Data:

p9728[n] <> p9729[n]

201680 (F01680) SI Motion CU: Checksum error safety monitoring functions

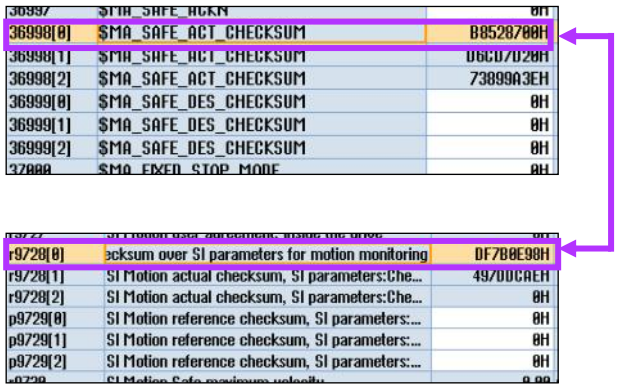
Fault value (r0949, interpret decimal):

0: Checksum error for SI parameters for motion monitoring.

1: Checksum error for SI parameters for actual values.

2: Checksum error for SI parameters for component assignment.

The safety system not only checks the equality of the 'Actual' and 'Desired' safe checksum data, it also checks that the 'Safe Motion Monitoring' (Index 0 of MD 36998/9 & r9728/9) checksum values within the Axis and Drive are also equal (subject to an internal offset/algorithm).



Error '27001 error in a monitoring channel, code %2, values: NCK %3, drive %4' will be displayed. Code %2 will describe the specific safe machine data that has changed in the axis or drive only.

E.g. Code = 5

Function enables MD36901 MA_SAFE_FUNCTION_ENABLE.

It is clear that the 'Desired' checksum values must be the same as the 'Actual' checksum values within the Global data (General Machine Data), Axis specific data and Drive specific safe data.

This is also true for the 'Safe Motion Monitoring' checksum data's between each axis and drive.

In order to easily and conveniently set the 'Desired' checksum values to the 'Actual' values, we can use the 'Setup' >> 'Safety' >> 'Confirm SI' function within HMI Operate.

Using this function will automatically calculate and set the 'Desired' checksum machine data and parameters.

Following a power cycle or NCK reset, the 'Actual' and 'Desired' checksum values should then be equal and the 'Safe Motion Monitoring' checksums (Axis/Drive index 0) should also be equal (subject to an internal offset/algorithm). i.e. all 'Safe Motion Monitoring' data value must be identical in axis and drive data.

For example, if a safety related data value is changed for an axis, e.g. MD 36901 $MA_SAFE_FUNCTION_ENABLE, this would result in the 'Actual' checksum data for index [0] of 36998 being different to index [0] of 36999.

This would result in error 27032 "Axis %1 checksum error of safe monitoring. Confirmation and acceptance test are required!".

If a 'Confirm SI Data' was executed, then error '27001 error in a monitoring channel, code %2, values: NCK %3, drive %4' would be displayed with Code 5. This is because the data in the Axis MD 36901 is not equal to the value in Drive parameter p9501.

So, the normal procedure when changing any safe data is to first 'Copy SI Data' then 'Confirm SI Data'.

This should help to avoid checksum errors during commissioning.

This checking method will also provide protection against illegal changes to safety related data in Machine Data, Drive parameters or changes to the Step7 Safe Hardware Configuration once commissioning has been completed.

Notes :

F-Motion Monitoring:

Enabling Axis Safety Functions

This section describes the following;

- Enable SI Options
- Enabling of SI Function for each axis
- Activation of Drive Startup
    - SI Encoder Matching
    - SBR (SAFE Braking Ramp) calculation
- Safe Brake Control
- Safe Time Values (STOP D Ramp, Velocity Switcing times...etc
- Safe Internal Input / Output Assignment to SAFE MD
- Safe Internal Input / Output Assignment to NCK-SPL
- Safe Internal Input / Output Assignment to PLC-SPL
- Safety Related Drive Interface (Axis DB)
- SAFE_DES_VELO_LIMIT and SBH (Safe Operational Stop)
- Check Polarity of Safe Actual Value
- Application of the Function Table
- Setting Safe Velocity Limits
- Safe End Limits and Safe Software Cams
    - User Agreement
    - Safe Reference Position Tolerance
    - Reducing Velocity when not Safely Referenced

NOTE: Any NC/PLC code shown in this document is to be treated as an example and for guidance purposes only!

Notes :

SI Options

In this section we will enable the safe motion monitoring functions for each axis.

Additional licenses will also need to be enabled for safe axes 2..n. The M64 option (SI Comfort) includes 1 safe axis within its scope.

Each C70 option allows for Safe Motion Monitoring functions to be enabled on additional axes/spindles.

Enabling of SI Function

The 'Safe Motion Monitoring' functions for each axis/spindle can be enabled using

MD36901 $MA_SAFE_FUNCTION_ENABLE.

This machine data is a bit mask that allows for specific safe functions to be enabled on a per axis/spindle basis.

Bit 0: Enable safely--reduced speed, safe operating stop
Bit 1: Enable safe limit switch
Bit 2: Reserved for functions with absolute reference (such as SE/SN)
Bit 3: Enable actual value synchronization, 2--encoder system
Bit 4: Enable external ESR activation (STOP E)
Bit 5: Enable SG correction
Bit 6: Enable the external stop requests / external STOPs
Bit 7: Enable cam synchronization
Bit 8: Enable safe cam, pair 1, cam+
Bit 9: Enable safe cam, pair 1, cam--
Bit 10: Enable safe cam, pair 2, cam+
Bit 11: Enable safe cam, pair 2, cam--
Bit 12: Enable safe cam, pair 3, cam+
Bit 13: Enable safe cam, pair 3, cam--
Bit 14: Enable safe cam, pair 4, cam+
Bit 15: Enable safe cam, pair 4, cam--
Bit 16: Enable synchronization (hysteresis and filtering "n<nx"

If bit 1 or a higher bit is set, then bit 0 must also be set since the control system switches to a safe operational stop in response to STOP C, D or E (a parameterizing alarm 27033 is displayed if an error is detected).

If an insufficient number of axes/spindles have been enabled for safe operation using the M64/C70 options, then this data may be overwritten with the value 0000 when booting.

The minimum value for enabling safety on each axis is **'41H'**.

This is...

Bit 0: Enable safely--reduced speed, safe operating stop

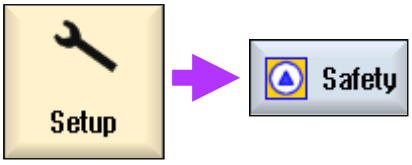
...and

Bit 6: Enable the external stop requests / external STOPs

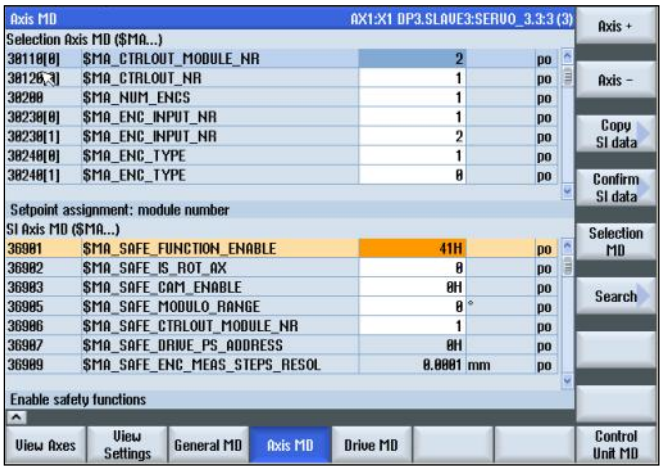
These bits should now be set for each **'safe'** axis. For the purpose if this course, the safe axes will be **X1**, **Y1**, **A1**, **C1** and the Spindle **SP1**. The Z1 axis is a simulation axis.

Note: It may be easier to enable the safe functions one axis at a time, repeating the process for each axis once the basic settings have been applied.

Using the 'Safety' commissioning screens of the 'Setup' area...

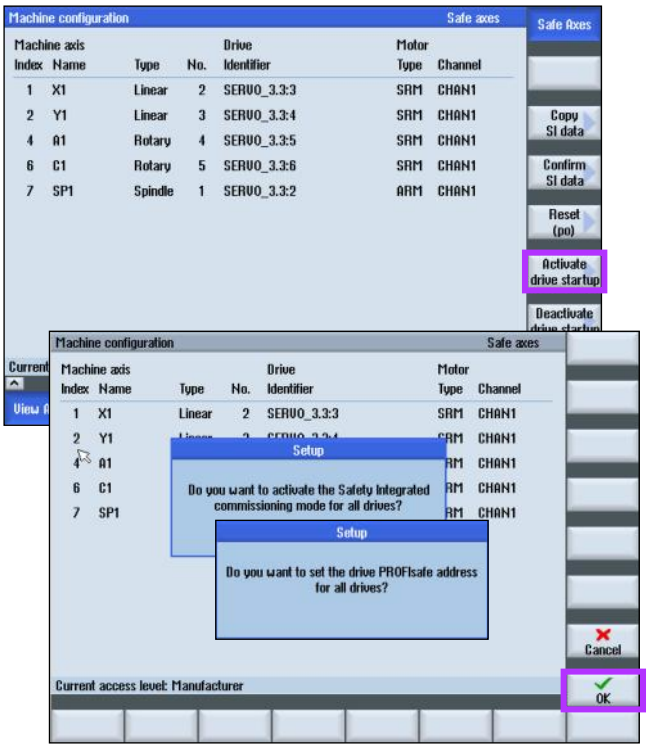


...select the 'Axis MD' page and set MD36901 to 41H for 1 or more safe axes.



Notes :

Activation of Drive Startup.

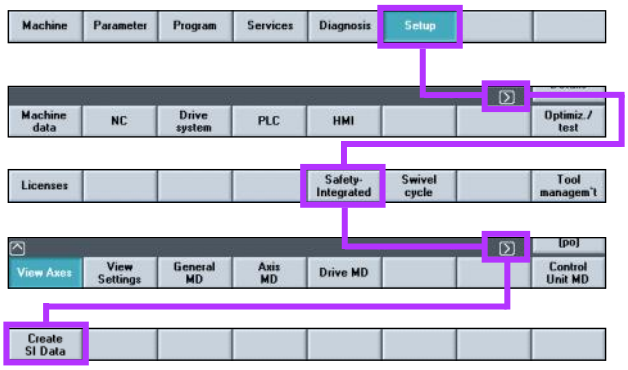


p10 = 95 (SI commissioning Mode), and a unique Profisafe drive address in p9810 has now been set for each safe axis.

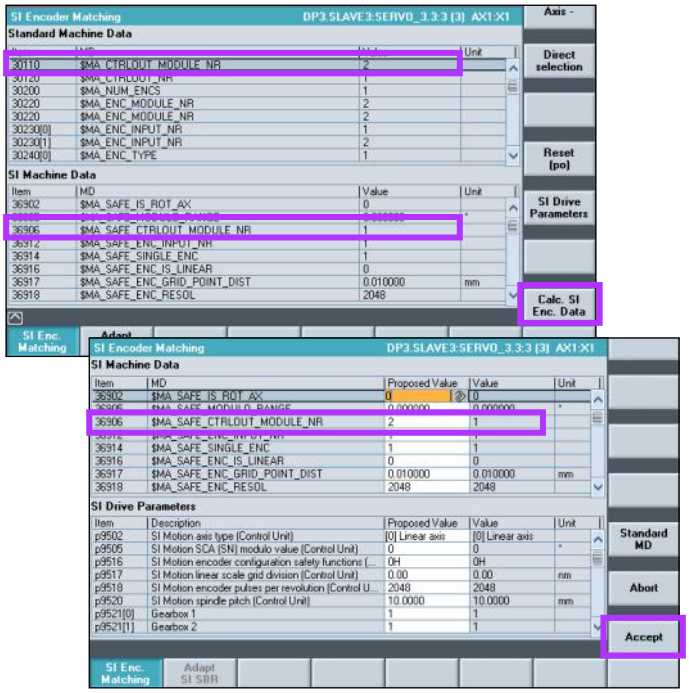
The next step requires the use of 'HMI Advanced V7.6' or the 'Startup(IBN) Tool V7.7'.

The functionality for completely starting-up Safety Integrated is not yet included within HMI Operate.

Select the 'Safety Integrated' area within 'Setup'.



Using the 'SI Enc. Matching' screen the safe encoder settings for both the NC(Axis MD) and the Drive are set automatically based on the non safe encoder settings.

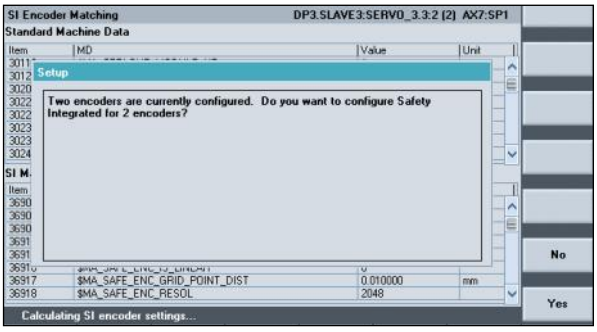


As you can see, the encoder data has now been copied to the safe encoder data in both the NC and Drive for the selected axis.

E.g.
'MD 30110 $MA_CTRLOUT_MODULE_NR' is copied to 'MD 36906 $MA_SAFE_CTRLOUT_MODULE_NR'.

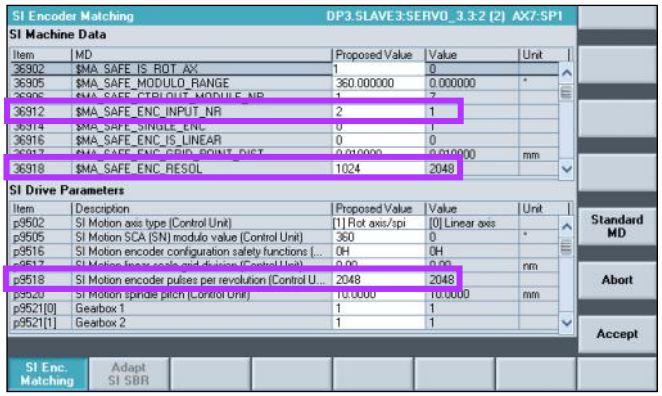
If the axis/spindle has two encoders (i.e. direct measuring system), then the direct measuring system can easily be assigned as the NC Safe Encoder. The motor encoder is always assigned as the safe encoder within the Drive.

Here you can see the dialog that appears for the Spindle.



Notes :

Notice the 'Proposed' values are now different in the NC and Drive safe data.

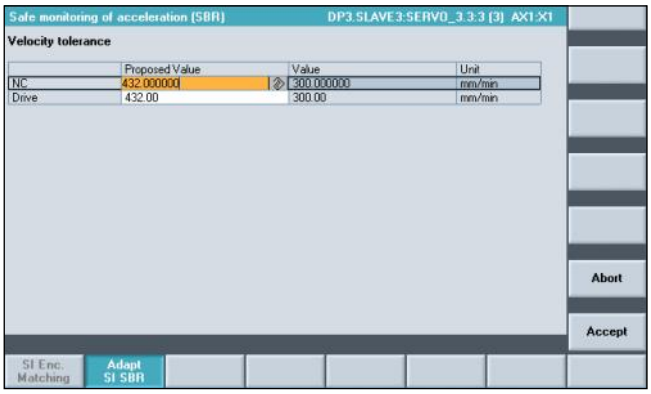


Pressing 'Accept' will write the data and automatically save the Drive data.

Ensure that this procedure is carried out for each axis/spindle in turn.

It is also possible to not use this method and the data can be set manually using HMI Operate. Using HMI Advanced reduces the chance of an error being introduced.

Using HMI Advanced, the SBR (Safe Braking Ramp) can also be calculated for all axes.

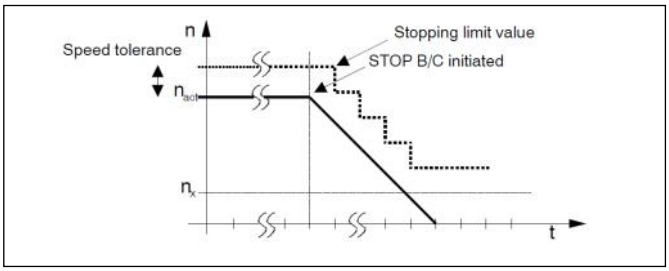


MD '36948 $MA_SAFE_STOP_VELO_TOL and drive' Parameter 'p9548 SI Motion SAM actual velocity tolerance' are both determined using the following formula;

SBR Tolerance = Acceleration * Safe Clock Cycle

Linear Axes: SBR[mm/min] =
$a[m/s^2] * t_{safe} * 1000[mm/min] * 60[s/min]$

Linear Axes: SBR[rev/min] =
$a[rev/s^2] * t_{safe} * 60[s/min]$



The following machine data should be taken into account when determining the acceleration:

MD 32300: MAX_AX_ACCEL
MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL
MD 35210: GEAR_STEP_POSCTRL_ACCEL
MD 35410: SPIND_OSCILL_ACCEL

Recommendation:
The value entered for the SBR tolerance should be approx. 20% higher than the calculated value.

Machine data/parameters for the SBR speed tolerance:

MD 36948: $MA_SAFE_STOP_VELO_TOL
p9548: SI motion SBR actual speed tolerance (Control Unit)

The speed limit is corrected until the speed, is less than that defined in the following machine data. After that, the limit value of the SBR monitoring is frozen to the value in MD/parameter 36946/p9546 plus the value in MD/parameter 36948/p9548.

MD 36946: $MA_SAFE_VELO_X (speed limit n<nx)
p9546: SI motion SSM (SGA n < nx) speed limit (CU)

E.g. Calculation of the X axis SBR:

Linear Axes: SBR[mm/min] =
$a[m/s^2] * t_{safe} * 1000[mm/min] * 60[s/min]$

MD 10091 $MN_INFO_SAFETY_CYCLE_TIME = 0.006s
MD 32300 $MA_MAX_AX_ACCEL = $1m/s^2$

SBR Tol = (0.006 * 1 * (60*1000)) + 20%
= **432mm/min**

Notes :

Now adapt the SBR for all axes and Spindle.

Note:
Any changes to the following machine data's will then require the SBR to be re-calculated;

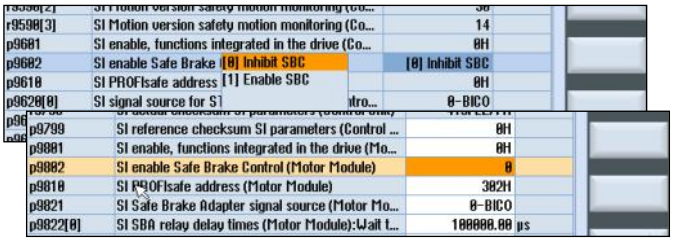
MD 10050: SYSCLOCK_CYCLE_TIME
MD 10090: SAFETY_SYSCLOCK_TIME_RATIO
MD 32300: MAX_AX_ACCEL

(For spindles)
MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL
MD 35210: GEAR_STEP_POSCTRL_ACCEL
MD 35410: SPIND_OSCILL_ACCEL

Note: From this point on, we no longer require the use of HMI Advanced or the Startup(IBN) tool.

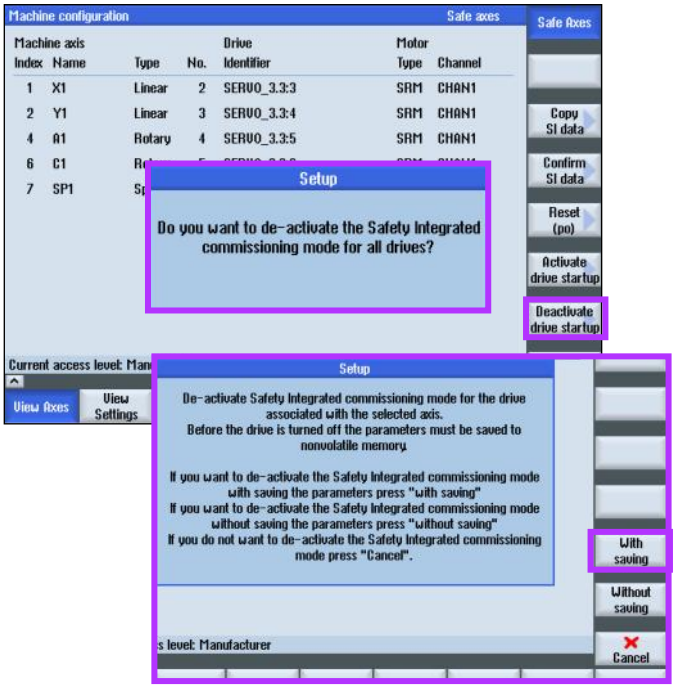
Safe Brake Control:

It makes sense to enable SBC (Safe Brake Control) for all safe axes that have a braked motor where the brake is controlled by Sinamics.
By setting p9602 = 1 & p9802 = 1, SBC is enabled for that axis. No other parameter settings are required. P/M isolation of the motor holding brake will now be enabled.



SBC should not be enabled for motors that do not have a holding brake.

We can now deactivate drive startup mode (p10 = 95 >> 0).



Exit the 'Safety Integrated Commissioning Mode' and Save the drive data.

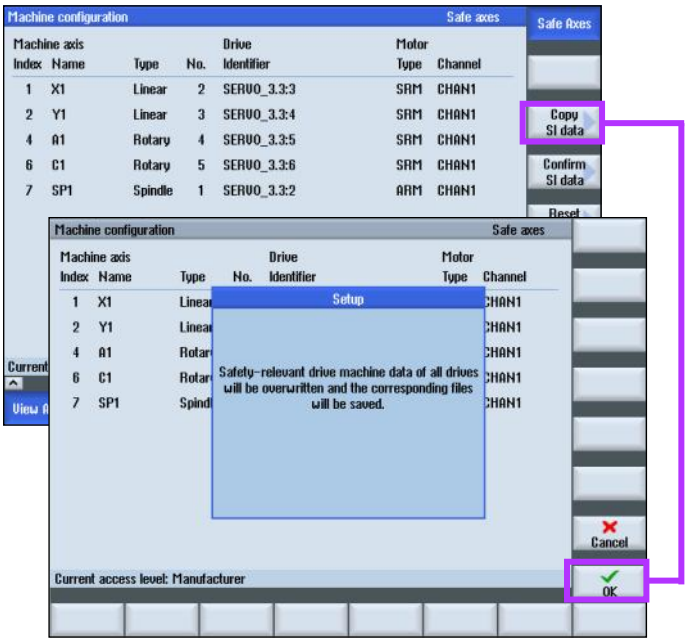
We know from the last section that any changes to 'Safety Related Machine Data' must be 'Confirmed' in order to generate the 'Desired' checksum values.

If the data is related to 'Safe Motion Monitoring' (as it is in this case) then the safe data must also be copied to the 'Drive MD area'.

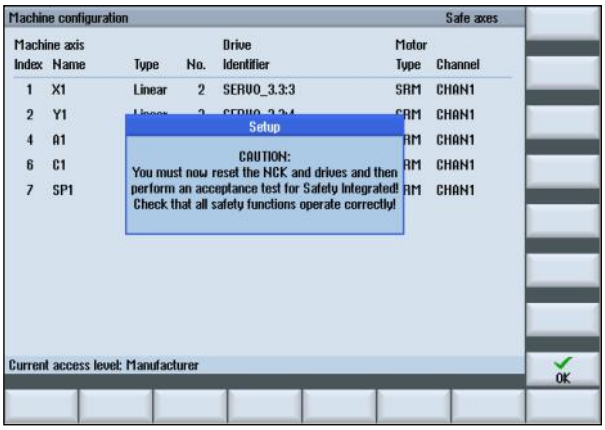
Note: When using the 'Copy SI data' or 'Confirm SI data' functions, be aware that if the 'Axis MD' page is active, then the data will only be copied/confirmed for the displayed axis. When the 'View Axes' screen is displayed, all safe axes will be copied/confirmed.

Notes :

Now use the ‘Copy SI data’ function and acknowledge the dialog informing you that the ‘Safety Relevant Data’ will be copied.

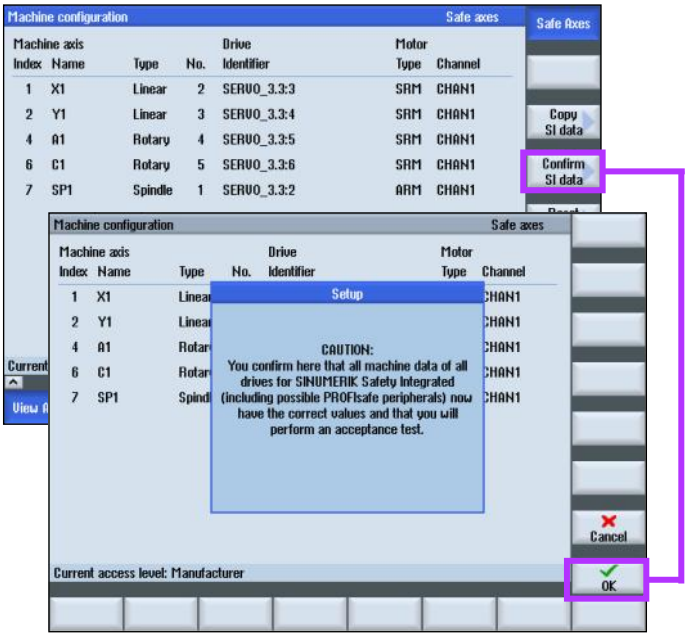


Once the safe data has been copied from the Axis MD to the Drive parameters, the drive data is automatically saved and the following dialogue will appear.



During the commissioning phase, we can ignore this request for an ‘acceptance test’. Although we must perform an NCK reset to activate the changes.

Now carry out a ‘Confirm SI data’.



Using the ‘Confirm SI data’ function will now update the ‘Desired’ checksum values in MD’s 13319, 36999 and p9729.

Once again, the drive data is automatically saved. Now perform an NCK ‘Reset(po)’

Once the NCK reset has completed, the following 3 alarms should be present;

27097 ‘SPL Start not executed’

22001 Axis <??>: Brake ramp longer than Stop D time:
Reason : 1

27033 Axis <??> parameterization of MD
$MA_SAFE_EXT_STOP_INPUT[0] invalid

Error 27097 is there due to error 27033. We will address this error later.

Alarm 22001 is due to
MD 36953 SAFE_STOP_SWITCH_TIME_D being less than the time it would take for the axis to decelerate from maximum velocity to standstill including jerk.

STOP_D is a safe stop request using the position control loop acceleration profile (e.g. same profile as for NC Stop, Feed-hold...etc).

The following machine data must be taken into account when calculating a value for MD 36953;

MD 32000 MAX_AX_VELO
MD 32300 MAX_AX_ACCEL
MD 32431 MAX_AX_JERK

Notes :

Safe Time Values

Note: For MD's with multiple indexes, always use the largest value.

For linear axes, a suitable value for MD 36953 can be calculated as follows;

MD 32000 MAX_AX_VELO = 10000mm/min
MD 32300 MAX_AX_ACCEL = 1m/s^2
MD 32431 MAX_AX_JERK = 1000000m/s^3

MD 36953 = ((MD 32000/MD 32300) / (1000*60) + (MD 32300/ MD 32431)^2) + 20%

MD 36953 SAFE_STOP_SWITCH_TIME_D = 0.2s

For rotary axes use the formula;

MD 32000 MAX_AX_VELO = 100rpm
MD 32300 MAX_AX_ACCEL = 10rev/s^2
MD 32431 MAX_AX_JERK = 2777777m/s^3

MD 36953 = ((MD 32000/MD 32300) / (60) + (MD 32300/MD 32431)^2) + 20%

Rotary axes MD 36953 = 0.2s

For spindles, the following machine data must be considered;

MD 32000 MAX_AX_VELO = 1000rpm
MD 32300 MAX_AX_ACCEL = 30m/s^2

MD 35130 GEAR_STEP_MAX_VELO_LIMIT = 1000rpm
MD 35200 GEAR_STEP_SPEEDCTRL_ACCEL = 30m/s^2

MD 35135 GEAR_STEP_PC_MAX_VELO_LIMIT = 1000rpm
MD 35210 GEAR_STEP_POSCTRL_ACCEL = 30m/s^2

MD 35135 GEAR_STEP_PC_MAX_VELO_LIMIT = 1000rpm
MD 35212 SPEEDCTRL_ACCEL2 = 30m/s^2

The worst case (slowest acceleration/deceleration) of the above three pairs of machine data must be used for the calculation.

MD 36953 = ((MD 32000/MD 32300) / (60)) + 20%
or
MD 36953 = ((MD 35130/MD 35200) / (60)) + 20%
or
MD 36953 = ((MD 35135/MD 35210) / (60)) + 20%
or
MD 36953 = ((MD 35135/MD 35212) / (60)) + 20%

Spindle MD 36953 = 0.667s

MD 36952 SAFE_STOP_SWITCH_TIME_C should be set to 20% higher than 'p1135 Off 3 ramp time'. The default value for p1135 is 0s. In this case, set MD36953 to no higher than MD 36953. e.g. MD 36952 = 0.2s and reduce after testing later.

STOP_C is a drive specific stop request at maximum current. E.g. Emergency Stop.

Axes MD 36952 SAFE_STOP_SWITCH_TIME_C = 0.2s
Spindle MD 36952 SAFE_STOP_SWITCH_TIME_C = 0.667s

MD 36951 SAFE_VELO_SWITCH_DELAY can also be set using the same calculation values as used for MD 36953 SAFE_STOP_SWITCH_TIME_D.
This MD allows for transition from one safe velocity to another without error.

Axes MD 36951 SAFE_VELO_SWITCH_DELAY = 0.2s
Spindle MD 36951 SAFE_VELO_SWITCH_DELAY = 0.667s

Note: The calculations for these machine data's are 'Recommendation Values Only' They provide a 'Starting Point' and they should be increased or decreased as needed. You should take care in testing that the values will be sufficient for maximum velocity.

SAFE_STOP_SWITCH_TIME_C
SAFE_STOP_SWITCH_TIME_D
SAFE_VELO_SWITCH_DELAY

These machine data values must be set as low as possible in order to ensure safe operation of the machine.

Once the above machine data's have been assigned, 'Copy and confirm SI data' then NCK Reset to activate the changes.

Notes :

Safe Internal Input / Output Assignment to NC MD

The error '22001 Axis <??>: Brake ramp longer than Stop D time: Reason : 1' should no longer be displayed and the following two errors will remain;

27097 'SPL Start not executed'

27033 Axis <??> parameterization of MD $MA_SAFE_EXT_STOP_INPUT[0] invalid.

Error 27033 is displayed as
MD 36977 $MA_SAFE_EXT_STOP_INPUT[0] = 0H (Default). This machine data is used to assign an internal output ($A_OUTSI) to the axis. The assigned $A_OUTSI can then be controlled by the NCK-SPL (Safe Programmable Logic - SAFE.SPF).

In fact, all safe machine data '$MA_SAFE_.....**_INPUT'** will be assigned to a safety related internal output ($A_OUTSI[1..n]) if required.

Conversely, all safe machine data '$MA_SAFE_.....**_OUTPUT'** will be assigned to a safe internal input ($A_INSI[1..n]) if required.

These '$MA_SAFE_.....**_INPUT'** and '$MA_SAFE_.....**_OUTPUT'** machine data's provide the interface for selecting safe functions between the NCK-SPL and each axis or spindle.

Before the error 27033 is resolved, it is important to understand the assignment of these machine data's.

The $A_OUTSI and $A_INSI safe system variables are assigned to the SAFE_...._INPUT' & 'SAFE_...._OUTPUT' machine data's as follows using an eight-digit hexadecimal number.

Coding of the input assignment = is mm xx nn
i = Inversion
0: No inversion
8: Inversion before processing

s = Segment No.
4: Internal image in the system memory (system variable)

mm = Module No. 01--02
01: Addressing the internal SPL interface $A_OUTSI/$A_INSI
02: Addressing the external SPL interface (only for input signals, $A_INSE)

xx = Submodule No. 01--06 Index of the system variable word (each 32 bit)

nn = I/O No. 01--20 Bit number in the system variable word $A_OUTSID[xx], $A_INSED[xx]
If several output signals are set, then the signal involved is first inverted. The (in some cases inverted) output signals are then AND'ed and the result is output at the terminal.

e.g. $A_OUTSI[10] = '0401010AH'
$A_INSI[33] = '04010201H'

The required '$MA_SAFE_.....**_INPUT'** machine data's for each axis and spindle must now be assigned to $A_OUTSI system variables.

For the purposes of this course, the following assignments will be made;

All Safe Axes;
36970 SAFE_SVSS_DISABLE_INPUT = 0H = SG or SBH Active
36971 SAFE_SS_DISABLE_INPUT = 04010101 = $A_OUTSI[1]
36972 SAFE_VELO_SELECT_INPUT[0] = 04010102 = $A_OUTSI[2]
36972 SAFE_VELO_SELECT_INPUT[1] = 04010103 = $A_OUTSI[3]
36977 SAFE_EXT_STOP_INPUT[0] = 04010104 = $A_OUTSI[4]
36977 SAFE_EXT_STOP_INPUT[1] = 04010105 = $A_OUTSI[5]
36977 SAFE_EXT_STOP_INPUT[2] = 04010106 = $A_OUTSI[6]
36977 SAFE_EXT_STOP_INPUT[3] = 0H = No Assignment

Spindle;
36970 SAFE_SVSS_DISABLE_INPUT =0H = SG or SBH Active
36971 SAFE_SS_DISABLE_INPUT =04010107 =$A_OUTSI[7]
36972 SAFE_VELO_SELECT_INPUT[0] =04010102 =$A_OUTSI[2]
36972 SAFE_VELO_SELECT_INPUT[1] =04010103 =$A_OUTSI[3]
36977 SAFE_EXT_STOP_INPUT[0] =04010108 = $A_OUTSI[8]
36977 SAFE_EXT_STOP_INPUT[1] =04010109 = $A_OUTSI[9]
36977 SAFE_EXT_STOP_INPUT[2] =0401010A = $A_OUTSI[10]
36977 SAFE_EXT_STOP_INPUT[3] =0H = No Assignment

Now 'Confirm SI data' then NCK Reset to activate the changes.

Machine data descriptions:

36970 SAFE_SVSS_DISABLE_INPUT
Signal 0 = SG **OR** SBH is **Selected** (active)
Signal 1 = SG **AND** SBH is **De-selected** (not active)

36971 SAFE_SS_DISABLE_INPUT
Signal 0 = SBH is **Selected** (active) if 36970 = 0
Signal 1 = SG is **Selected** (active) if 36970 = 0

36970 & 36971 work as a pair and are used to select SBH (SOS - Safe Operational Stop) or SG(SLS - Safe Limited Speed/Safe Reduced Velocity).
36971 signal level is irrelevant if 36970 = 1

36972 SAFE_VELO_SELECT_INPUT[0..1]

Index [0]	Index[1]	SG Level
0	0	1
0	1	2
1	0	3
1	1	4

36972[0..1] are used to select the four different safe velocity levels.

Notes :

MD 36977 $MA_SAFE_EXT_STOP_INPUT[0..3]

Index[0] = STOP A - SH/STO (Safe Pulse Disable)
Index[1] = STOP C - Brake at current limit
Index[2] = STOP D - Brake on acceleration path
Index[3] = STOP E - ESR + Brake on acceleration path

All indexes require a signal level = 1 to De-Select the STOP. (If STOP E is not enabled (MD36901 bit 4), Index[3] can re-main = 0)

For STOP C, D & E the drive remains in closed position loop control.

STOP A has the highest priority followed by STOP C, then STOP D...etc.

For the purpose of this course, all of the axes have been assigned the same $A_OUTSI system variables. This means that the selected safety functions such as 'Safe Velocity Level' and 'Safe External STOP's' will be common across all axes.

It is often required for the spindle to be in a different 'Safe' state to the axes. This is the reason that the Spindle has a different assignment of $A_OUTSI system variables for the SPL to manipulate.

All axes:

36978	$MA_SAFE_SVSS_DISABLE_INPUT	0H	po
36971	$MA_SAFE_SS_DISABLE_INPUT	4010101H	po
36972[0]	$MA_SAFE_VELO_SELECT_INPUT	4010102H	po
36972[1]	$MA_SAFE_VELO_SELECT_INPUT	4010103H	po
36973	$MA_SAFE_POS_SELECT_INPUT	0H	po
36974[0]	$MA_SAFE_GEAR_SELECT_INPUT	0H	po
36974[1]	$MA_SAFE_GEAR_SELECT_INPUT	0H	po
36974[2]	$MA_SAFE_GEAR_SELECT_INPUT	0H	po
36977[0]	$MA_SAFE_EXT_STOP_INPUT	4010104H	po
36977[1]	$MA_SAFE_EXT_STOP_INPUT	4010105H	po
36977[2]	$MA_SAFE_EXT_STOP_INPUT	4010106H	po
36977[3]	$MA_SAFE_EXT_STOP_INPUT	0H	po

Spindle:

36978	$MA_SAFE_SVSS_DISABLE_INPUT	0H	po
36971	$MA_SAFE_SS_DISABLE_INPUT	4010107H	po
36972[0]	$MA_SAFE_VELO_SELECT_INPUT	4010102H	po
36972[1]	$MA_SAFE_VELO_SELECT_INPUT	4010103H	po
36973	$MA_SAFE_POS_SELECT_INPUT	0H	po
36974[0]	$MA_SAFE_GEAR_SELECT_INPUT	0H	po
36974[1]	$MA_SAFE_GEAR_SELECT_INPUT	0H	po
36974[2]	$MA_SAFE_GEAR_SELECT_INPUT	0H	po
36977[0]	$MA_SAFE_EXT_STOP_INPUT	4010108H	po
36977[1]	$MA_SAFE_EXT_STOP_INPUT	4010109H	po
36977[2]	$MA_SAFE_EXT_STOP_INPUT	401010AH	po
36977[3]	$MA_SAFE_EXT_STOP_INPUT	0H	po

Safe Internal Input / Output Assignment to NCK-SPL

Now 'DEFINE' names for the $A_OUTSI system variables within the NCK-SPL (/CMA.DIR/SAFE.SPF).

Modify the 'DEFINE' section as follows;
(Or, within the 'Archives/Manufacturer/A102' folder, the archive '**A102_SAFE_002.ARC**' can be loaded to save time.)

```
DEFINE OE2_Spare          AS $A_OUTSE[2]
DEFINE OE3_Spare          AS $A_OUTSE[3]

;=======================================
;Internal Outputs - $A_OUTSI
DEFINE OI1_SBH_AX_DES     AS $A_OUTSI[1]
DEFINE OI2_SG_BIT0        AS $A_OUTSI[2]
DEFINE OI3_SG_BIT1        AS $A_OUTSI[3]
DEFINE OI4_STOPA_DES_AX   AS $A_OUTSI[4]
DEFINE OI5_STOPC_DES_AX   AS $A_OUTSI[5]
DEFINE OI6_STOPD_DES_AX   AS $A_OUTSI[6]
DEFINE OI7_SBH_SP_DES     AS $A_OUTSI[7]
DEFINE OI8_STOPA_DES_SP   AS $A_OUTSI[8]
DEFINE OI9_STOPC_DES_SP   AS $A_OUTSI[9]
DEFINE OI10_STOPD_DES_SP  AS $A_OUTSI[10]

;=======================================
;Markers - $A_MARKERSI
DEFINE MI1_SIREL1_IN1     AS $A_MARKERSI[1]
```

The newly defined $A_OUTSI system variables can now be used within the NCK-SPL.

Each $A_OUTSI will be controlled by individual synchronous actions. See IDS=207 onwards.

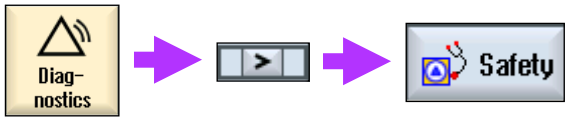
```
;Net 7:Assign $A_MARKERSI[6] to KA1_KA2 Contactor Output
IDS=206 DO OE1_KA1_KA2 = MI6_SIREL1_OUT0

;Net 8:Assign SBH for all axes
IDS=207 DO OI1_SBH_AX_DES = FALSE

;Net 9:Assign SBH for Spindle
IDS=208 DO OI7_SBH_SP_DES = FALSE

;Net 10: Assign SG bit0 for all axes & Spindle
IDS=209 DO OI2_SG_BIT0 = FALSE

;Net 11: Assign SG bit1 for all axes & Spindle
IDS=210 DO OI3_SG_BIT1 = FALSE

;Net 12: Assign all axes STOP A deselection to SIRELAY OUT1(1.0s delay)
IDS=211 DO OI4_STOPA_DES_AX = MI7_SIREL1_OUT1

;Net 13: Assign all axes STOP C deselection to SIRELAY OUT0(No delay)
IDS=212 DO OI5_STOPC_DES_AX = MI6_SIREL1_OUT0

;Net 14: Statically deselect all axes STOP D
IDS=213 DO OI6_STOPD_DES_AX = TRUE

;Net 15: Assign Spindle STOP A deselection to SIRELAY OUT2(2.0s delay)
IDS=214 DO OI8_STOPA_DES_SP = MI8_SIREL1_OUT2

;Net 16: Assign Spindle STOP C deselection to SIRELAY OUT0(No delay)
IDS=215 DO OI9_STOPC_DES_SP = MI6_SIREL1_OUT0

;Net 17: Statically deselect Spindle STOP D
IDS=216 DO OI10_STOPD_DES_SP = TRUE

M17
```

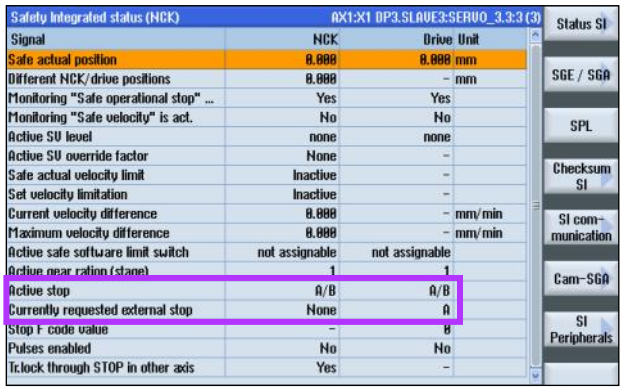
Notes :

Once the SAFE.SPF has been modified (or the 'Archives/ Manufacturer/A102' folder, archive '**A102_SAFE_002.ARC**' has been loaded. It should now be possible to reset the emergency stop.

Error '27090 Error in data cross check NCL-PLC, $A_OUTSI [x], NCK:1;' will be displayed but we can ignore this for now.

Using the 'Diagnostics' >> 'Safety' area, the status of the requested STOPs can be seen.





As you can see, when the Emergency Stop is reset, the NCK column has no 'Currently requested external stop'.

Notice that when the Emergency Stop is pressed on the HT2, the 'Currently requested external stop' changes from 'None' to 'C' then 1s later, it changes again to 'A'. 'STOP D' is never displayed here as it's statically deselected within the SAFE.SPF.

This is due to IDS=211, IDS=212 and IDS=213 in the SAFE.SPF.

```
;Net 12: Assign all axes STOP A deselection to SIRELAY OUT1(1.0s delay)
IDS=211 DO OI4_STOPA_DES_AX = MI7_SIREL1_OUT1

;Net 13: Assign all axes STOP C deselection to SIRELAY OUT0(No delay)
IDS=212 DO OI5_STOPC_DES_AX = MI6_SIREL1_OUT0

;Net 14: Statically deselect all axes STOP D
IDS=213 DO OI6_STOPD_DES_AX = TRUE
```

Using the 'Status SI' >> 'Select Axis' soft-keys, select the spindle and test the emergency stop again.

Notice the same behavior, except the time taken to go from 'C' to 'A' is longer at 2s. This longer delay is due to IDS=214 in the SAFE.SPF.

```
;Net 15: Assign Spindle STOP A deselection to SIRELAY OUT2(2.0s delay)
IDS=214 DO OI8_STOPA_DES_SP = MI8_SIREL1_OUT2
```

Safe Internal Input / Output Assignment to PLC-SPL

As the NCK-SPL (SAFE.SPF) has been modified to control the $A_OUTSI system variables, we must now modify the PLC-SPL (FC91 "fcSI:PLC-SPL") and 'UDT19' so as to add the names of the 'Internal Outputs'.

Firstly, update UDT18 within the S7 project. To save time, compile the 'A102_UDT18_002' file within the 'Source' folder.

SAFE.SPF:

```
DEFINE OE2_Spare          AS $A_OUTSE[2]
DEFINE OE3_Spare          AS $A_OUTSE[3]

;=======================================
;Internal Outputs - $A_OUTSI
DEFINE OI1_SBH_AX_DES     AS $A_OUTSI[1]
DEFINE OI2_SG_BIT0        AS $A_OUTSI[2]
DEFINE OI3_SG_BIT1        AS $A_OUTSI[3]
DEFINE OI4_STOPA_DES_AX   AS $A_OUTSI[4]
DEFINE OI5_STOPC_DES_AX   AS $A_OUTSI[5]
DEFINE OI6_STOPD_DES_AX   AS $A_OUTSI[6]
DEFINE OI7_SBH_SP_DES     AS $A_OUTSI[7]
DEFINE OI8_STOPA_DES_SP   AS $A_OUTSI[8]
DEFINE OI9_STOPC_DES_SP   AS $A_OUTSI[9]
DEFINE OI10_STOPD_DES_SP  AS $A_OUTSI[10]

;=======================================
;Markers - $A_MARKERSI
DEFINE MI1_SIREL1_IN1     AS $A_MARKERSI[1]
```

UDT18:

A_INSI64	BOOL	FALSE	INSIP64
OI1_SBH_AX_DES	BOOL	FALSE	OUTSIP1
OI2_SG_BIT0	BOOL	FALSE	OUTSIP2
OI3_SG_BIT1	BOOL	FALSE	OUTSIP3
OI4_STOPA_DES_AX	BOOL	FALSE	OUTSIP4
OI5_STOPC_DES_AX	BOOL	FALSE	OUTSIP5
OI6_STOPD_DES_AX	BOOL	FALSE	OUTSIP6
OI7_SBH_SP_DES	BOOL	FALSE	OUTSIP7
OI8_STOPA_DES_SP	BOOL	FALSE	OUTSIP8
OI9_STOPC_DES_SP	BOOL	FALSE	OUTSIP9
OI10_STOPD_DES_SP	BOOL	FALSE	OUTSIP10
A_OUTSI11	BOOL	FALSE	OUTSIP11

Notes :

Now modify the PLC-SPL (FC91 “fcSI:PLC-SPL”) and download it to the PLC.
(Once again, to save time and syntax errors, compile the source file ‘**A102_FC91_002**’)

```
;Net 7:Assign $A_MARKERSI[6] to KA1_KA2 Contactor Output
IDS=206 DO OE1_KA1_KA2 = MI6_SIREL1_OUT0

;Net 8:Assign SBH for all axes
IDS=207 DO OI1_SBH_AX_DES = FALSE

;Net 9:Assign SBH for Spindle
IDS=208 DO OI7_SBH_SP_DES = FALSE

;Net 10: Assign SG bit0 for all axes & Spindle
IDS=209 DO OI2_SG_BIT0 = FALSE

;Net 11: Assign SG bit1 for all axes & Spindle
IDS=210 DO OI3_SG_BIT1 = FALSE

;Net 12: Assign all axes STOP A deselection to SIRELAY OUT1(1.0s delay)
IDS=211 DO OI4_STOPA_DES_AX = MI7_SIREL1_OUT1

;Net 13: Assign all axes STOP C deselection to SIRELAY OUT0(No delay)
IDS=212 DO OI5_STOPC_DES_AX = MI6_SIREL1_OUT0

;Net 14: Statically deselect all axes STOP D
IDS=213 DO OI6_STOPD_DES_AX = TRUE

;Net 15: Assign Spindle STOP A deselection to SIRELAY OUT2(2.0s delay)
IDS=214 DO OI8_STOPA_DES_SP = MI8_SIREL1_OUT2

;Net 16: Assign Spindle STOP C deselection to SIRELAY OUT0(No delay)
IDS=215 DO OI9_STOPC_DES_SP = MI6_SIREL1_OUT0

;Net 17: Statically deselect Spindle STOP D
IDS=216 DO OI10_STOPD_DES_SP = TRUE

M17
```

```
Network 8 : IDS=207: Assign SBH for all axes
      A     "mFALSE"
      =     "SPL".SPL_DATA.OI1_SBH_AX_DES

Network 9 : IDS=208: Assign SBH for Spindle
      A     "mFALSE"
      =     "SPL".SPL_DATA.OI7_SBH_SP_DES

Network 10 : IDS=209: Assign SG bit0 for all axes & Spindle
      A     "mFALSE"
      =     "SPL".SPL_DATA.OI2_SG_BIT0

Network 11 : IDS=210: Assign SG bit1 for all axes & Spindle
      A     "mFALSE"
      =     "SPL".SPL_DATA.OI3_SG_BIT1

Network 12 : IDS=211: Assign all axes STOP A DES to SIRELAY OUT1(1.0s delay)
      A     "SPL".SPL_DATA.MI7_SIREL1_OUT1
      =     "SPL".SPL_DATA.OI4_STOPA_DES_AX

Network 13 : IDS=212: Assign all axes STOP C DES to SIRELAY OUT0(No delay)
      A     "SPL".SPL_DATA.MI6_SIREL1_OUT0
      =     "SPL".SPL_DATA.OI5_STOPC_DES_AX

Network 14 : IDS=213: Statically deselect all axes STOP D
      A     "mTRUE"
      =     "SPL".SPL_DATA.OI6_STOPD_DES_AX

Network 15 : IDS=214: Assign Spindle STOP A DES to SIRELAY OUT2(2.0s delay)
      A     "SPL".SPL_DATA.MI8_SIREL1_OUT2
      =     "SPL".SPL_DATA.OI8_STOPA_DES_SP

Network 16 : IDS=215: Assign Spindle STOP C DES to SIRELAY OUT0(No delay)
      A     "SPL".SPL_DATA.MI6_SIREL1_OUT0
      =     "SPL".SPL_DATA.OI9_STOPC_DES_SP

Network 17 : IDS=216: Statically deselect Spindle STOP D
      A     "mTRUE"
      =     "SPL".SPL_DATA.OI10_STOPD_DES_SP
```

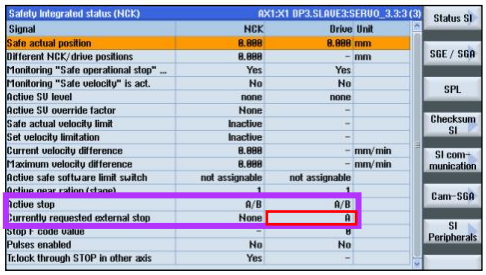
The absolute address range for the internal outputs ‘$A_OUTSI’ and safe markers ‘$A_MARKERSI’ within DB18 in the PLC listed below. These will be useful when making the above changes if the source file is not used.

$A_OUTSI[1]	=	DB18.DBX62.0
$A_OUTSI[2]	=	DB18.DBX62.1
$A_OUTSI[3]	=	DB18.DBX62.2
$A_OUTSI[4]	=	DB18.DBX62.3
$A_OUTSI[5]	=	DB18.DBX62.4
$A_OUTSI[6]	=	DB18.DBX62.5
$A_OUTSI[7]	=	DB18.DBX62.6
$A_OUTSI[8]	=	DB18.DBX62.7
$A_OUTSI[9]	=	DB18.DBX63.0
$A_OUTSI[10]	=	DB18.DBX63.1
$A_MARKERSI[1]	=	DB18.DBX70.0
$A_MARKERSI[2]	=	DB18.DBX70.1
$A_MARKERSI[3]	=	DB18.DBX70.2
$A_MARKERSI[4]	=	DB18.DBX70.3
$A_MARKERSI[5]	=	DB18.DBX70.4
$A_MARKERSI[6]	=	DB18.DBX70.5
$A_MARKERSI[7]	=	DB18.DBX70.6
$A_MARKERSI[8]	=	DB18.DBX70.7

After the PLC-SPL (FC91 “fcSI:PLC-SPL”) has been modified and downloaded to the PLC. You will notice that the error ‘27090 Error in data cross check NCL-PLC, $A_OUTSI[x], NCK:1;’ can now be cleared with ‘Reset’.

This is because there is no longer any difference between the NCK & PLC-SPL. The status of the ‘Safety Integrated System Related Variables’ are also the same.

However, you can see that the ‘Drive’ column within the safety diagnostics screen still shows a request for STOP A.



We have not yet informed the ‘Drive’ (via the axis DB) about the status of the requested external Stops A, C & D.

For every ‘$MA_SAFE_....._INPUT’ and ‘$MA_SAFE_....._OUTPUT’ machine data, there exists an equivalent interface bit within the axis DB’s. The next page details this relationship.

Notes :

Axis Machine Data Signals From PLC	PLC Interface - DB<Axis> PLC to NCK
36970 $MA_SAFE_SVSS_DISABLE_INPUT	DBX22.0
36971 $MA_SAFE_SS_DISABLE_INPUT	DBX22.1
36972 $MA_SAFE_VELO_SELECT_INPUT[0]	DBX22.3
36972 $MA_SAFE_VELO_SELECT_INPUT[1]	DBX22.4
36973 $MA_SAFE_POS_SELECT_INPUT	DBX23.4
36974 $MA_SAFE_GEAR_SELECT_INPUT[0]	DBX23.0
36974 $MA_SAFE_GEAR_SELECT_INPUT[1]	DBX23.1
36974 $MA_SAFE_GEAR_SELECT_INPUT[2]	DBX23.2
36977 $MA_SAFE_EXT_STOP_INPUT[0]	DBX32.2
36977 $MA_SAFE_EXT_STOP_INPUT[1]	DBX32.3
36977 $MA_SAFE_EXT_STOP_INPUT[2]	DBX32.4
36977 $MA_SAFE_EXT_STOP_INPUT[3]	DBX32.5
36978 $MA_SAFE_OVR_INPUT[0]	DBX33.4
36978 $MA_SAFE_OVR_INPUT[1]	DBX33.5
36978 $MA_SAFE_OVR_INPUT[2]	DBX33.6
36978 $MA_SAFE_OVR_INPUT[3]	DBX33.7

Axis Machine Data Signals To PLC	PLC Interface - DB<Axis> NCK to PLC
36980 $MA_SAFE_SVSS_STATUS_OUTPUT	DBX108.0
36981 $MA_SAFE_SS_STATUS_OUTPUT	DBX108.1
36982 $MA_SAFE_VELO_STATUS_OUTPUT[0]	DBX110.3
36982 $MA_SAFE_VELO_STATUS_OUTPUT[1]	DBX110.4
36985 $MA_SAFE_VELO_X_STATUS_OUTPUT	DBX110.5
36987 $MA_SAFE_REFP_STATUS_OUTPUT	DBX108.7
36988 $MA_SAFE_CAM_PLUS_OUTPUT[0]	DBX109.0
36988 $MA_SAFE_CAM_PLUS_OUTPUT[1]	DBX109.2
36988 $MA_SAFE_CAM_PLUS_OUTPUT[2]	DBX109.4
36988 $MA_SAFE_CAM_PLUS_OUTPUT[3]	DBX109.6
N36989 $MA_SAFE_CAM_MINUS_OUTPUT[0]	DBX109.1
N36989 $MA_SAFE_CAM_MINUS_OUTPUT[1]	DBX109.3
N36989 $MA_SAFE_CAM_MINUS_OUTPUT[2]	DBX109.5
N36989 $MA_SAFE_CAM_MINUS_OUTPUT[3]	DBX109.7
N36990 $MA_SAFE_ACT_STOP_OUTPUT[0]	DBX111.4
N36990 $MA_SAFE_ACT_STOP_OUTPUT[1]	DBX111.5
N36990 $MA_SAFE_ACT_STOP_OUTPUT[2]	DBX111.6
N36990 $MA_SAFE_ACT_STOP_OUTPUT[3]	DBX111.7

Notes :

Safety Related Drive Interface (Axis DB)

In order to ensure that the status of the PLC (Drive) safety interface matches the signal level status of the $MA_SAFE_....._INPUT' and '$MA_SAFE_....._OUTPUT' interface of the NC, the PLC program must again be edited. However, in order to maintain the similarity between the NCK-SPL (SAFE.SPF) and the PLC-SPL (FC91 "fcSI:PLC-SPL"), the handling of the PLC safety interface will be done within the SI support block FC90 "fcSI". You will recall that this block supports and calls the PLC-SPL (FC91 "fcSI:PLC-SPL") block.

FC90 "fcSI"

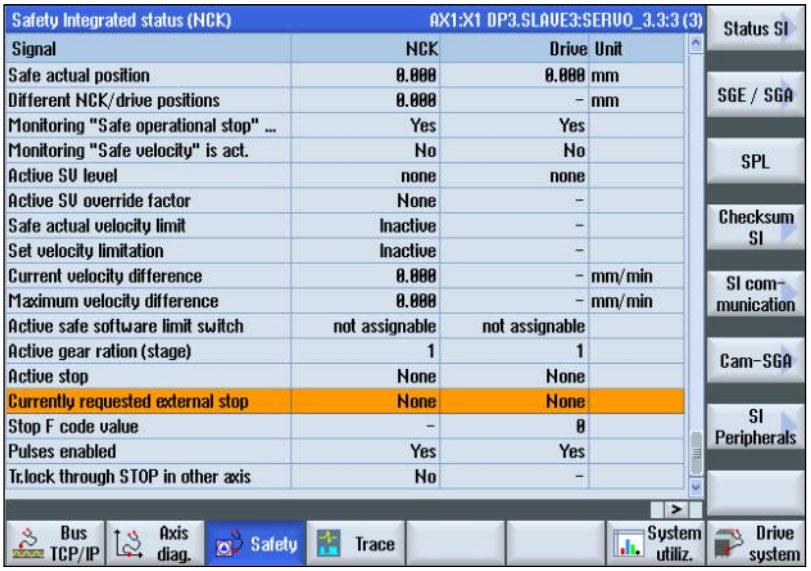
```
Network 7 : Deselection of safe standstill - All axes
      A     "SPL".SPL_DATA.OI1_SBH_AX_DES
      =     "Axis_X1".A_SBH
      =     "Axis_Y1".A_SBH
      =     "Axis_A1".A_SBH
      =     "Axis_C1".A_SBH

Network 8 : Deselection of safe standstill - Spindle
      A     "SPL".SPL_DATA.OI7_SBH_SP_DES
      =     "Spindle_SP1".A_SBH

Network 9 : Selection SG bit 0 - All Axes & Spindle
      A     "SPL".SPL_DATA.OI2_SG_BIT0
      =     "Axis_X1".A_SGselect0
      =     "Axis_Y1".A_SGselect0
      =     "Axis_A1".A_SGselect0
      =     "Axis_C1".A_SGselect0
      =     "Spindle_SP1".A_SGselect0

Network 10 : Selection SG bit 1 - All Axes & Spindle
      A     "SPL".SPL_DATA.OI3_SG_BIT1
      =     "Axis_X1".A_SGselect1
      =     "Axis_Y1".A_SGselect1
      =     "Axis_A1".A_SGselect1
      =     "Axis_C1".A_SGselect1
      =     "Spindle_SP1".A_SGselect1

Network 11 : Deselection external Stop A - Axes
      A     "SPL".SPL_DATA.OI4_STOPA_DES_AX
      =     "Axis_X1".A_extStopA
      =     "Axis_Y1".A_extStopA
      =     "Axis_A1".A_extStopA
      =     "Axis_C1".A_extStopA

Network 12 : Deselection external Stop C - Axes
      A     "SPL".SPL_DATA.OI5_STOPC_DES_AX
      =     "Axis_X1".A_extStopC
      =     "Axis_Y1".A_extStopC
      =     "Axis_A1".A_extStopC
      =     "Axis_C1".A_extStopC

Network 13 : Deselection external Stop D - Axes
      A     "SPL".SPL_DATA.OI6_STOPD_DES_AX
      =     "Axis_X1".A_extStopD
      =     "Axis_Y1".A_extStopD
      =     "Axis_A1".A_extStopD
      =     "Axis_C1".A_extStopD

Network 14 : Deselection external Stop A - Spindle
      A     "SPL".SPL_DATA.OI8_STOPA_DES_SP
      =     "Spindle_SP1".A_extStopA

Network 15 : Deselection external Stop C - Spindle
      A     "SPL".SPL_DATA.OI9_STOPC_DES_SP
      =     "Spindle_SP1".A_extStopC

Network 16 : Deselection external Stop D - Spindle
      A     "SPL".SPL_DATA.OI10_STOPD_DES_SP
      =     "Spindle_SP1".A_extStopD
```

Using the status of the individual internal output ($A_OUTSI) that have been assigned to each of the used $MA_SAFE_....._INPUT machine data, the corresponding interface bit in each axis DB<31..61> can be controlled.

FC90 "fcSI" must be edited to include the networks shown here. Once again, to save time, the source file 'A102_FC90_001' can be compiled and downloaded.

Once the modified FC90 "fc90" has been loaded into the control and the emergency stop has been reset, select the '**Diagnostics**' >> '**Safety**' area.

You can now see that there are no longer any requested external STOPs, on the Drive side. The signal level of the NCK & Drive side are now the same.

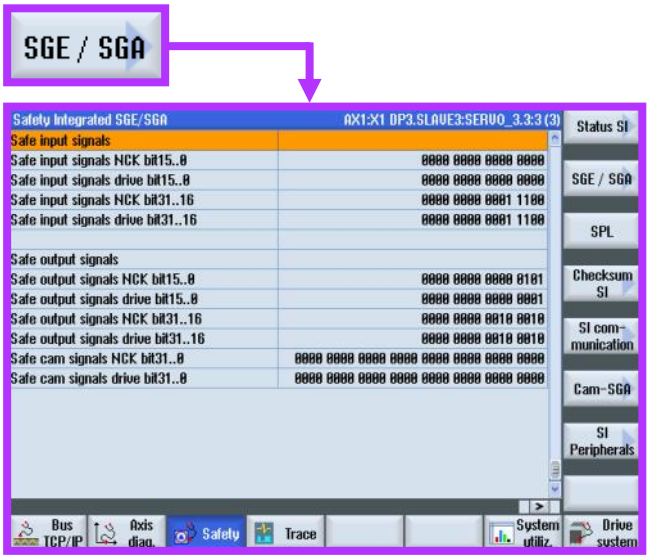


Observe all axes and the Spindle whilst creating an emergency stop, you should see the requested STOPs will be the same for both NCK-SPL and PLC-SPL.

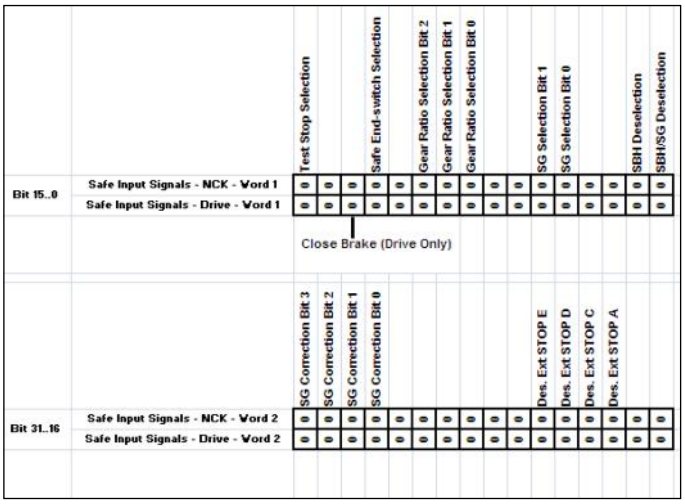
Use the Status SI soft-key to select further axes.

Notes :

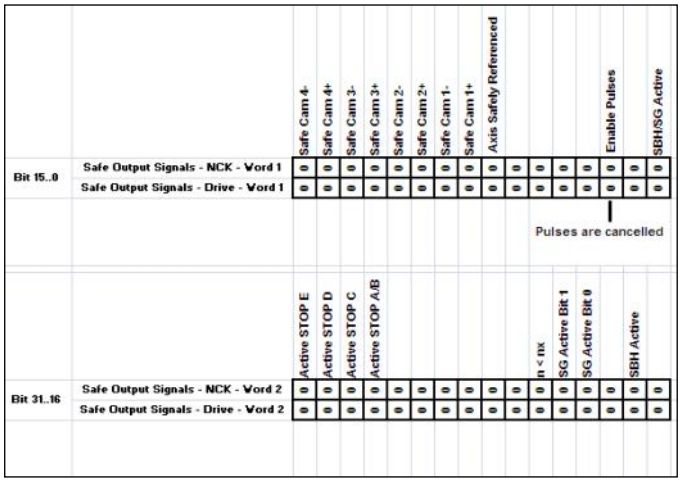
A more detailed overview of the status of the safety interface for each axis/spindle can be seen in the ‘SGE/SGA’ screen.



Description of the ‘Safe Input Signals’ SGE/SGA status display;



Description of the ‘Safe Output Signals’ SGE/SGA status display;



Notes :

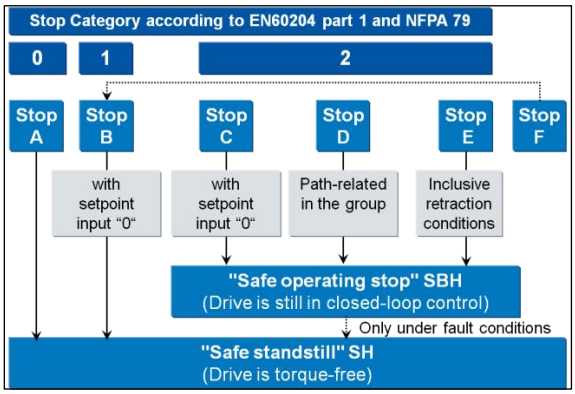
SAFE_DES_VELO_LIMIT and SBH

With the emergency stop reset, no errors displayed and the drives enabled, jogging an axis will result in the following errors;

27024 Axis X1 stop A triggered
27023 Axis X1 stop B triggered
27010 Axis X1 tolerance for safe standstill exceeded

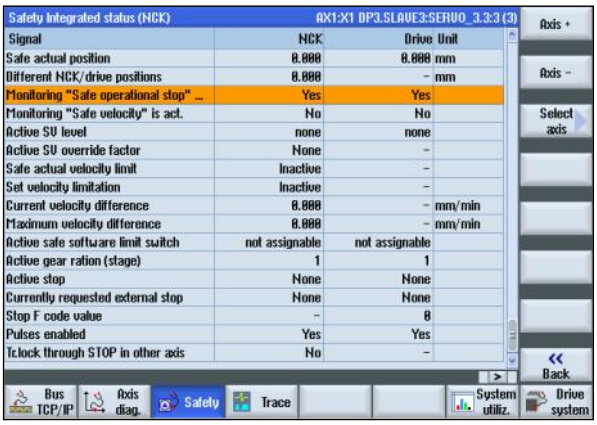
Error 27010 is displayed as the axis has exceeded the standstill tolerance set in $MA_SAFE_STANDSTILL_TOL.

The other errors (27023 & 27024) are in response to the error 27010 and inform you that the axis has been safely stopped.



These errors can only be cleared with Power off-on or an NCK Reset.

Once cleared, reset the emergency stop and observe the 'Status SI' page with the 'Diagnostics' >> 'Safety' area.



You can see above that the axis is actually being monitored for 'Safe Operational Stop' (SBH). So any movement, controlled or uncontrolled, will result in the error 27010.

Whilst we are in a 'Safe Operational Stop' (SBH) state, we could take measures in the PLC to create a feed-hold on the axis. This would stop the axis from moving and the error 27010 would no longer appear.

Alternatively, we can use the machine data 36933 $MA_SAFE_DES_VELO_LIMIT[0..3]. This machine data allows you to specify a percentage limitation on the speed set-point during SBH (Safe Operational Stop) and SG (Safe Reduced Velocity).

Value 100%: The set-point is limited to the active SG level.

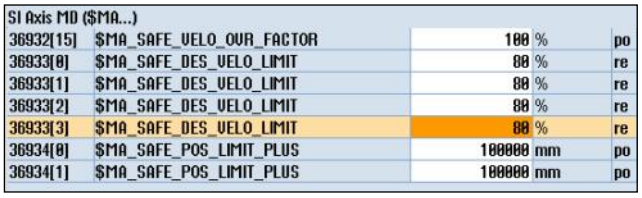
Value >0%: The set-point is limited to a % of the active SG level but the set-point is limited to 0 if SBH is active.

Value = 0%: Set-point limitation is inactive.

Note: For the purposes of this course, set all indexes of MD 36933 SAFE_DES_VELO_LIMIT[0..3] = 80%.

The SAFE machine data is not included in the Cross check monitoring with the drive, nor is it included within the axis checksum (MD 36998 SAFE_ACT_CHECKSUM) as it is a 'Single' channel function. i.e. the set-point is limited in the NC only!

Once all axes and spindle values for MD 36933 have been set to 80% as shown here...



...the axis will no longer attempt to move when jogged...etc.

In order to remove the SBH condition from the axes & spindle, the SPL must be edited.

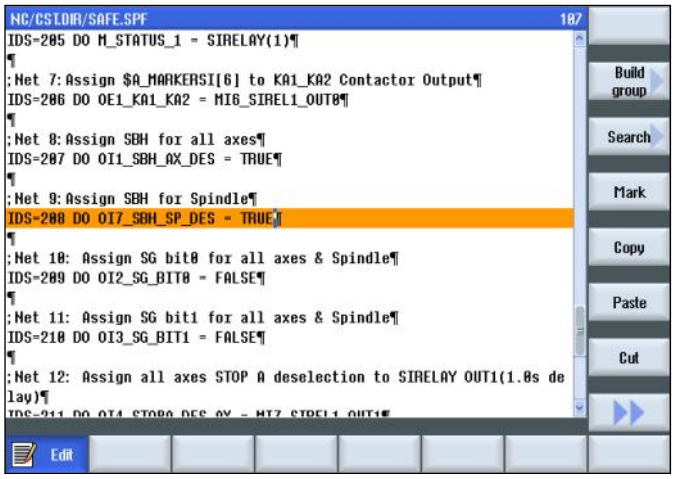
We know from the '$MA_SAFE_....._INPUT' machine data for the 'Safe Standstill' function (36971) which $A_OUTSI variables are assigned to control the SBH function;

Axes;
36971 SAFE_SS_DISABLE_INPUT = 04010101 = $A_OUTSI[1]
Spindle;
36971 SAFE_SS_DISABLE_INPUT = 04010107 = $A_OUTSI[7]

We also know that $A_OUTSI[1] is defined as 'OI1_SBH_AX_DES' and $A_OUTSI[7] as 'OI7_SBH_SP_DES'.

Notes :

We can see in the SPL that IDS 207 & IDS 208 control 'OI1_SBH_AX_DES' ($A_OUTSI[7]) and 'OI7_SBH_SP_DES' ($A_OUTSI[7]).



For now, we can change both of these from 'FALSE' to 'TRUE' As shown above.

The PLC-SPL must also be modified to reflect the same changes.

Edit FC91 "fcSI:PLC-SPL" in Networks 8 & 9 as shown here.

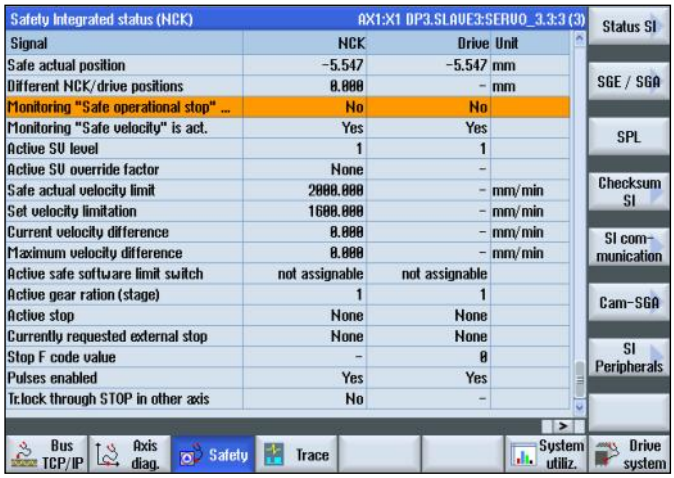
```
Network 8 : IDS=207: Assign SBH for all axes

      A      "mTRUE"
      =      "SPL".SPL_DATA.OI1_SBH_AX_DES


Network 9 : IDS=208: Assign SBH for Spindle

      A      "mTRUE"
      =      "SPL".SPL_DATA.OI7_SBH_SP_DES
```

The absolute address of 'mTRUE" is M1.1.

Save and load FC91 to the PLC and perform an NCK-Reset.

(The NCK Reset is required so that the NCK-SPL, 'SAFE.SPF', is executed and the changes activated.)

After the NCK reset and with the emergency stop reset, the 'Safe Operational Stop' (SBH) should be **De**selected (no longer active).

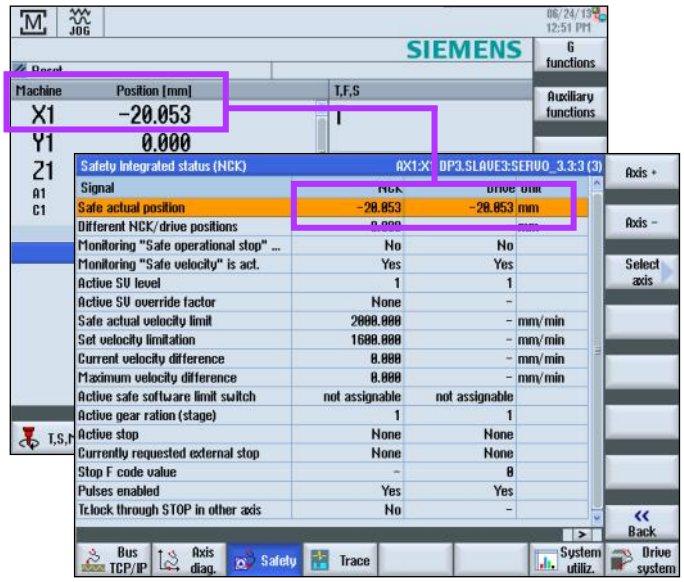


The axes should now jog without error.

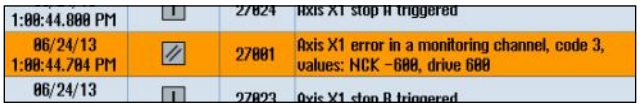
Notes :

Check Polarity of Safe Actual Value

Now that the axes can Jog, the SAFE actual position values must be checked to ensure they have the same polarity as each other. They must also be the same as the MCS axis position within the NC.



If the polarity is not correct, the error 27001 Code 3 will be displayed.



Using the Safety Diagnostics screen, you can determine which safe position polarity is incorrect when compared to the MCS actual position.

If the safe position of the NCK is incorrect, MD 36925 SAFE_ENC_POLARITY, must be adjusted.

If the safe position of the Drive is incorrect, p9516 bit 1 must be changed.

Once these safe MD's have been changed, a 'Confirm SI Data' must be executed followed by an NCK reset.

Once again, check the safe actual positions for correct polarity.

Notes :

Application of the Function Table

The function table that describes how the safety related devices interact with the safety functions must now be applied.

At the moment, the 'Internal Outputs' ($A_OUTSI) assigned to the SBH de-selection and SG selection are statically assigned the values TRUE and FALSE. See Networks 8 to 11 in the PLC-SPL(FC91 "fcSI:PLC-SPL") and IDS=207 to 210 in the NCK-SPL (SAFE.SPF).

Operating mode	*Protective doors*	*Axis / spindle*	*Monitoring*
Production	Closed	X1, Y1, A1, C1	SG4 (SV4)
		SP1	SG4 (SV4)
	Released	X1, Y1, A1, C1	SBH (SOS)
		SP1	SBH (SOS)
Setup	Closed	X1, Y1, A1, C1	SG2 (SV2)
		SP1	SG2 (SV2)
	Released	X1, Y1, A1, C1	SG1 (SV1)
		SP1	SBH (SOS)/
			SG1 (SV1) With HT2 Enable

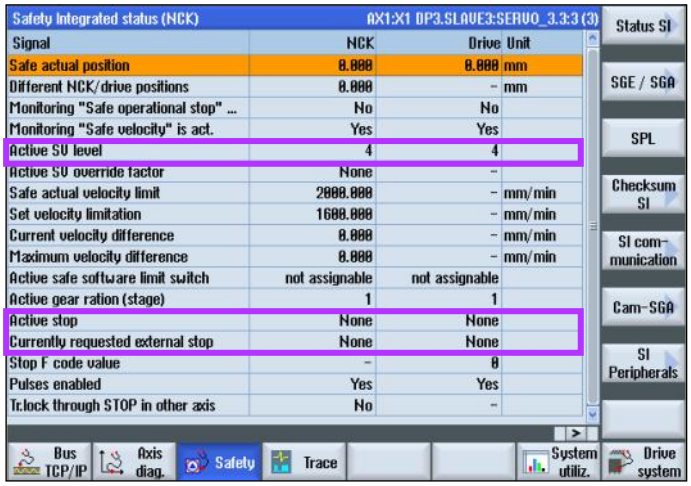
Edit the NCK-SPL (SAFE.SPF) as shown here;

```
;Net 8:Assign SBH for all axes
IDS=207 DO OI1_SBH_AX_DES = (IE3_OP_Door OR IE10_Setup)

;Net 9:Assign SBH for Spindle
IDS=208 DO OI7_SBH_SP_DES = (IE3_OP_Door OR (IE10_Setup AND IE4_HT2_Enable))

;Net 10: Assign SG bit0 for all axes & Spindle
IDS=209 DO OI2_SG_BIT0 = IE3_OP_Door

;Net 11: Assign SG bit1 for all axes & Spindle
IDS=210 DO OI3_SG_BIT1 = (IE3_OP_Door AND (NOT IE10_Setup))
```

Now edit the PLC-SPL(FC91 "fcSI:PLC-SPL") to match the new functionality;

```
Network 8 : IDS=207: Assign SBH for all axes
      0     "SPL".SPL_DATA.IE3_OP_Door
      0     "SPL".SPL_DATA.IE10_Setup
      =     "SPL".SPL_DATA.OI1_SBH_AX_DES

Network 9 : IDS=208: Assign SBH for Spindle
      0     "SPL".SPL_DATA.IE3_OP_Door
      0
      A     "SPL".SPL_DATA.IE10_Setup
      A     "SPL".SPL_DATA.IE4_HT2_Enable
      =     "SPL".SPL_DATA.OI7_SBH_SP_DES

Network 10 : IDS=209: Assign SG bit0 for all axes & Spindle
      A     "SPL".SPL_DATA.IE3_OP_Door
      =     "SPL".SPL_DATA.OI2_SG_BIT0

Network 11 : IDS=210: Assign SG bit1 for all axes & Spindle
      A     "SPL".SPL_DATA.IE3_OP_Door
      AN    "SPL".SPL_DATA.IE10_Setup
      =     "SPL".SPL_DATA.OI3_SG_BIT1
```

Check that the expected functionality is working correctly. This can be done using the 'Diagnostics' >> 'Safety' >> 'Status SI' screen;



Notes :

Setting the Safe Velocity Limits

Each axis/spindle has 4 different 'Safe Velocity Limits' that can be selected.

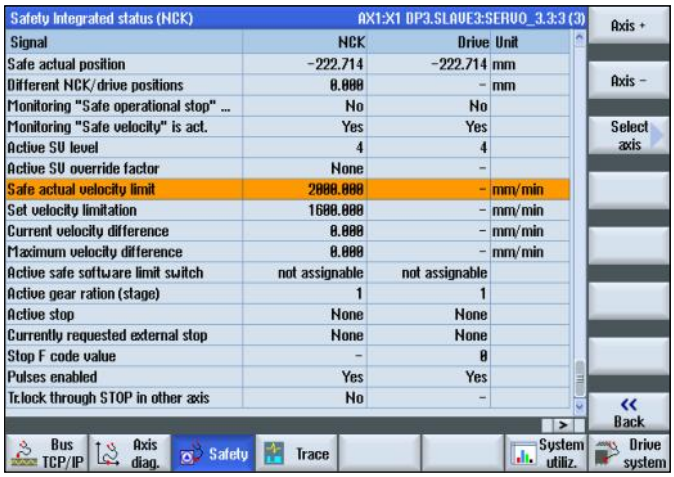
We can see from the function table that we are using SG1, SG2 and SG4 for the axes and the spindle.

Each velocity level is set using the safe machine data '36931[0..3] $MA_SAFE_VELO_LIMIT'. The default value for this machine data is 2000mm/min (5.555rpm for rotary axes).

36931 $MA_SAFE_VELO_LIMIT[0] = SG1
36931 $MA_SAFE_VELO_LIMIT[1] = SG2
36931 $MA_SAFE_VELO_LIMIT[2] = SG3
36931 $MA_SAFE_VELO_LIMIT[3] = SG4

The technical documentation prepared by the machine manufacturer (Function Table, Description of Safety Related Devices….etc) should state the different velocity limitation for each level.

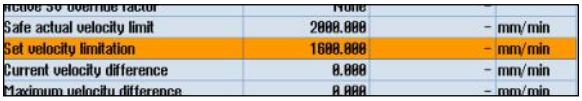
The 'Safe Actual Velocity Limit' can be seen using the 'Diagnostics' >> 'Safety' >> 'Status SI' screen;



The 'Safe Actual Velocity Limit' is the value above which, the error '**27011 Axis X1 safe velocity exceeded**' will be displayed.

This error will also cause an internal '**STOP C**' with error '**27022 Axis X1 stop C triggered**'.

Notice that the axes have a 'Set velocity limit' of 1600mm/min.



This due to MD 36933 SAFE_DES_VELO_LIMIT[0] = 80% that we set earlier. i.e. 2000 * 80% = 1600.

This machine data creates a 'Single Channel - Non-Safe' Set-point limitation so as to prevent the actual velocity exceeding the limit set in 36931 SAFE_VELO_LIMIT[0..3]

The active index for MD 36933 SAFE_DES_VELO_LIMIT[0..3] is selected using the PLC interface only (No NCK selection required as this is a single channel function). It can be selected using DB<Axis>.DBX34.0 & DBX34.1;

DBX34.1	DBX34.0	Active Index of 35933
0	0	0
0	1	1
1	0	2
1	1	3

Lets assume that the OEM has stated that the SG levels require setting as follows;

Linear Axes;
SG1 = 2000mm/min
SG2 = 5000mm/min
SG3 = 10000mm/min
SG4 = 10000mm/min

Rotary Axes;
SG1 = 10rpm
SG2 = 50rpm
SG3 = 100rpm
SG4 = 100rpm

Spindle;
SG1 = 50rpm
SG2 = 500rpm
SG3 = 1000rpm
SG4 = 1000rpm

We must set 36931[0..3] as follows;
SGx = SG / 0.8
2500 = 2000 / 0.8

Linear Axes;
36931[0] = 2500
36931[1] = 6250
36931[2] = 12500
36931[3] = 12500

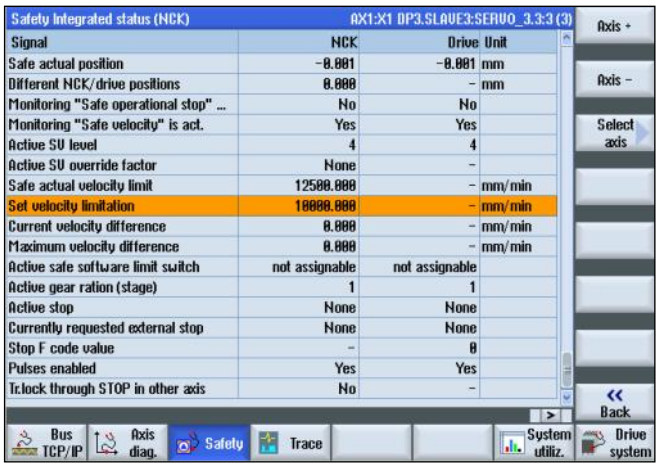
Rotary Axes;
36931[0] = 12.5
36931[1] = 62.5
36931[2] = 125
36931[3] = 125

Spindle;
36931[0] = 62.5
36931[1] = 625
36931[2] = 1250
36931[3] = 1250

Notes :

When the new values for 36931 SAFE_VELO_LIMIT[0..3] have all been entered, **'Copy SI Data' & 'Confirm SI Data'** must now be executed followed by an **'NCK Reset'** to activate the new values.

Using the 'Diagnostics' >> 'Safety' screens, you will now see that the 'Set Velocity Limitation' is 80% of the 'Safe Actual Velocity Limit'.

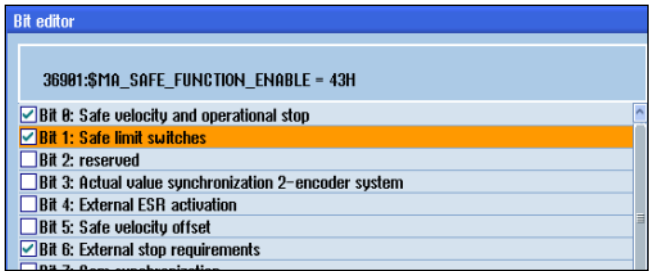


Check this is correct for all axes and spindle for all SG levels.

Safe End Limits and Safe Software Cams

It may be required to use 'Safe End Limits' on one or more axes. Up to two 'Safe End Limit Pairs' are available per axis.

To enable the 'Safe End Limits', in this case for the X1 Axis, MD 36901 SAFE_FUNCTION_ENABLE bit 1 must be set.



The Machine data's….

36934[0] SAFE_POS_LIMIT_PLUS
36935[0] SAFE_POS_LIMIT_MINUS

...can also be set with limit values e.g.

36935[0] = (36100 POS_LIMIT_MINUS) - 10mm
36934[0] = (36110 POS_LIMIT_PLUS) + 10mm

Set the X axis Safe Limits to the values shown here.

36934[0] SAFE_POS_LIMIT_PLUS = 110.000mm
36935[0] SAFE_POS_LIMIT_MINUS = -110.000mm

Once the values have been set and the 'Safe Limit Switches' have been enabled with machine data '**36901 SAFE_FUNCTION_ENABLE bit 1'**, **'Copy SI Data' & 'Confirm SI Data'** must now be executed followed by an **'NCK Reset'** to activate the new values.

The alarms...

201797 Axis X1 Servo….SI Motion CU: Axis not safely referenced.
27000 Axis X1 is not safely referenced

...will be displayed.

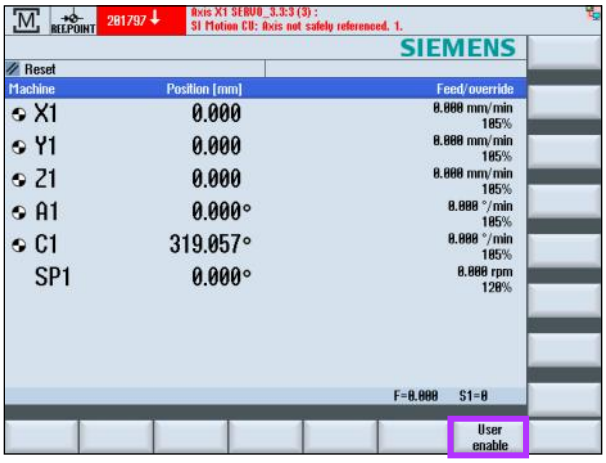
These alarms will remain after the axes have been referenced (To reference all axes, select Jog-Ref mode & press Cycle Start).

Now that a 'Safe Position Related Function' is active (i.e. Safe End Switches or Safe Cams) the axis in question must be 'Safely Referenced' during setup.

Notes :

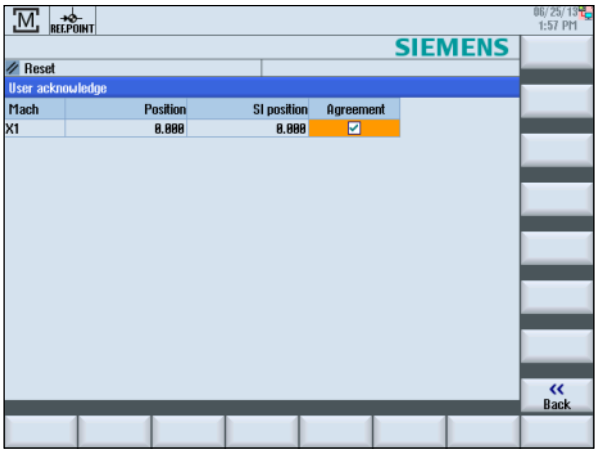
User Agreement

In order to 'Safely Reference' an axis, the 'User Agreement' must be given by the machine user. This is done in the Machine area of Jog-Ref mode.



Key-switch/Password level 3 (Orange Key) or higher must be set in order to make the 'User Enable' soft-key available.

Using the 'Select' key, the user agreement is given to the X1 axis and both alarms will automatically be cleared.



You have now confirmed that the 'SI Position' is the same as the 'NC Position'.

From now on following power-on, the alarms...
201797 Axis X1 Servo….SI Motion CU: Axis not safely referenced.
27000 Axis X1 is not safely referenced

...will be displayed until the axis is again referenced.

The SI position value is stored in SRAM. On power-up of the system, and once the axes have been referenced, the actual NC position is compared to the buffered SI position. If the difference between the two values exceed the value stored in **MD 36944 SAFE_REFP_POS_TOL**, the user agreement will be removed.

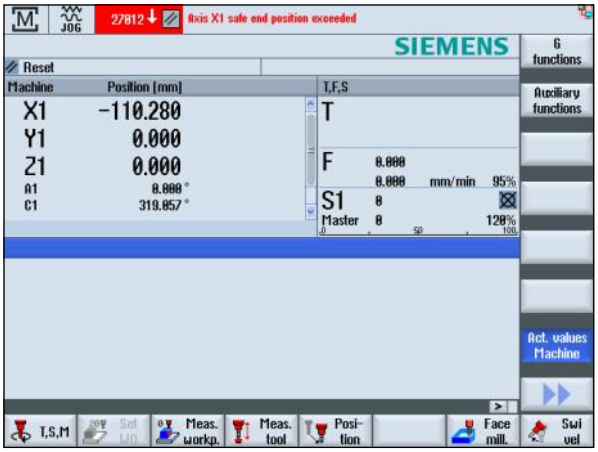
The 'User Agreement' will also be removed if the Safe NCK and Drive actual position values exceed the value of **MD 36942 SAFE_POS_TOL**.

Note: The X1 axis has an incremental encoder. This means that as the reference cam is simulated within the PLC, the axis will never reference in the same position.

The newly referenced actual NCK position will never be the same as the buffered 'SI Position' value.

It will always exceed the default tolerance value of 0.01mm for MD 36944 SAFE_REFP_POS_TOL and the User Agreement will always be removed after referencing.

The 'Safe End Limits' are now active and the axis cannot be moved past the set 'Safe Limit Values' of +/- 110mm.



If the axis is moved past the Safe Limits, the error '**27012 Axis X1 safe end position exceeded**' will be displayed.

The only possibility to recover the axis now is to remove the 'User Agreement' for the X1 axis, reset the fault and Jog the axis clear of the end limit.

Notes :

Removing the 'User Agreement' to recover an axis that has exceeded it's 'Safe End Limit' is one possibility.

It is also possible to select the second 'Safe End Limits' which would be set greater than the first limit, this would then allow recovery from the end limit without the need to remove the 'User Agreement'.

To do this, MD 36973 SAFE_POS_SELECT_INPUT must be assigned to an internal output ($A_OUTSI) and controlled.

- Assign $A_OUTSI[11] to the X1 Axis;
  36973 SAFE_POS_SELECT_INPUT = **0401010B**

- Set the limit values for the second 'Safe End Limit Pair';
  36934[1] SAFE_POS_LIMIT_PLUS = 120.000mm
  36935[1] SAFE_POS_LIMIT_MINUS = -120.000mm

- Now edit the NCK-SPL (SAFE.SPF) to add a new definition for $A_OUTSI[11];
  DEFINE OI11_SE2_AX AS $A_OUTSI[11]

- Add a new synchronous action at the end of the SAFE.SPF;
  ;Net 18: Select Safe End Limit 2
  IDS = 217 DO OI11_SE2_AX = IE9_OT_Override

The PLC-SPL must also now be modified in the same way.

- Edit UDT18 to add the definition for $A_OUTSI[11];

+25.1	OI10_STOPD_DES_SP	BOOL	FALSE	OUTSIP10
+25.2	OI11_SE2_AX	BOOL	FALSE	OUTSIP11
+25.3	A_OUTSI12	BOOL	FALSE	OUTSIP12

- Edit the PLC-SPL (FC91 "fcSI:PLC-SPL") to add Net 18;

```
Network 18: IDS=217: Select Safe End Limit 2
      A     "SPL".SPL_DATA.IE9_OT_Override
      =     "SPL".SPL_DATA.OI11_SE2_AX
```

- Edit the PLC-SPL support block (FC90 "fc:SI") to write the status of $A_OUTSI[11] to the X1 axis safe interface signal for Safe End Switch 2 Selection;
  (MD 36973 = DB<Axis>.DBX23.4)

```
Network 17: Safe End Switch 2 Selection X Axis
      A     "SPL".SPL_DATA.OI11_SE2_AX
      =     "Axis_X1".A_SE
```

- Download the modified FC90 and FC91 to the PLC.

Copy and Confirm of SI Data and NCK Reset are now required to activate the changes in the NC Safe MD and NCK-SPL(SAFE.SPF).

- After the NCK-Reset, reference all axes (Jog-Ref >> Cycle Start).

- Issue the 'User Agreement' for the X1 axis.

- Jog the X1 axis to +/- 110mm so that it activates the first 'Safe End Limit' and the error 27012 is displayed.

- Activate 'End Limit Override' key-switch and reset the errors.

- Jog the X1 axis clear of the end limit and release the key-switch.

Using this method to select the second 'Safe End Limits' can be improved by using the non-safe hardware limit signals for the axes DB<Axis>.DBX12.0 & DBX12.1.

This will prevent the axis from being allowed to travel towards the second Safe Limit.

To do this we will need to determine which end limit has been exceeded, Plus or Minus.

However, no interface signal exists to indicate that the axis has exceeded a 'Safe End Limit'. For this, we can use a 'Safe Cam'.

- Enable SAFE Cam pair 1 for the X1 axis;
  36901 SAFE_FUNCTION_ENABLE bit 8 & 9

- Set Cam Plus & Minus values the same as the 'Safe End Limit 1' values. (36934[0] & 36935[0]).
  36936[0] SAFE_CAM_POS_PLUS = 110.000mm
  36937[0] SAFE_CAM_POS_MINUS = -110.000mm

Notes :

- Edit the PLC-SPL support block (FC90 “fc:SI”) to read the status of SAFE CAM pair 1 and control the hardware limit signals DB<Axis>.DBX12.0 & DBX12.1.

```
Network 18 : X Axis End Limit Minus
      AN    "Axis_X1".E_SN1Plus
      AN    "Axis_X1".E_SN1Minus
      A     "SPL".SPL_STATUS14          //End of SPL program reached
      =     "Axis_X1".A_HWLimitMinus

Network 19 : X Axis End Limit Plus
      A     "Axis_X1".E_SN1Plus
      A     "Axis_X1".E_SN1Minus
      =     "Axis_X1".A_HWLimitPlus
```

(SAFE Cam 1 Plus = DB<Axis>.DBX109.0)
(SAFE Cam 1 Minus = DB<Axis>.DBX109.4)

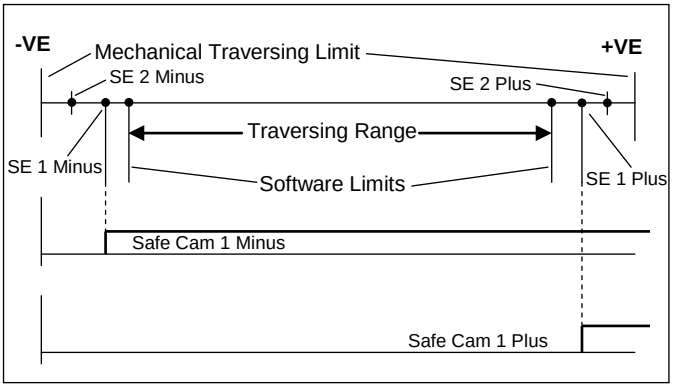
Note: To prevent the Minus hardware limit alarm from appearing during power-up, DB<Axis>.DBX12.0 is interlocked with DB18.DBX137.5 (NCK-SPL Completed).

- Save and Download the modified FC90.
- Copy & Confirm SI Data, then NCK Reset.
- Remove User Agreement for X1 axis (only for this class due to incremental encoder).
- Reference all axes and re-issue the User Agreement for the X1 axis.

As we are only using the Safe Cam status in 1 channel only, no modification of the SPL (NCK or PLC) is required.

When the axis is now jogged onto the first safe limit, it will also be sat on Safe Cam 1 (Plus or Minus). The cam signal is connected to the standard non-safe hardware limit interface of the axis.

The alarm ‘**21614 Axis X/X1 hardware limit switch + / -**’ will also now be displayed along with the alarm ‘**27012 Axis X1 safe end position exceeded’**. This will now prevent the axis from moving towards the second ‘Safe End Limit’ during recovery.



As a result of system--related actual value differences, the cam signals of the monitoring channels can have different states.

In order to prevent this, the cam synchronization can be activated. This rounds off the results of both channels. The cam SGAs at the input position of the SPL are synchronized if the user has parameterized this using the function enable.

Cam signal synchronization is enabled using machine data **MD 36901: $MA_SAFE_FUNCTION_ENABLE, bit 7**

When cam synchronization is activated, cam signals are output with a hysteresis that takes into account the approach direction. This helps to prevent the SGAs from ”flickering” if the axis is positioned exactly on the cam.

The magnitude of the hysteresis is determined by machine data: **MD 36940: $MA_SAFE_CAM_TOL**

Notes :

Reducing velocity when not Safely Referenced

It could be useful to limit the velocity of the axes if they are not 'Safely Referenced'. (i.e. Before referencing or without 'User Agreement').

To do this we must monitor the 'Safely Referenced State' of the axis within the SPL and ensure SG1 is selected if the axis is not safely referenced.

- Assign 'Internal Input 1' ($A_INSI[1]) to the 'Safely Referenced State of the X1 axis;
  MD 36987 SAFE_REFP_STATUS_OUTPUT = 04010101

- Create a definition for $A_INSI[1] in the NCK-SPL (SAFE.SPF);

```
;=======================================
;Internal Inputs - $A_INSI
DEFINE II1_AX_REF          AS $A_INSI[1]

;=======================================
;Internal Outputs - $A_OUTSI
DEFINE OI1 SBH AX DES     AS $A_OUTSI[1]
```

- Also in the NCK-SPL, Interlock the SG selection bits 0 & 1 with **II1_AX_REF**;

```
;Net 10: Assign SG bit0 for all axes & Spindle
IDS=209 DO OI2_SG_BIT0 = (IE3_OP_Door AND II1_AX_REF)

;Net 11: Assign SG bit1 for all axes & Spindle
IDS=210 DO OI3_SG_BIT1 = (IE3_OP_Door AND (NOT IE10_Setup) AND II1_AX_REF)
```

- Within the Step 7 project, Edit UDT18 to include the definition for $A_INSI[1];

+15.7	A_OUTSE64	BOOL	FALSE	OUTSEP64
+16.0	II1_AX_REF	BOOL	FALSE	INSIP1
+16.1	A_INSI2	BOOL	FALSE	INSIP2

- Edit the PLC-SPL support block (FC90 "fcSI") to assign the 'Safely Referenced Status' to $A_INSI[1] (Insert a new Network 4);

```
Network 4 : Assign Safely Referenced Status to $A_INSI[1]
      A     "Axis_X1".E_SRef
      =     "SPL".SPL_DATA.A_INSI1
```

Absolute addresses for the code are;
A DB31.DBX 108.7
= DB18.DBX 54.0

- Edit the PLC-SPL (FC91 "fcSI:PLC-SPL") to Interlock the SG selection bits 0 & 1 with **II1_AX_REF**;

```
Network 10 : IDS=209: Assign SG bit0 for all axes & Spindle
      A     "SPL".SPL_DATA.IE3_OP_Door
      A     "SPL".SPL_DATA.A_INSI1
      =     "SPL".SPL_DATA.OI2_SG_BIT0

Network 11 : IDS=210: Assign SG bit1 for all axes & Spindle
      A     "SPL".SPL_DATA.IE3_OP_Door
      AN    "SPL".SPL_DATA.IE10_Setup
      A     "SPL".SPL_DATA.A_INSI1
      =     "SPL".SPL_DATA.OI3_SG_BIT1
```

- Download FC90 & FC91 to the PLC.
- Copy, Confirm SI Data and NCK Reset.
- Remove User Agreement for X1 axis. (This class only due to incremental encoder).
- Check that SG level 1 is now active, even with operator door closed, Production mode...etc. (Normally SG4);

Monitoring "Safe operational stop" ...	No	No	
Monitoring "Safe velocity" is act.	Yes	Yes	
Active SU level	1	1	
Active SU override factor	None	-	
Safe actual velocity limit	2500.000	-	mm/min
Set velocity limitation	2000.000	-	mm/min

- Reference all axes (Jog-REF & Cycle Start)
- Issue User agreement and check SG Level selection is working as expected. i.e SG4 is now selected with operator door closed and production mode selected.

Monitoring "Safe operational stop" ...	No	No	
Monitoring "Safe velocity" is act.	Yes	Yes	
Active SU level	4	4	
Active SU override factor	None	-	
Safe actual velocity limit	12500.000	-	mm/min
Set velocity limitation	10000.000	-	mm/min

Notes :

Test Stop:

This section describes the following;

- Description of Test Stop
- Safe Brake Test
- Pulse Disable Test
- External Stop Test
- SPL I/O Test
- SPL Protection

NOTE: Any NC/PLC code shown in this document is to be treated as an example and for guidance purposes only!

Notes :

Test Stop (forced Checking Proceedure) Description

A forced checking procedure must be carried out for all static (steady-state) signals and data. Within the required time (8 h), the state must change from a logical 1 to a logical 0 -- or vice versa. If the state remains static in a fault situation, then this is detected at the latest as a result of this forced checking procedure and the subsequent comparison.

A forced checking procedure must be used, e.g. for components that are required to stop a process (e.g. contactors and power semiconductors) - the so-called shutdown path and for the shutdown condition. Generally, it is not possible to test a shutdown condition, e.g. violation of a limit value criterion, using other methods such as e.g. crosswise data comparison, when the machine is in an acceptable (good) condition.

This also applies to errors along the entire shutdown path including associated hardware and software and circuit-breakers.

By integrating a test stop every eight hours with a comparison and expected status, faults can also be detected when the machine is in an acceptable (good) condition.

Note: Acceptable (good) condition means that there are no machine faults that are apparent to the operator

Note: For Safety Integrated, a forced checking procedure interval of one year is permissible.

The forced checking procedure is used to detect faults/errors in the software and hardware of the two monitoring channels. In order to do this, the safety - related parts in both channels must be processed at least once during a defined period in all safety-related branches. Any faults/errors in the monitoring channel would cause deviations and will be detected by the cross--wise data comparison.

For Safety Integrated, the forced checking procedure interval is max. 1 year. This involves components from the SINUMERIK 840D sl / SINAMICS S120 system.

Possible requirements relating to shorter forced checking procedure intervals of safety-related components (e.g. PROFIsafe I/O modules, sensors such as e.g. emergency stop buttons, actuators such as e.g. brakes, etc.) are not influenced.

The forced checking procedure must be initiated by the user or integrated in the process as an automatic procedure,

e.g.

- When the axes are stationary after the system has been powered--up
- When the protective door is opened
- In defined cycles (e.g. every 8 hours) The maximum permissible is every year).
- In the automatic mode, dependent on the time and event.

The forced checking procedure also includes testing the safety -related sensors and actuators at the safety--related inputs/ outputs.

In this case, the entire circuit including the Safe Programmable Logic (SPL) is tested to ensure that it is correctly functioning.

Warning

The test interval duration of max. 1 year may only be extended under the following conditions:

- In the time after the test interval has expired, no hazards for personnel may be allowed to occur -- they must be completely excluded (e.g. the protective door is closed and is also interlocked)
- After the test interval has expired, before a possible hazard to personnel (e.g. for a request to open a protective door), a test stop or a forced checking procedure must be carried out to absolutely ensure the availability of the shutdown paths and the safety-related inputs/outputs.

This means that for the duration of the automatic mode (with the protective door closed and interlocked), a fixed cycle is not strictly specified. After expiry of the time, the forced checking procedure can be carried out before the next opening of the protective door.

The ‘Test Stop’ or ‘Force Checking Procedure’ for Sinumerik safety integrated can include some or all of the following tests;

- Safe Brake Test. **Note: This must be done first as subsequent tests involve testing the shutdown paths of the Sinamics drives (STO). This could lead to unexpected movement of the axis if the brake is faulty.**
- Pulse Disable Test (STO)
- External Stop Test (STOP A,C D...etc)
- SPL Inputs & Outputs (Check-back circuit of contactors...etc)

Notes :

Safe Brake Test

Note:
If used, the safe brake test should be carried out prior to other test stops. Subsequent tests should be inhibited if the brake test fails. This reduces the possibility any unexpected movement from the axis due to a faulty brake when testing the shutdown path of the pulses (STO).

The 'Safe Brake Test' is primarily intended for so-called 'Vertical Axes'.

The functionality is based on "travel to fixed stop" (FXS). The travel to fixed stop can be individually parameterized to test the function of the mechanical braking system. It is activated and deselected from the PLC.

The machine manufacturer can use his PLC user program to close the brake at a suitable instant in time (nominal value every 8h) and to initiate that the drive produces an additional force in addition to the weight of the axis. In an error/fault-free state, the brake can produce the necessary braking torque/ force that the axis hardly moves.

When an error/fault occurs, the actual position value exits the parameterizable monitoring window and the function test of the brake system is negatively acknowledged.

The brake test must always be started when the axis is at a standstill.

The direction in which the drive produces its torque / its force is specified by the PLC using a "traversing motion" via FC 18.

The direction of travel should be selected so that the motor applies force in the direction of the already existing force due to weight as a result of the load.

The target of this motion must be able to be reached without any danger (no collision, sufficient distance to the end stops).

Safe Brake Test Machine Data

The user can use the following axial NCK machine data to parameterize the function test of the mechanical braking system:

MD 37000: $MA_FIXED_STOP_MODE
Enable brake test

MD 37030: $MA_FIXED_STOP_THRESHOLD
Threshold for fixed stop detection

MD 36966 $MA_SAFE_BRAKETEST_TORQUE
Enter the test torque

MD 36967: $MA_SAFE_BRAKETEST_POS_TOL
Position tolerance, brake test

MD 36968: $MA_SAFE_BRAKETEST_CONTROL
Bit 0 = 0: As average value of the torque limiting, drive parameter p1532: "Torque limit offset" is used
Bit 0 = 1: The measured torque at the instant in time that the brake test is selected is used as the average value of the torque limit.

MD 37000 $MA_FIXED_STOP_MODE:
The function test of the mechanical braking system is enabled by setting bit 1 in MD 37000 $MA_FIXED_STOP_MODE. If the user needs to travel to a fixed stop with this axis from the part program, bit 0 can also be set. It is internally monitored to check that only one type of travel to fixed stop is active at a time. In the case of an error, Alarm 20092, "Axis % Travel to fixed stop still active" is issued.

MD 37030 $MA_FIXED_STOP_THRESHOLD:
The contour deviation that is determined is always used in the brake test to detect that the fixed stop has been reached. The required threshold value must be set in MD 37030 $MA_FIXED_STOP_THRESHOLD.

This means that the traversing distance from the PLC via FC 18 must be greater than this threshold value.

If the travel distance that is entered is too short, after the end position is reached on the setpoint side, Alarm 20096 "Axis % 1 brake test aborted, additional information %2" is output. The supplementary info contains the value 2 "End position reached, motion stopped".

Notes :

MD 36966 $MA_SAFE_BRAKETEST_TORQUE:
The machine manufacturer must parameterize the required brake test torque as percentage in the axial MD 36966 $MA_SAFE_BRAKETEST_TORQUE.

The magnitude of the torque to be configured is orientated to the maximum holding force of the brake, according to the data sheet. Internally, this is used to calculate the drive torque that is required in addition to the weight of the axis to load the brake. The drive torque to load the holding brake is limited to the maximum motor torque if the desired test torque would require a higher drive torque.

Value for MD 36966 = ((test torque of the brake / p2003) * 100) * 1.3)

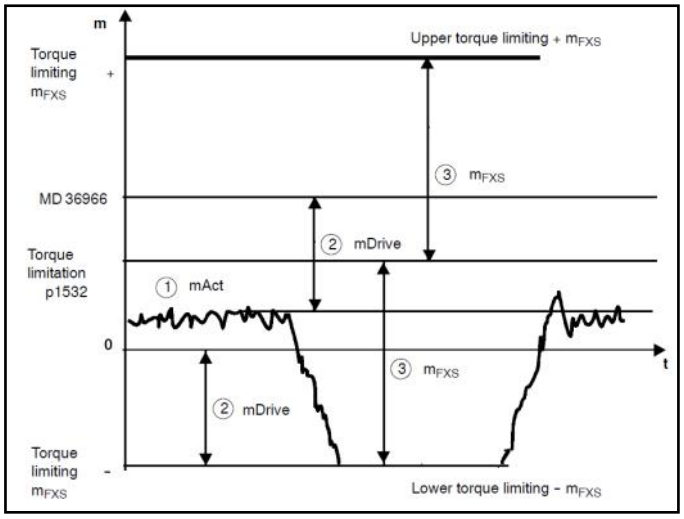
The value from $MA_SAFE_BRAKETEST_TORQUE refers to the reference torque or the reference force from drive parameter p2003, whose image is saved in $MA_SAFE_BRAKETEST_TORQUE_NORM.
The magnitude of the torque to be configured is orientated to the maximum holding force of the brake to be tested. Incorrect parameterization in MD $MA_SAFE_BRAKETEST_TORQUE could mean that the drive with reduced torque cannot even apply the required holding torque. These parameter assignments are detected when the brake test is selected and results in Alarm 20095 ”Axis %1 inadmissible holding torque %2”.

MD 36967 $MA_SAFE_BRAKETEST_POS_TOL:
The monitoring window for the maximum permissible movement during the brake test is defined in the axial MD 36967 $MA_SAFE_BRAKETEST_POS_TOL. The PLC actively monitors this position window - from the start of the brake test and not only when it has been detected that the fixed stop has been reached. This is a difference when compared to activating the traversing to the fixed stop function from the part program.

MD 36968 $MA_SAFE_BRAKETEST_CONTROL:
Principally, the automatic determination of the load torque available using MD 36998 $MA_SAFE_BRAKETEST_CONTROL, bit 0 = 1 is preferred. This is because over the complete traversing path of a suspended axis the torque situation continually changes to some extent or the other. The torque situation is dependent on the different tools/workpieces being used and can vary significantly.

Using the automatic torque determination function, the instantaneous holding torque available at standstill is automatically determined and is temporarily used as average value for the torque limiting in the drive. In this case, it must be ensured that at the start of the test, the brake is open, otherwise an incorrect reference value will be determined. With the automatic torque determination function, the plausibility of the load torque is not checked.

The currently available holding torque is displayed in r1509 ”Force setpoint before force limiting”. If the automatic torque determination function is not used (MD 36998 $MA_SAFE_BRAKETEST_CONTROL=0), then p1532 ”Torque limit offset” should be parameterized. Also in this case, when selecting the brake test, the holding torque required for the force due to the weight is internally measured and the effective brake test torque adapted. Contrary to the automatic torque determination function, the plausibility of the load torque is checked.



1) When selecting the brake test, the holding torque required for the force due to the weight of the axis is internally measured (mAct).

2) The drive must only additionally provide the difference between this torque and the braking torque from MD 36966 $MA_SAFE_BRAKETEST_TORQUE. In the diagram, this torque is designated with mDrive.

3) For the non automatic torque determination function, the following applies: The torque limiting of SINAMICS is symmetrical around the torque from drive parameter p1532. In the diagram, the measured torque m_Act is however less than p1532.
This is the reason that mFXS is entered as torque limiting.
mFXS is the sum from mdrive and the drive parameter p1532.
If the measured torque mact matches that parameterized in drive parameter p1532, mFXS becomes the value from the $MA_SAFE_BRAKETEST_TORQUE.

Notes :

Torque Limits

The torque limits p1520[0] and p1521[0] and the reference torque p2003 are pre-assigned when commissioning the motor

Further, the torque limits p1522[0] and p1523[0] are pre-assigned as follows:

p1522[0] 63:2902:5 reference to +100% referred to p2003
p1523[0] 63:2902:12 reference to -100% referred to p2003

When using the safe brake test, these pre--assignments for p1522[0] / p1523[0] may not be changed.

To ensure that the brake test functions without any errors, it should be checked as to whether the required test torque in MD 36966 $MA_SAFE_BRAKETEST_TORQUE is not prevented from being generated due to the fact that torque limits are effective in the drive.

Further, e.g. the following parameters can also have a limiting effect:
p1530/p1531 (power limit, motoring/regenerating)
p0640 (current limit)
p0326 (motor stall torque correction factor)

Traversing direction for the brake test

The brake test must always be started when the axis is at a standstill.

The direction in which the drive produces its force is specified by the PLC using the direction specified by the traversing motion of FC 18.

For a brake test, the motor should apply a force to the brake that is applied in addition to the force due to weight. The target of this traversing motion must be able to be reached without incurring any potential hazard (sufficient clearance to end stops).

Brake control for SINUMERIK 840D sl
If Safety Integrated is activated for an axis, then the brake can be closed using the interface signal ”Close brake”, DB31-61.DBX23.5. The feedback signal is realized using the interface bit ”Motor holding brake open”, DB31-61.DBX92.5. In this case, it involves a single-channel control.
If the brake is to be safely controlled, then the SBC function integrated in the drive must also be activated. **(p9602 & p9802=1)**
The interface bits for the brake, only activated in conjunction with Safety Integrated, have a higher priority than the standard interface signal DB31-61, DBX20.5 ”Open motor holding brake”. The ”Extended brake control” function of the S120 is used independently of SBT.

Safe Brake Test Sequence

Warning
The brake test must be carried out before carrying out the test stop. If the brake test was not successful (i.e. the brake cannot hold the axis), then it is not permissible that the test stop is carried out. Users must carefully take this into consideration when configuring the brake test and test stop. The brake may only be tested when the axis is in an absolutely safe position.

The brake test must always be started when the axis is at a standstill. For the entire duration of the brake test, the enable signals of the parameterized axis must be set to enable (e.g. the controller inhibit, feed enable signals). It must be ensured that the feed override of 100% is effective.

Within the PLC, FB11(“SI_BrakeTest”) and FC18 “SpinCtrl” are used to perform the actual ‘Safe Brake Test’.

FB11(“SI_BrakeTest”) controls the sequence for the SBT and enables the ’Travel to Fixed Stop Mode’. FB11 also reads/ writes the axis VDI signals (DB31-61) for the axis involved. These signals include the following;

DBX11.0	**Activate Brake Test**
DBX71.0	**Brake Test Active**
DBX23.5	**Close Brake**
DBX92.5	**Brake Closed**
DBX64.6/7	**Traversing Command Active**
DBX62.5	**FXS(Fixed Stop) Reached**

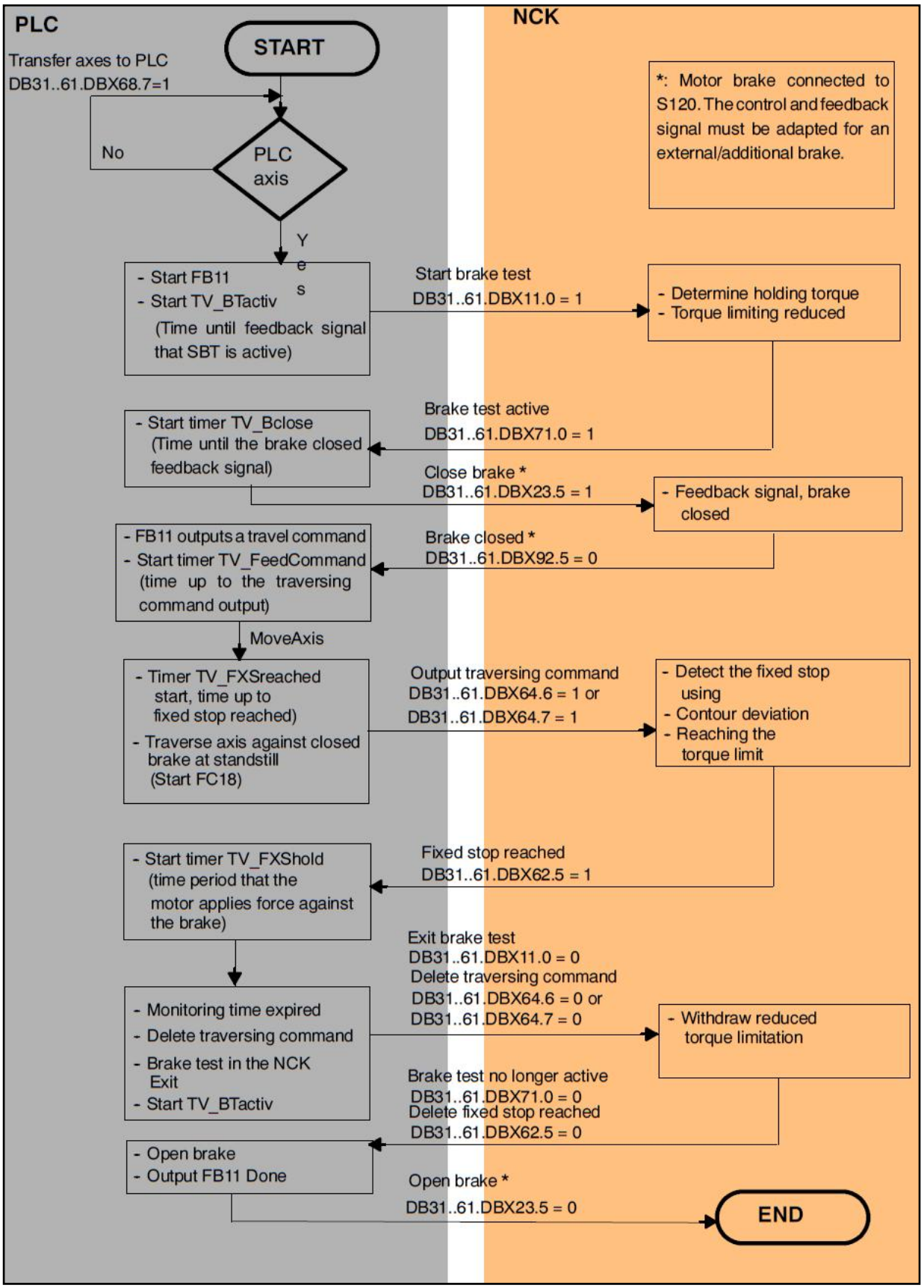
Before the brake test can be started via FB 11 (from the basic program), the NC axis to be tested must be transferred to the PLC as ”PLC-controlled axis”.

During the complete test, the axis must remain a PLC-controlled axis.

Start via FB 11 can be made after the transfer to the PLC.

FB11 provides a “MoveAxis” output that is then used to start FC18 for the ‘Travel to Fixed Stop’ mode.

Notes :


PLC
NCK
START
Transfer axes to PLC
DB31..61.DBX68.7=1
No
PLC axis
Yes
*: Motor brake connected to S120. The control and feedback signal must be adapted for an external/additional brake.
- Start FB11
- Start TV_BTactiv
(Time until feedback signal that SBT is active)
Start brake test
DB31..61.DBX11.0 = 1
- Determine holding torque
- Torque limiting reduced
Brake test active
DB31..61.DBX71.0 = 1
- Start timer TV_Bclose
(Time until the brake closed feedback signal)
Close brake *
DB31..61.DBX23.5 = 1
- Feedback signal, brake closed
Brake closed *
DB31..61.DBX92.5 = 0
- FB11 outputs a travel command
- Start timer TV_FeedCommand
(time up to the traversing command output)
MoveAxis
- Timer TV_FXSreached start, time up to fixed stop reached)
- Traverse axis against closed brake at standstill
(Start FC18)
Output traversing command
DB31..61.DBX64.6 = 1 or
DB31..61.DBX64.7 = 1
- Detect the fixed stop using
- Contour deviation
- Reaching the torque limit
Fixed stop reached
DB31..61.DBX62.5 = 1
- Start timer TV_FXShold
(time period that the motor applies force against the brake)
Exit brake test
DB31..61.DBX11.0 = 0
Delete traversing command
DB31..61.DBX64.6 = 0 or
DB31..61.DBX64.7 = 0
- Monitoring time expired
- Delete traversing command
- Brake test in the NCK Exit
- Start TV_BTactiv
- Withdraw reduced torque limitation
Brake test no longer active
DB31..61.DBX71.0 = 0
Delete fixed stop reached
DB31..61.DBX62.5 = 0
- Open brake
- Output FB11 Done
Open brake *
DB31..61.DBX23.5 = 0
END

Notes :

FB11 "SI_BrakeTest" Description

Declaration of the function:

```
VAR_INPUT
        Start : BOOL ; //Start of the brake test
        Quit : BOOL ; //Acknowledge fault
        Bclosed : BOOL ; //Brake closed input (single channel -- PLC)
        Axis : INT ; //Testing axis No.
        TimerNo : TIMER ; //Timer from user
        TV_BTactiv : S5TIME ; //TimeValue --> brake test active
        TV_Bclose : S5TIME ; //TimeValue --> close brake
        TV_FeedCommand : S5TIME ; //TimeValue --> force FeedCommand
        TV_FXSreached : S5TIME ; //TimeValue --> Fixed stop reached
        TV_FXShold : S5TIME ; //TimeValue --> test brake
END_VAR
VAR_OUTPUT
        CloseBrake : BOOL ; //Signal close brake
        MoveAxis : BOOL ; //Move axis >> FC18
        Done : BOOL; //Brake Test Complete
        Error : BOOL; //Error
        State : BYTE ; //Error byte
END_VAR
```

To support the commissioning of the brake test Alarm 20096, "Axis %1 brake test aborted, additional information %2" can be enabled with MD 11411 $MN_ENABLE_ALARM_MASK, bit 5 = 1. This alarm supplies more detailed information if the brake test is interrupted.

FB11 Error Codes (VAR_OUT:STATE)

State	Reason
1	Start conditions not fulfilled. E.g. Axis not in closed loop control, brake closed, axis disabled...etc
2	No N/C checkback in "Brake Test Active" on selection of brake test.
3	No "Brake applied" checkback by input signal Bclosed
4	No traversing command output
5	Fixed stop is not reached --> axis RESET was initiated
6	Traversing inhibit/approach too slow --> fixed stop cannot be reached. Monitoring time TV_FXSreached has expired.
7	Brake is not holding at all (the end position is reached)/approach speed is too high
8	Brake opens during the holding time
9	Error when deselecting the brake test
10	Internal error
11	"PLC--controlled axis" signal not enabled in the user program

FB88 "SI_BRAKE_TEST"

In order to simplify the programming within Step 7, FB88 is provided to help with the sequence control. FB88 will transfer the axis to the PLC, then call FB11 and also call FC18.

Once the SBT has completed with or without error, FB88 will transfer the axis back to the NCK channel from the PLC.

The In/Out parameters of FB11 are passed to the interface of FB88 along with further signals for FC18 status information.

```
VAR_INPUT
  BT_Start : BOOL ;          //Start for Braketest
  Reset : BOOL ;   //Quit Error
  FB11_Bclosed : BOOL ;    //brake closed input(single channel - PLC)
  FB11_AxisNo : INT ;        //testing axisno.
  FB11_TimerNo : TIMER ;
  FB11_TV_BTactiv : S5TIME ;        //TimeValue -> braketest activ
  FB11_TV_Bclose : S5TIME ;         //TimeValue -> close Brake
  FB11_TV_FeedCommand : S5TIME ;          //TimeValue -> force FeedCommand
  FB11_TV_FXSreached : S5TIME ; //TimeValue -> Fixed stop reached
  FB11_TV_FXShold : S5TIME ;        //TimeValue -> test brake
  TimerNo_Bclosed_delay : TIMER ;
  Time_Bclosed_delay : S5TIME ;
END_VAR
VAR_IN_OUT
  FB11_CloseBrake : BOOL ;          //Signal Close brake
  FB11_Done : BOOL ;
  FB11_Error : BOOL ;
  FB11_State : BYTE ;       //Errorbyte
  FC18_InPos : BOOL ;
  FC18_Error : BOOL ;
  FC18_State : BYTE ;
END_VAR
```

Notes :

Test Stop Example: Safe Brake Test

In order to execute a test stop we need to now call FB88.

Within the S7 project, a prepared block (FC92 fcSI:TestStop") can be called from within FC90 "fcSI" as shown here.

```
Network 20 : X Axis End Limit Plus
      A     "Axis_X1".E_SN1Plus
      A     "Axis_X1".E_SN1Minus
      =     "Axis_X1".A_HWLimitPlus

Network 21 : Test Stop
      CALL  "fcSI:TestStop"
```

FC92 "fcSI:TestStop" contains sample code with a call to FB88 for the Safe Brake Test.

The block also contains a timer to request a TestStop every 30s (this would normally be set to 8 hours).

Network 7 of FC92 "fcSI"TestStop" contains the following call to FB88;

```
Network 7 : Call FB88 - Safe Brake Test
      AN    "Axis_Y1".E_HoldBrakeOpened
      =     "idbSI_SAFE_BT".FB11_Bclosed

      CALL  "fbSI_SAFE_BT" , "idbSI_SAFE_BT"
       BT_Start              :="mSI:TS:SBT:FB88Start"
       Reset                 :="Chan1".A_Reset
       FB11_Bclosed          :=
       FB11_AxisNo           :=2
       FB11_TimerNo          :="tSI:SBT:FB88Timer"
       FB11_TV_BTactiv       :=S5T#200MS
       FB11_TV_Bclose        :=S5T#200MS
       FB11_TV_FeedCommand   :=S5T#200MS
       FB11_TV_FXSreached    :=S5T#2S
       FB11_TV_FXShold       :=S5T#1S
       TimerNo_Bclosed_delay:="tSI:SBT:FB88BClsd"
       Time_Bclosed_delay    :=S5T#200MS
       FC18_Pos              :=-5.000000e+000
       FC18_FRate            :=1.000000e+002
       FB11_CloseBrake       :="Axis_Y1".A_BrakeToClose
       Done                  :="mSI:TS:SBT:FB88Done"
       FB11_Error            :="mSI:TS:SBT:FB88_FB11Err"
       FB11_State            :="mbSI:SBT:FB88:FB11State"
       FC18_InPos            :="mSI:SBT:FB88:FC18InPos"
       FC18_Error            :="mSI:SBT:FB88:FC18Error"
       FC18_State            :="mbSI:SBT:FB88:FC18State"
```

Check the axis number is correct (FB11_AxisNo) and that the correct brake status and command signals are passed to 'FB11_Bclosed' & 'FB11_CloseBrake'.

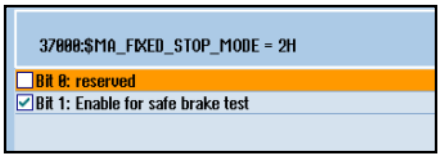
Note: 'FB11_Bclosed' must be supplied with the inverse of the axis DB brake status bit DB<Axis>.DBX92.5 or the block will return error code 1.

We must now set the necessary machine data for the Safe Brake Test function;

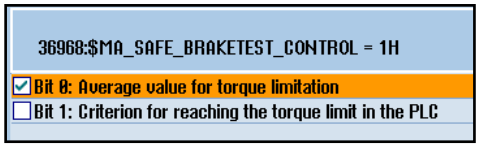
MD 11411 ENABLE_ALARM_MASK Bit 5 (20H)

This enable alarm 20096 to be displayed. This alarm will display detailed information relating Safe Brake Test in the event of an error.

MD37000 FIXED_STOP_MODE Bit 1 (2H)



36968 SAFE_BRAKETEST_CONTROL Bit 0 (1H)



MD36966 SAFE_BRAKETEST_TORQUE

This machine data describes the torque required to be applied to the brake during the safe brake test.
The value entered here is a percentage of the reference torque (p2003).

E.g. p2003 = 2.81Nm

The value can then be calculated as;

36966 = ((r1509 / p2003) * 100) * 1.3

r1509 (Actual Torque Setpoint) should be observed to determine the required holding torque that the drive uses to keep the axis in position.
(In this case, the axis is under no load so we can make an assumption and use a suitable value e.g. 0.25Nm).

36966 = ((0.25 / 2.81) * 100) * 1.3 = 23%

23% of p2003 = 0.64Nm
(During the test, this value can be observed in r80)

This equates to a value 30% greater than the force required to hold the axis in position.

Notes :

Check that the calculated value for the Safe Brake Test does not exceed the holding torque of the motor brake;

Motor Type	Brake Holding Torque (Nm)
1FK701_	0.4
1FK702_	1
1FK703_	1.9
1FK704_	4
1FK706_	13
1FK708_	22
1FK7100	23
1FK7101	43
1FK7103	
1FK7105	

Ensure the following have been completed;

- **Call FC92 “fcSI:TestStop” from FC90 “fcSI” and download FC92 and FC90 to the PLC.**
- **Set the machine data 11411, 37000, 36968 & 36966 as described.**
- **Copy & Confirm SI data and NCK Reset to activate.**
- **Once the Emergency Stop has been reset and the axes are enabled, User key T01 should be flashing to indicate that a test stop is required. (Ensure operator door is closed!).**
- **Press user key T01 to perform a Safe Brake Test. Observe the Torque Actual Value in r80. During the test, this should represent the value set in MD 36966 SAFE_BRAKETEST_TORQUE.**

Using the Trace function to observe the ‘Following Error’ & ‘Torque Setpoint’ during the brake test;

Once FXS (Fixed Stop) has been reached, the axis remains in FXS mode for the duration of ‘FB11_TV_FXShold’.

If FB11_TV_FXSReached is too short, FB11 will return Error code 6. NC Alarm ‘20096 Axis Y1 Brake Test Cancelled’ will also be displayed.

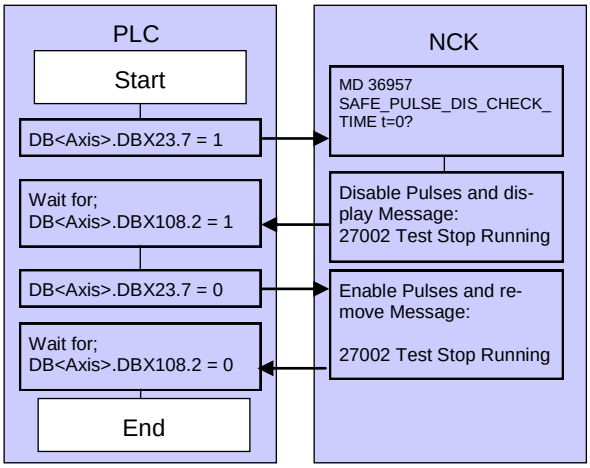
Notes :

Test Stop Example: Pulse Disable Test

Once the Safe Brake Test has completed successfully, the second part of the 'Test Stop' can now be enabled.

The 'Pulse Disable' test checks the shutdown paths of both NCK & Drive monitoring channels.

The test is initiated by setting DB<Axis>.DBX23.7 (Test Stop Selection) and the response is monitored by DB<Axis>.DBX108.2 (Pulses Cancelled) within the Drive and operator message '27002 Axis ?? Test stop is running' within the NC.



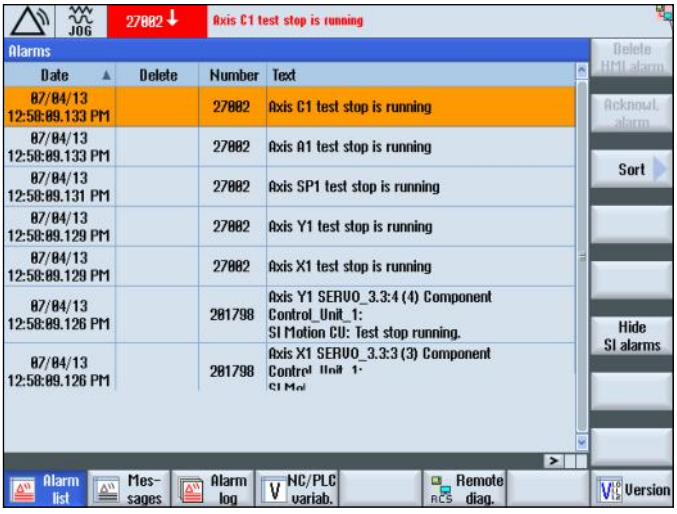
FC86 "fcSI_TSTSTP_SL" can be used to carry out this test.

- Call FC86 within FC92 Network 9 as shown here;

```
Network 9 : Test Pulse Disable
      CALL  "fcSI_TSTSTP_SL"                          FC86
       start      :="mSI:SBT:Complete"                M70.0
       clear      :="Chanl".A_Reset                   DB21.DBX7.7
       num_axis   :=5
       test_axis_1:=1
       test_axis_2:=2
       test_axis_3:=4
       test_axis_4:=6
       test_axis_5:=7
       test_axis_6:=0
       aux_dword  :="mdSI:TS:FC86aux_dword"           MD72
       ready      :="mSI:TS:FC86ready"                M70.1
       serror     :="mSI:TS:FC86serror"               M70.2
       error      :="mSI:TS:FC86error"                M70.3

      A     "mSI:TS:FC86ready"                       M70.1
      S     "mSI:TS:SPT:Complete"                    M70.4

Network 10 : Safety Integrated: Safe Brake Test: Complete
      A     "mSI:SBT:Complete"                       M70.0
      A     "mSI:TS:SPT:Complete"                    M70.4
      R     "mSI:TS:RequestActive"                   M60.5

Network 11 : Safety Integrated: Test Stop is Active
      ON    "mSI:TS:RequestActive"                   M60.5
      O     "Chanl".A_Reset                          DB21.DBX7.7
      R     "mSI:TS:SBT:FB88Start"                   M61.1
      R     "mSI:SBT:Complete"                       M70.0
      R     "mSI:TestStopActive"                     M60.0
      R     "mSI:TS:SPT:Complete"                    M70.4
      R     "mSI:TS:ExtStp:Complete"                 M76.6
```

- Modify Network 10 & 11 to include M70.4
- Load the modified FC92 to the PLC and run the Test Stop and observe the 27002 messages for each axis.

Increasing MD 36957 to ~0.5s will allow the 27002 messages to be displayed for longer.



You should also now see messages '201798 Test Stop Running' from the Drive monitoring channel being displayed.

Notes :

Test Stop Example: External Stop Test

In order to test the 'External Stop's' (A, C & D), the PLC user program must explicitly request a STOP in the PLC-SPL(FC91 "fcSI:PLC-SPL") and then in the NCK-SPL(SAFE.SPF).

This can be done by interlocking the 'Safety Related Internal Output' ($A_OUTSI[n]) with a standard 'non-safe' signal generated by the user program. The 'non-safe' signal can then be passed through to the NCK-SPL using the single channel PLCSIIN[n] interface.

The status of the External STOP's can be monitored with DB<Axis>.DBX111.4,5 & 6 for STOP A, C & D.

FC87 "fcSI_EXTSTP_SL" is provided for the purpose of carrying out the 'External Stop Test'.

- In FC92, Call FC87 and parameterize as shown;

```
Network 10 : Test External Stops
      CALL  "fcSI_EXTSTP_SL"                          FC87
       start        :="mSI:TS:SPT:Complete"          M70.4
       clear        :="Chan1".A_Reset                DB21.DBX7.7
       num_axis     :=5
       test_axis_1  :=1
       test_axis_2  :=2
       test_axis_3  :=4
       test_axis_4  :=6
       test_axis_5  :=7
       test_axis_6  :=0
       test_axis_7  :=0
       test_axis_8  :=0
       req_stopd_plc:="mSI:TEST_D"                   M76.0
       req_stopc_plc:="mSI:TEST_C"                   M76.1
       req_stopa_plc:="mSI:TEST_A"                   M76.2
       req_stopd_nck:="SPL".IP2_TEST_D               DB18.DBX132.1
       req_stopc_nck:="SPL".IP3_TEST_C               DB18.DBX132.2
       req_stopa_nck:="SPL".IP4_TEST_A               DB18.DBX132.3
       aux_dword    :="mdSI:TS:FC87aux_dword"        MD78
       ready        :="mSI:TS:FC87ready"             M76.3
       serror       :="mSI:TS:FC87serror"            M76.4
       error        :="mSI:TS:FC87error"             M76.5

      A     "mSI:TS:FC87ready"                       M76.3
      S     "mSI:TS:ExtStp:Complete"                 M76.6

Network 11 : Safety Integrated: Safe Brake Test: Complete
      A     "mSI:SBT:Complete"          M70.0
      A     "mSI:TS:SPT:Complete"       M70.4
      A     "mSI:TS:ExtStp:Complete"    M76.6
      R     "mSI:TS:RequestActive"      M60.5

Network 12 : Safety Integrated: Test Stop is Active
      ON    "mSI:TS:RequestActive"      M60.5
      O     "Chan1".A_Reset             DB21.DBX7.7
      R     "mSI:TS:SBT:FB88Start"      M61.1
      R     "mSI:SBT:Complete"          M70.0
      R     "mSI:TestStopActive"        M60.0
      R     "mSI:TS:SPT:Complete"       M70.4
      R     "mSI:TS:ExtStp:Complete"    M76.6
```

Note: UDT18 has been updated to include the PLCSIIN symbolic names.

- Modify Network 11 & 12 to include M76.6
- Modify the PLC-SPL FC91 "fcSI:PLC-SPL"

```
Network 12 : IDS=211: Assign all axes STOP A DES to SIRELAY OUT
      A     "SPL".SPL_DATA.MI7_SIREL1_OUT1      DB18.DBX70.6
      AN    "mSI:TEST_A"                        M76.2
      =     "SPL".SPL_DATA.OI4_STOPA_DES_AX     DB18.DBX62.3

Network 13 : IDS=212: Assign all axes STOP C DES to SIRELAY OUT
      A     "SPL".SPL_DATA.MI6_SIREL1_OUT0      DB18.DBX70.5
      AN    "mSI:TEST_C"                        M76.1
      =     "SPL".SPL_DATA.OI5_STOPC_DES_AX     DB18.DBX62.4

Network 14 : IDS=213: Statically deselect all axes STOP D
      A     "mTRUE"                             M1.1
      AN    "mSI:TEST_D"                        M76.0
      =     "SPL".SPL_DATA.OI6_STOPD_DES_AX     DB18.DBX62.5

Network 15 : IDS=214: Assign Spindle STOP A DES to SIRELAY OUT2
      A     "SPL".SPL_DATA.MI8_SIREL1_OUT2      DB18.DBX70.7
      AN    "mSI:TEST_A"                        M76.2
      =     "SPL".SPL_DATA.OI8_STOPA_DES_SP     DB18.DBX62.7

Network 16 : IDS=215: Assign Spindle STOP C DES to SIRELAY OUT0
      A     "SPL".SPL_DATA.MI6_SIREL1_OUT0      DB18.DBX70.5
      AN    "mSI:TEST_C"                        M76.1
      =     "SPL".SPL_DATA.OI9_STOPC_DES_SP     DB18.DBX63.0

Network 17 : IDS=216: Statically deselect Spindle STOP D
      A     "mTRUE"                             M1.1
      AN    "mSI:TEST_D"                        M76.0
      =     "SPL".SPL_DATA.OI10_STOPD_DES_SP    DB18.DBX63.1
```

- Download FC92 & FC91 to the PLC
- Edit the NCK-SPL to DEFINE symbolic names for the $A_PLCSIIN[n] variables.

```
DEFINE MI9_SIREL1_OUT3    AS $A_MARKERSI[9]¶
¶
;=======================================¶
;Single Channel Inputs from PLC - $A_PLCSIIN¶
DEFINE IP1_PLC_READY      AS $A_PLCSIIN[1]¶
DEFINE IP2_TEST_D         AS $A_PLCSIIN[2]¶
DEFINE IP3_TEST_C         AS $A_PLCSIIN[3]¶
DEFINE IP4_TEST_A         AS $A_PLCSIIN[4]¶
¶
;=======================================¶
;SIRELAY(1)¶
DEFINE M_STATUS_1         AS $AC_MARKER[1]¶
```

- Interlock the 'External Stop De-selection' internal outputs ($A_OUTSI[n]) for the axes and spindle.

```
;Net 12: Assign all axes STOP A deselection to SIRELAY OUT1(1.0s de
lay)¶
IDS=211 DO OI4_STOPA_DES_AX = (MI7_SIREL1_OUT1 AND (NOT IP4_TEST_A)
)¶
¶
```

Notes :

```
;Net 13: Assign all axes STOP C deselection to SIRELAY OUT0(No dela
y)¶
IDS=212 DO OI5_STOPC_DES_AX = (MI6_SIREL1_OUT0 AND (NOT IP3_TEST_C)
)¶
¶
```

```
;Net 14: Statically deselect all axes STOP D¶
IDS=213 DO OI6_STOPD_DES_AX = (NOT IP2_TEST_D)¶
¶
```

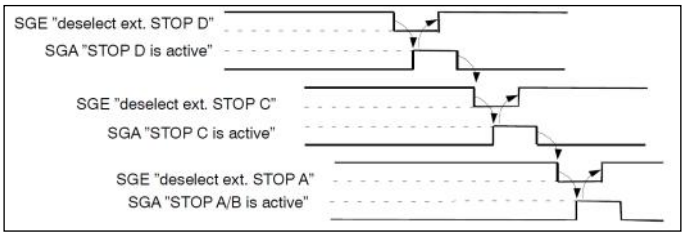
```
;Net 15: Assign Spindle STOP A deselection to SIRELAY OUT2(2.8s del
ay)¶
IDS=214 DO OI8_STOPA_DES_SP = (MI8_SIREL1_OUT2 AND (NOT IP4_TEST_A)
)¶
¶
```

```
;Net 16: Assign Spindle STOP C deselection to SIRELAY OUT0(No delay
)¶
IDS=215 DO OI9_STOPC_DES_SP = (MI6_SIREL1_OUT0 AND (NOT IP3_TEST_C)
)¶
¶
;Net 17: Statically deselect Spindle STOP D¶
```

```
;Net 17: Statically deselect Spindle STOP D¶
IDS=216 DO OI10_STOPD_DES_SP = (NOT IP2_TEST_D)¶
¶
```

- Save the SAFE.SPF file and Execute an NCK Reset to activate the changes.
- Test Stop will now perform three tests;
  1) Safe Brake Test
  2) Pulse Disable Test
  3) External Stop Test

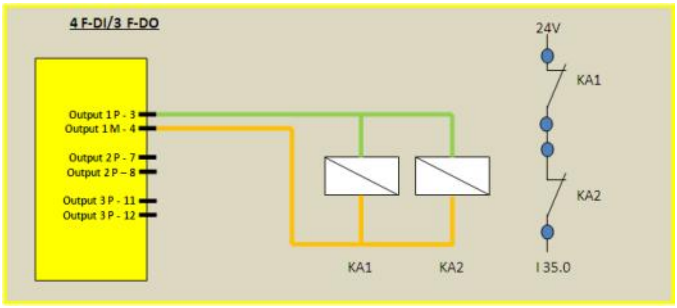
The External Stop's D,C & A will now be tested individually both the NCK and the Drive monitoring channels.



Test Stop Example: PLC SPL IO Test (KA1 & KA2)

The final test to be carried out is for the KA1 & KA2 contactors that are controlled by $A_OUTSE[1] (OE1_KA1_KA2).

The check-back circuit is connected to I36.0.



The contactors must be briefly switched off to test the circuit.

FB89 is provided for this purpose. The check back signal is expected to be seen within 1 second of the test signal.

The test signal should be interlocked with the safe output within the SPL.

- Modify FC92 (fcSI:TestStop) to add a new network as shown here: (Note: UDT18 has been edited to include the symbolic name for $A_PLCSIIN[5] := "SPL".IP5_TEST_KA1_KA2)

```
Network 11: Test SPL IO - KA1 & KA2 Contactors
      CALL  "fbSPL_IO_TestStop" , "idbSPL_IO_TestStop"    FB89 / DB89
       Start     :="mSI:TS:ExtStp:Complete"               M76.6
       FDOStatus:="SPL".SPL_DATA.OE1_KA1_KA2              DB18.DBX46.0
       Feedback :="iPP72x48:X222.11:36.0"                 I36.0
       Test      :="SPL".IP5_TEST_KA1_KA2                 DB18.DBX132.4
       Error     :="mSI:TS:SPL_IOTestError"               M82.0
```

- Now modify the PLC-SPL FC91 "fcSI:PLC-SPL"

```
Network 7: IDS=206: Assign MarkerSIP[6] to KA1_KA2 Contactor Output
      A     "SPL".SPL_DATA.MI6_SIREL1_OUT0    DB18.DBX70.5      -- MARKERSIP6
      AN    "SPL".IP5_TEST_KA1_KA2            DB18.DBX132.4     -- PLCSIIN_5
      =     "SPL".SPL_DATA.OE1_KA1_KA2        DB18.DBX46.0      -- OUTSEP1
```

- The NCK-SPL(SAFE.SPL) must also be modified to include the changes made to the PLC-SPL.

```
;=====================================¶
;Single Channel Inputs from PLC - $A_PLCSIIN¶
DEFINE IP1_PLC_READY    AS $A_PLCSIIN[1]¶
DEFINE IP2_TEST_D       AS $A_PLCSIIN[2]¶
DEFINE IP3_TEST_C       AS $A_PLCSIIN[3]¶
DEFINE IP4_TEST_A       AS $A_PLCSIIN[4]¶
DEFINE IP5_TEST_KA1_KA2 AS $A_PLCSIIN[5]¶
¶
```

```
¶
;Net 7: Assign $A_MARKERSI[6] to KA1_KA2 Contactor Output¶
IDS=206 DO OE1_KA1_KA2 = (MI6_SIREL1_OUT0 AND (NOT IP5_TEST_KA1_KA2
))¶
¶
```

- Save the NCK-SPL, NCK Reset and test

Notes :

NCK-SPL Protection

Once the logic is complete, the 'NCK-SPL Protection' must be enabled to remove the alarm '27005 NCK SPL protection not activated'.

This is done in two parts, once in the NCK and also within the PLC.

To activate the NCK-SPL Protection;

- Within OB100, set DB18.DBX36.0 = 1

```
Network 2 : Emergency stop
      SET
      R     "NC".A_EMERGENCY            DB10.DBX56.1
      S     "SPL".SPL_READY             DB18.DBX36.0
```

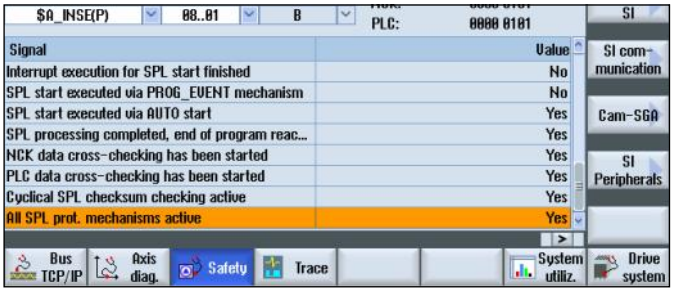
- Using MD 11500[0..1], specify the range of Safe' synchronous actions used within the NCK-SPL (SAFE.SPF).

11402	$MN_PLC_OB40_TRACE_DEPTH	2	po	active (cf)
11500[0]	$MN_PREVENT_SYNACT_LOCK	200	po	
11500[1]	$MN_PREVENT_SYNACT_LOCK	220	po	Reset (po)
11510	$MN_IPO_MAX_LOAD	0 %	po	

Here you can see the range has been set from 200 to 220. This will cover the actual upper limit of 'Safe Synchronous Actions' actually used.

- Two consecutive NCK-Reset's are now required to fully activate the protection. After the first NCK reset, the error 27090 Error in data cross check NCK-PLC' will be active.

You can observe if the protection is successfully activated within the 'Diagnostics' >> 'Safety' >> 'SPL' screen



The status of the SPL protection can also be detected within the PLC using DB18.

DB18.DBX137.4 SPL_STATUS[13] 'All SPL protective mechanisms active'.

Notes :

Safety Integrated Acceptance Test (SIAT):

- Description - SINUCOM NC
- Pulse Disable Paths
- External STOP’s
- SPL I/O
- Emergency Stop
- Functional Relationships
- Safe Operational Stop
- Safe Reduced Velocity
- Safe Software Limits
- Safe Brake Test
- Saving the Results and Generating the Certificate

Notes :

Description

The requirements associated with an acceptance test can be derived from the EU Machinery Directive. Accordingly, the machine manufacturer (OEM) is responsible for the following:

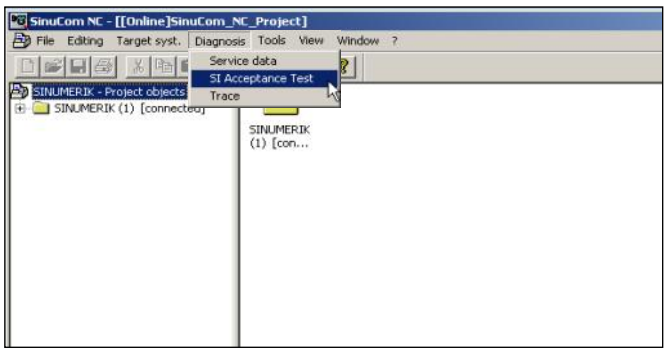
- To carry out an acceptance test for safety--related functions and machine parts, and….
- ….to issue an ”Acceptance certificate” that includes the results of the test.

When using the Safety Integrated function, the acceptance test is used to check the correct configuring of the SI monitoring functions used in the NCK, PLC and drive. The test objective is to verify proper implementation of the defined safety functions, to check test mechanisms (forced checking procedure measures) and to examine the response of individual monitoring functions by specifically violating tolerance limits. This should be carried out for all safety functions, i.e. for the axial monitoring functions, the SPL, the safety-related communication relationships, the safety--related I/O etc.

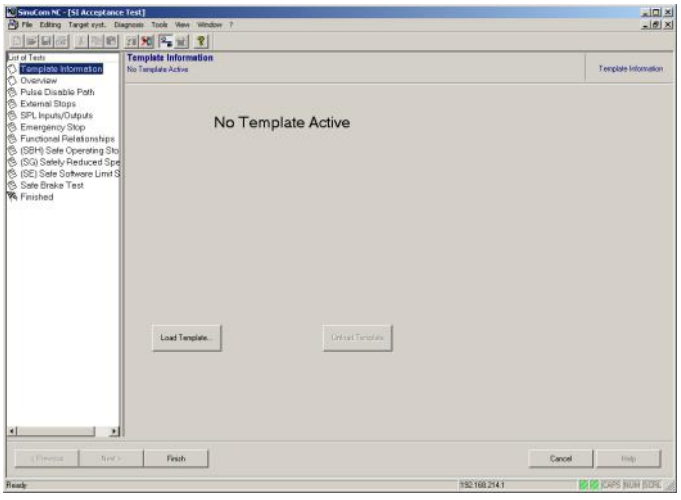
Warning:
If machine data for SI functions is changed, a new acceptance test must be carried out for the modified SI function and recorded in the acceptance report.

The SIAT is carried out by using SinucomNC V7.6 or higher.

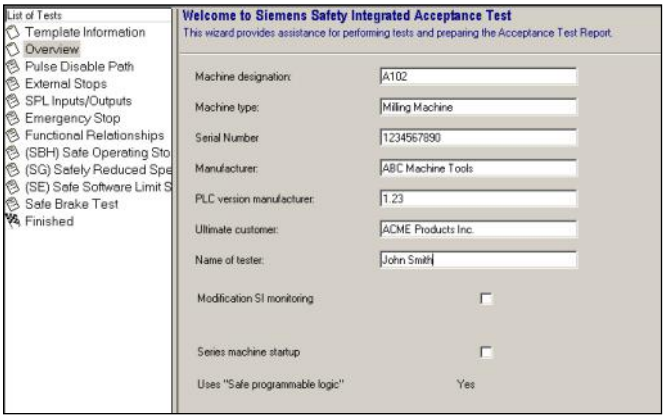
Once SinucomNC has started, select the ‘Diagnosis’ >> ‘SI Acceptance Test’ menu item.



After a short delay and once the software has initialized, the following screen should be visible.



Select the item ‘Overview’ from the list on the left hand side and fill out the required information.



The tick-boxes for ’Modification SI monitoring’ and ‘Series machine startup’ should not be selected for the first time the SIAT is carried out.

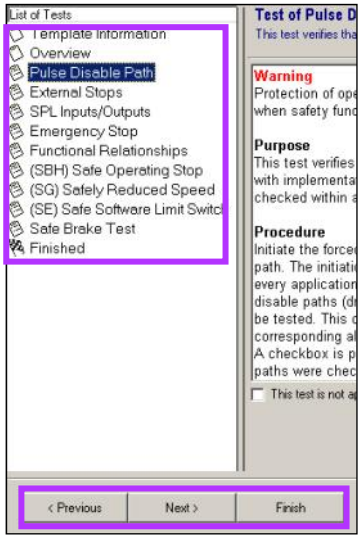
’Modification SI monitoring’ is used to record a subsequent modification to an already tested machine.

‘Series machine startup’ is selected for subsequent identical machines. This enables a reduced scope of tests to be carried out.

Notes :

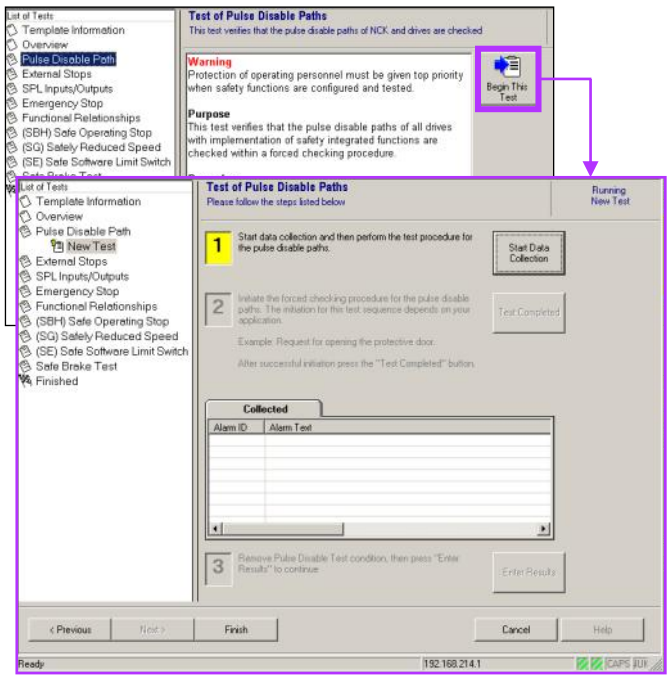
Navigation

Using the ‘Next’ and ‘Previous’ buttons in the bottom left corner of the application will allow navigation through the various tests. You can also directly click on the list of tests displayed on the left hand side of the application.



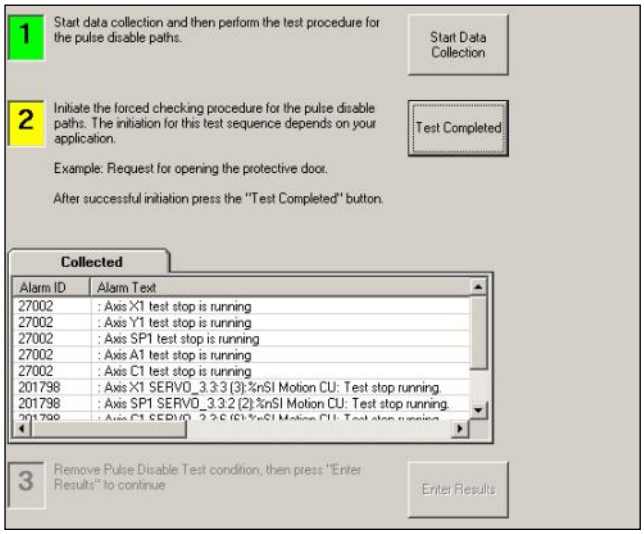
Test of Pulse Disable Paths

With the ‘Pulse Disable Path’ item selected, click on ‘Begin this test’.



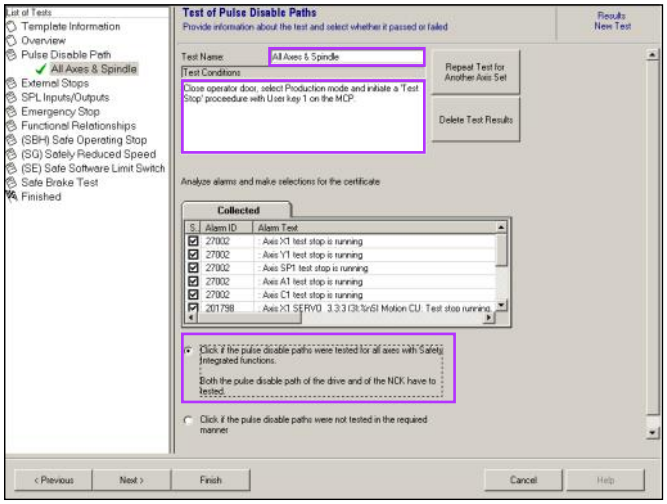
Now execute a test stop procedure (User key 1 on MCP).

This test is testing the functionality provided by the call of FC86 in the test stop routine.



Once the test stop has completed, click on ‘Test Completed’ followed by ‘Enter Results’

Within the ‘Enter Results’ screen, give the test a name, specify the test conditions and select the radio button for a successful test as shown here;



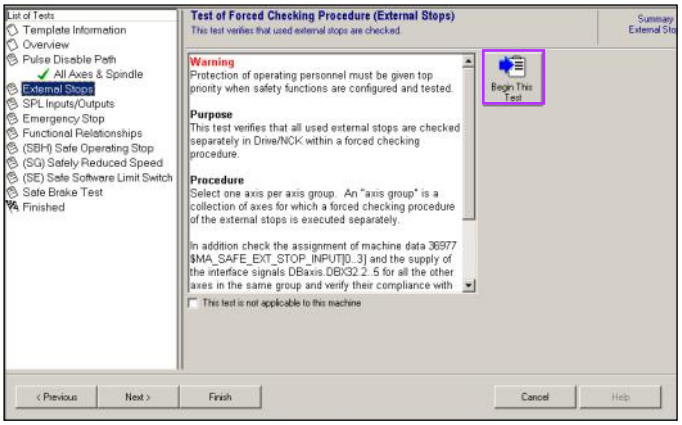
As we tested all axes including the spindle in the same group (only 1 call of FC86) we do not need to repeat the test for another axis set.

Click ‘Next>‘ to advance to the next test.

Notes :

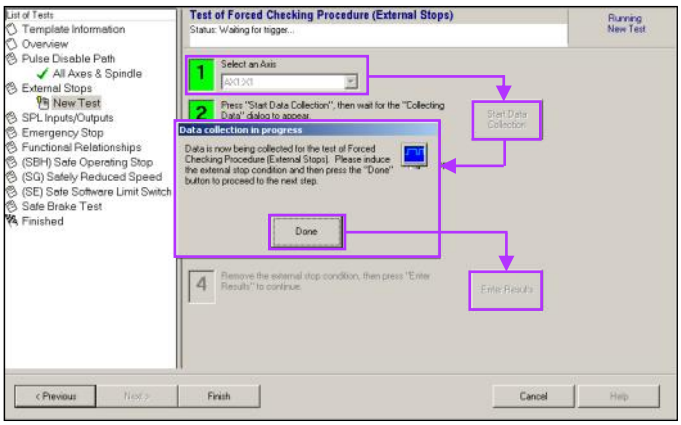
External Stops Test

The 'External Stop' test should be carried out for at least 1 axis from each 'Group'. A 'Group' contains axes that are switched together. This can be determined by the values of MD36977 SAFE_EXT_STOP_INPUT[0..3]. In this case, X1, Y1, A1 & C1 are in group 1 and the Spindle is in Group 2.

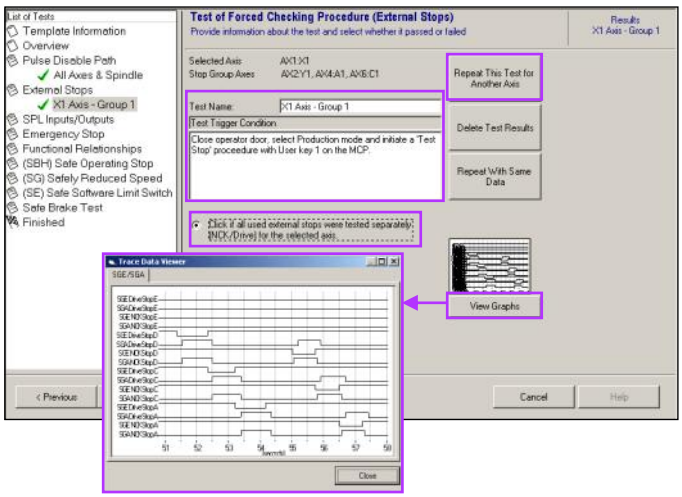


With the 'External Stop Test' page displayed, click on 'Begin this Test' to continue.

- Select an axis from group 1
- Click on 'Start Data Collection. Wait for the dialog to appear.
- Execute a Test Stop (MCP User Key 1)
- Acknowledge Dialogue with 'Done' when Test Stop is complete.
- Click on 'Enter Results' when available.

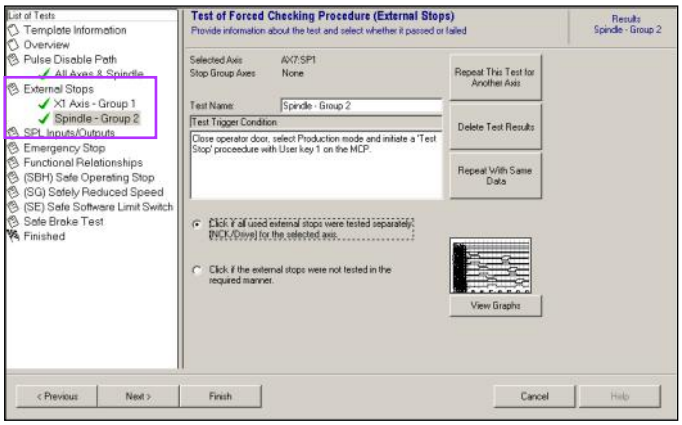


- Click on 'View Graphs' to ensure that the signal states have switched correctly.
- Enter a test name, and specify the test conditions.
- Acknowledge that the test completed successfully.



- Now click 'Repeat This Test for Another Axis' and select the Spindle.

- Enter the Test Name, Conditions and acknowledge that the test completed successfully.
- Click 'Next' to advance to the next test.



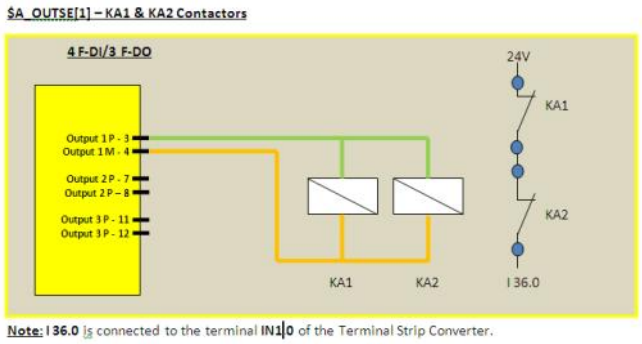
Notes :

SPL Inputs/Outputs Test

Purpose

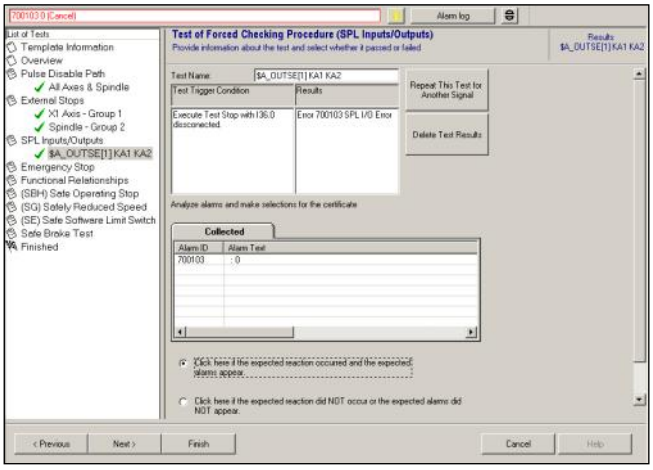
This test verifies that relevant SPL-IO (for example output circuits that are switched off for safety reasons under certain conditions) are checked within a forced checking procedure. So when a check-back input is removed, the appropriate reactions and alarms should be generated.

$A_OUTSE[1] controls KA! & KA2 relays. They have a check-back circuit connected to I36.0.



The check-back circuit (I36.0) will be disconnected for this test to ensure that the fault is detected.

- Temporarily disconnect I36.0 from terminal strip (IN1.0).
- Click on ‘Start Data Collection’
- Execute a ‘Test Stop’
- Error 700103 should be collected
- Enter Results, de-selecting all messages and alarms except error 700103.
- Enter test name, trigger condition, results...etc.



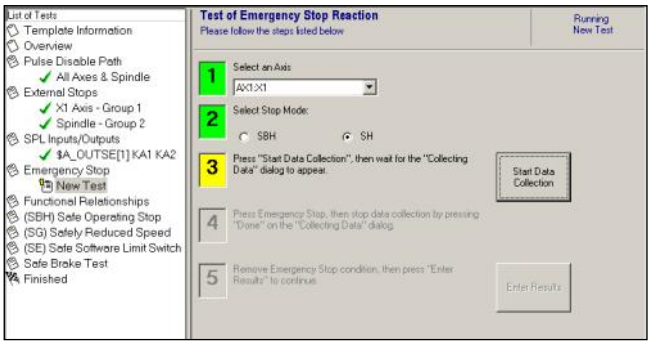
- Acknowledge the successful detection of the error.
- Click Next to advance to next test.
- Re-connect I36.0 to terminal strip IN1.0.
- Reset alarm with MCP Reset.

Emergency Stop Test

Purpose

This test verifies the reaction of the emergency stop function for axes if the emergency stops are implemented via the external stops of the safe programmable logic (SPL).

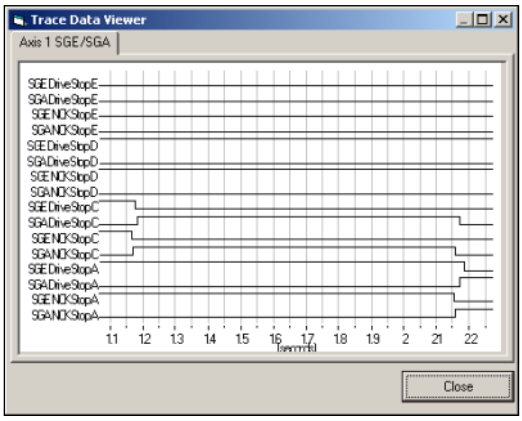
- With the Emergency Stop test selected, click on ‘Begin This Test’
- Select the first of two axes.
- Select ‘SH’. (In this case, Emergency Stop results in ‘SH’ (Safe Standstill or STO) using External STOP C >> STOP A.)



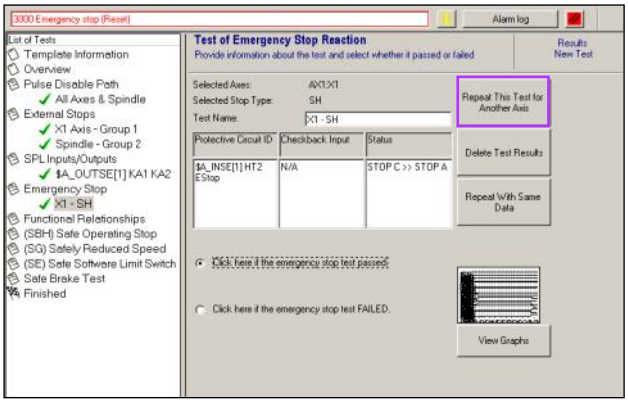
- Click on ‘Start Data Collection’.

Notes :

- Wait for 'Data Collection in Progress' dialogue to appear.
- Initiate Emergency Stop using HT2 Emergency Stop Button.
- After STOP A is selected (use the Safety Diagnostics screen to observe the currently active STOP), click done to close dialogue.
- View the graph to see the transition from STOP C to STOP A.

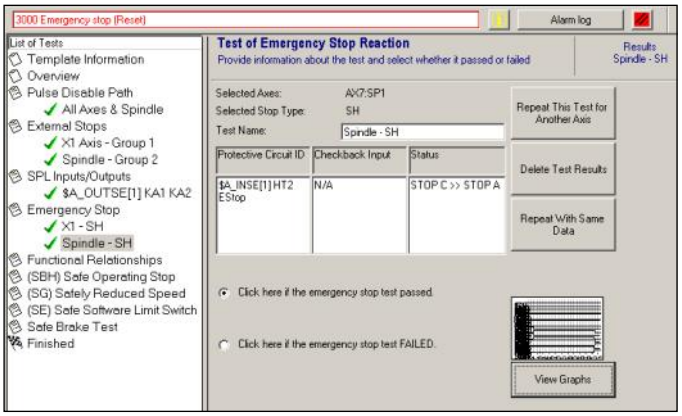


- Enter the Test Name and other details e.g as shown;



- Acknowledge that the test was successful
- Repeat the test for another axis e.g. Spindle.

- Enter the test data for the Spindle e.g. as shown;



- Click 'Next' to select the next test.

Functional Relationships Test

Purpose

This test verifies functional relationships. The relationship between the active monitoring function of the relevant axis and the mode / sensor technology is verified.

The Functional Relationship Table created by the OEM is now used to confirm that the SPL operation in accordance with that table;

Each condition described by the function table should be brought about and a check made using the HMI Safety Diagnostics screen to ensure that the correct Safe state is active.

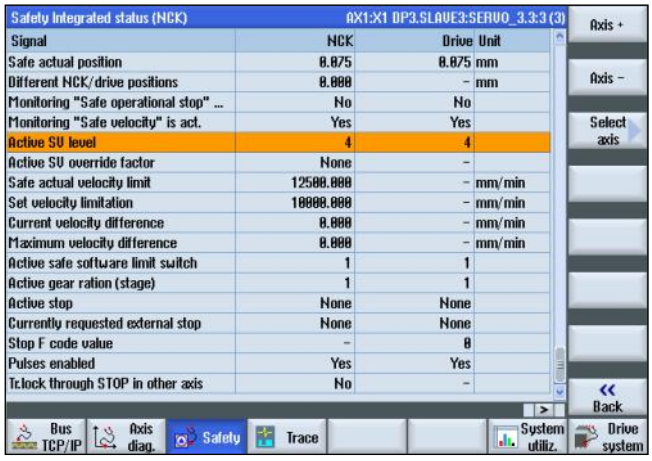
Notes :

Functional Relationships Test con't

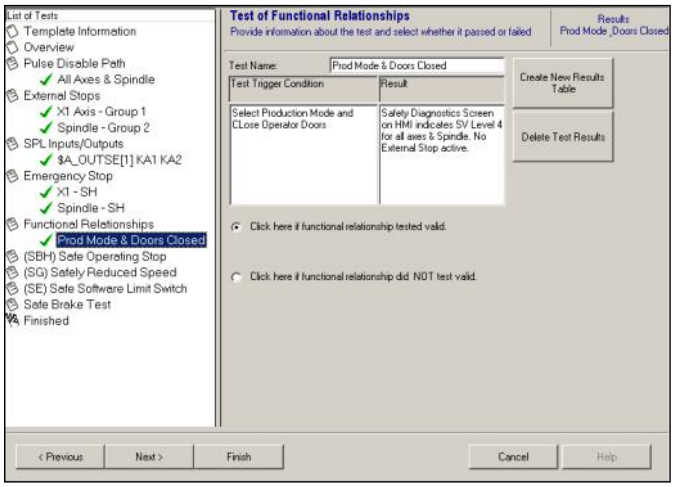
Functional Relationships Table:

Operating mode	*Protective doors*	*Axis / spindle*	*Monitoring*
Production	Closed	X1, Y1, A1, C1 SP1	SG4 SG4
	Released	X1, Y1, A1, C1 SP1	SBH SBH
Setup	Closed	X1, Y1, A1, C1 SP1	SG2 SG2
	Released	X1, Y1, A1, C1 SP1	SG1 SBH / SG1 With HT2 Enable

- Select the first condition: 'Production Mode' and 'Protective Doors' closed.
- Using the Safety Diagnostics Screen, check that all axes and spindle have an 'Active SV Level' of 4 in both NCK and Drive.
- Also Check that there are no 'Active Stop's' selected.



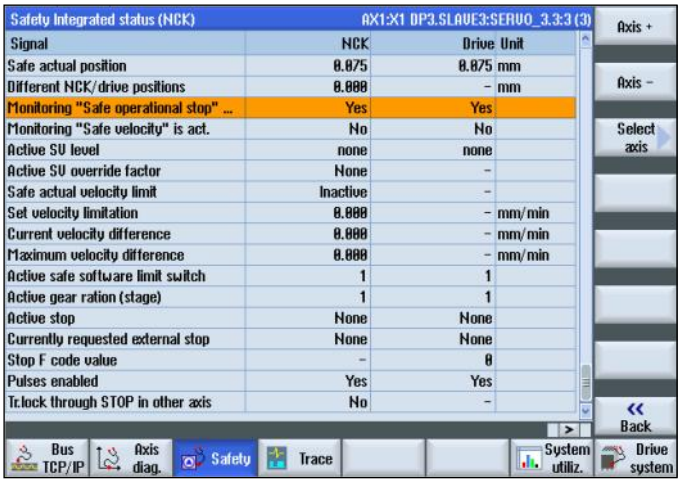
- Click 'Begin This Test' in SinucomNC.
- Click 'Enter Results' and enter a Test Name, trigger condition and Result. E.g. as shown here;



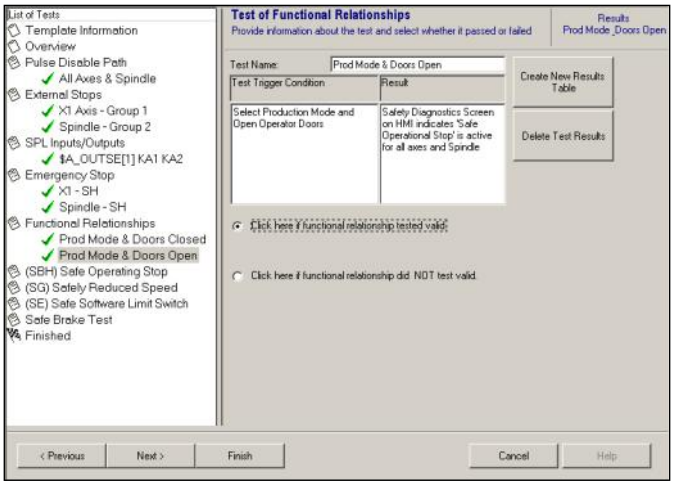
- Acknowledge a valid test
- Click on 'Create New Results Table'.

Notes :

- Create the second condition: Open the Operator Door.
- Using the Safety Diagnostics Screen, check that all axes are in a ‘Safe Operational Stop’ state.



- Enter the test results. E.g. as shown here;



- Repeat this procedure for all conditions described by the function table. There should be a test for the following;
  1. Production Mode with doors closed.
  2. Production Mode with doors open.
  3. Setup Mode with doors closed.
  4. Setup Mode with doors open.
  5. Setup Mode with doors open and HT2 Enable.
- Once successfully completed, click ‘Next’ to advance to the next test.

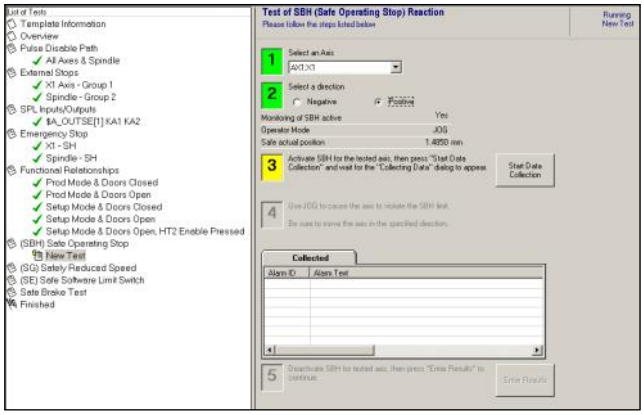
Safe Operating Stop Test

Purpose

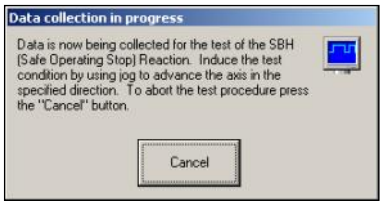
This test verifies that exceeding the safe operating stop tolerance limit results in a drive shutdown.

WARNING: THE SELECTED AXIS WILL MOVE DURING THIS TEST!

- With the Safe Operating Stop test selected, click on ‘Begin This Test’.
- Select the first axis: X1 and select a suitable direction.



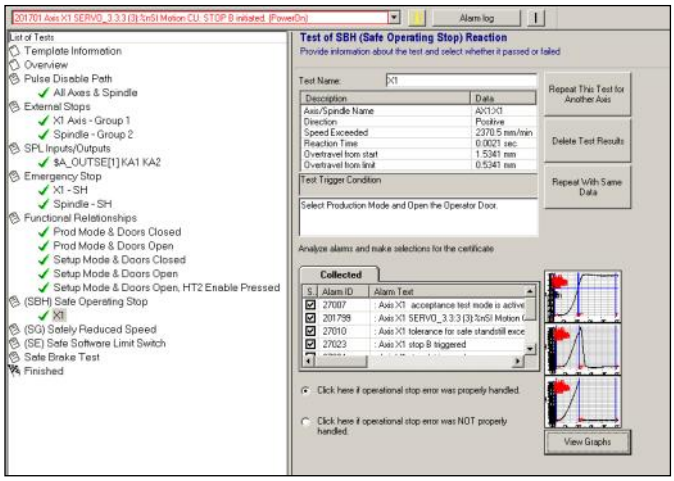
- Now create a ‘Safe Operational Stop’ condition with ‘Production Mode’ and ‘Operator Door’ open.
- Select Jog Mode.
- Click on ‘Start Data Collection’.
- Wait for the data collection dialogue to appear;



- Jog the axis in the direction selected for the test.
- Wait for the data to be collected and click ‘Enter Results’.

Notes :

- Enter test description data. E.g. as shown here;



- Observe results data and graphs, OEM to confirm results are within specification.
- Acknowledge test was correctly handled.
- Reset Alarms.
- Repeat this test for all axes and spindle.
- Click ‘Next’ to advance next test.

Safe Reduced Velocity Test

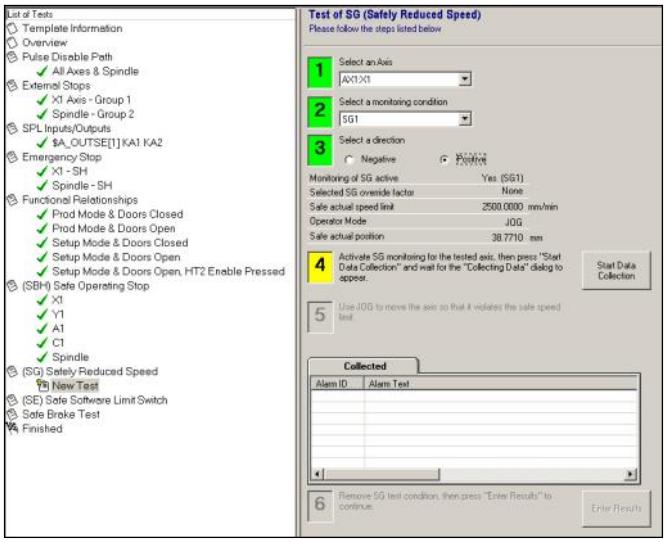
Purpose

This test verifies that safe speed limit violations are detected. If a safe speed limit violation occurs, the appropriate alarm(s) should be generated.

WARNING: THE SELECTED AXIS WILL MOVE DURING THIS TEST!

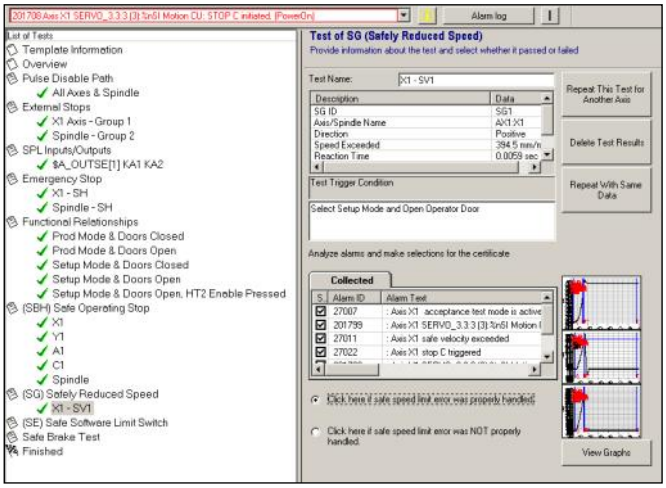
Note: If, during this test, the error ‘**27013 Safe Monitoring For Acceleration Exceeded**’ is displayed, then the MD 36948 SAFE_STOP_VELO_TOL may need to be increased or the drive may require tuning.

- With the ‘Safe Reduced Velocity’ test selected, click on ‘Begin This Test’.
- Select the X1 axis
- Select a Safe Velocity Level (i.e. SV = 1)
- Open ‘Operator Door’ and Select ‘Setup Mode’ (This will select SV Level 1 for all axes).
- Select a suitable direction for the test.



- Click on ‘Start Data Collection’ and wait for dialogue box.
- Jog the axis in the selected direction.
- Wait for dialogue box to clear and click ‘Enter Results’.

- Enter test description data. E.g. as shown here;



- Verify test results are acceptable and acknowledge a successful test.
- Reset Errors
- Repeat test for all Axes & Spindle and all used SV Levels.

  Note: It may not be possible or practical to test the Maximum Safe Velocity Limit (i.e. SV4) without error due to the limitation of the axis machine data and mechanical components.

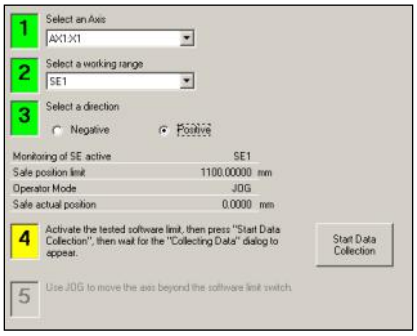
Notes :

Safe Software Limit Switch Test

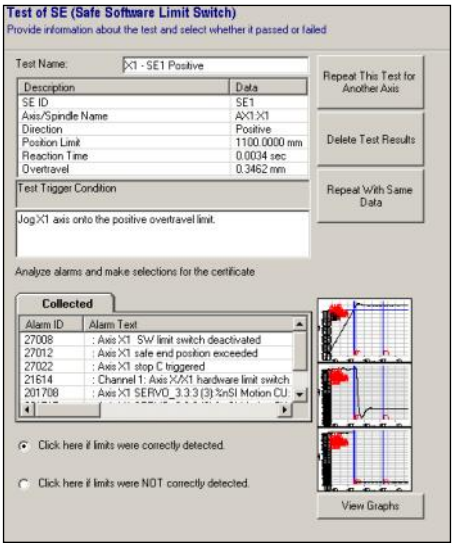
Purpose

This test verifies that limit switch overtravel is detected. If limit switch overtravel occurs, the appropriate alarm(s) should be generated and the drive should be shut down.

- Ensure X1 axis is 'Safely Referenced' and the 'User Agreement is set'.
- With the 'Safe Software Limit Switch Test' selected, click on 'Begin This Test'.
- Select the X1 axis and SE1 (Safe End-switch Pair 1)
- Select the Positive Direction and click 'Start Data Collection'



- Jog the X1 axis onto the Positive Limit
- Click 'Enter Results'
- Enter the Test Name and Trigger Condition.



- Verify result data and acknowledge the test.
- Recover the axis using the 'End Limit Override' keyswitch.

- Repeat the test for the SE1 Negative direction.
- Enter the test data and acknowledge the test.

In order to test the SE2 limits It is necessary to temporally modify the PLC user program.

The Hardware Over-travel interface signals must be inhibited for the axis (DB<Axis>.DBX12.0 & 1). This is done in FC90 as shown here.

```
Network 19 : X Axis End Limit Minus
      AN    "Axis_X1".E_SN1Plus          DB31.DBX109.0
      AN    "Axis_X1".E_SN1Minus         DB31.DBX109.1
      A     "SPL".SPL_STATUS14           DB18.DBX137.5
      CLR
      =     "Axis_X1".A_HWLimitMinus     DB31.DBX12.0

Network 20 : X Axis End Limit Plus
      A     "Axis_X1".E_SN1Plus          DB31.DBX109.0
      A     "Axis_X1".E_SN1Minus         DB31.DBX109.1
      CLR
      =     "Axis_X1".A_HWLimitPlus      DB31.DBX12.1
```

Once this done, SE2 can then be tested by selecting the 'End Limit Override' keyswitch, whilst jogging the axis onto the SE2 limit.

It is only possible to recover the axis by removing the 'User Agreement' before jogging clear.

WARNING: CARE MUST BE TAKEN TO AVOID MACHINE DAMAGE!

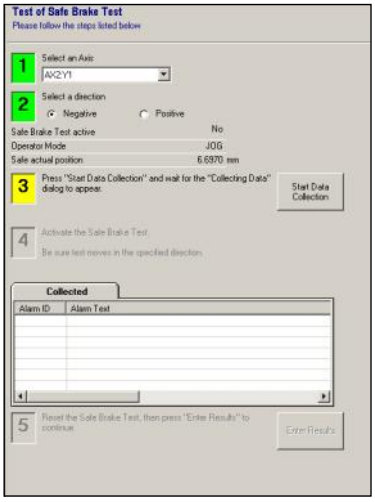
Note: Remember to remove the modifications from the PLC.

Notes :

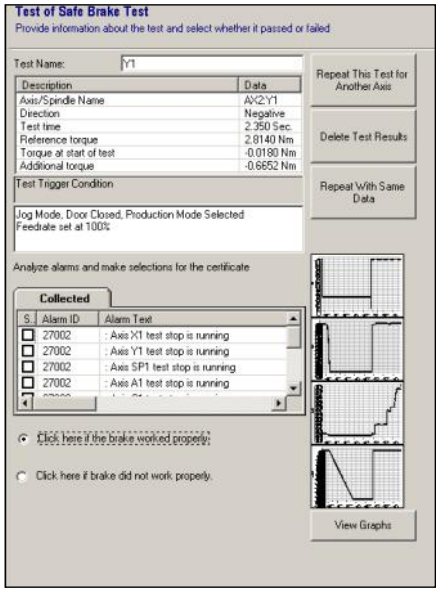
Safe Brake Test

Purpose
This test verifies the holding torque and the functionality of the brake.

- With the ‘Safe Brake Test’ selected, click on ‘Begin This Test’.
- Select the Y1 axis and a Negative Direction.
- Click ‘Start Data Collection’.



- Ensure the Feedrate is set to 100%.
- Execute a ‘Test Stop’ (MCP User Key 1).
- When completed, enter the test data and trigger conditions.



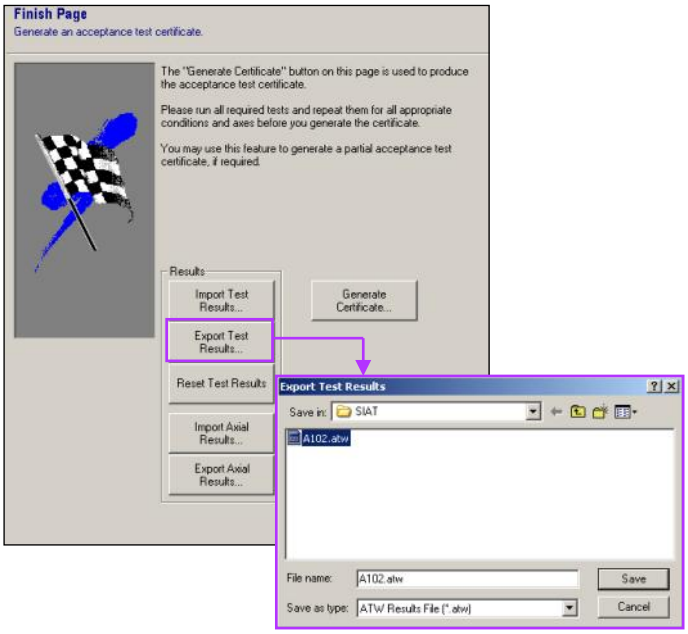
- Verify the result data and acknowledge the test.
- Repeat this test for all ‘Safe Brake Tested’ axes.

- Click ‘Next’ to complete the acceptance test procedure.

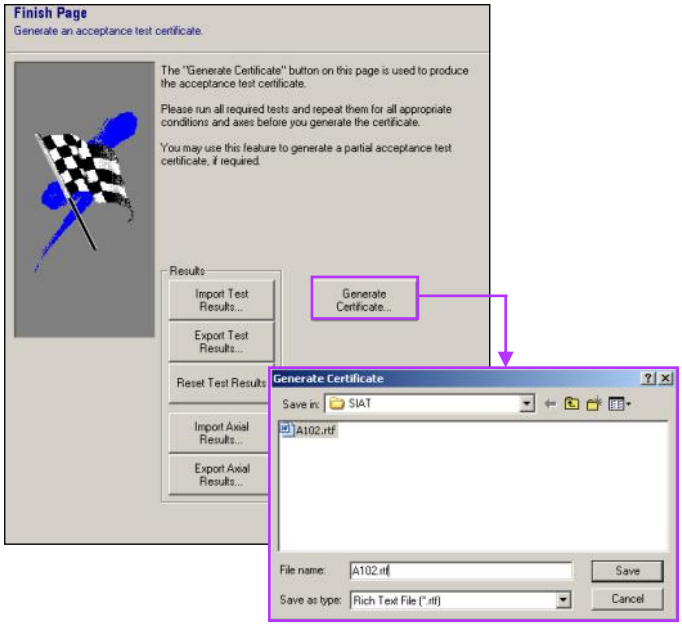
Notes :

Saving the Test Results and Generating the Certificate

With the ‘Finish’ page now selected, export the test results and store the .ATW file safely for future use.



Now ‘Generate Certificate…’



The Certificate file should now be edited to include machine specific information;

- Section 1.1.1 Machine diagram detailing the Safe Axes/ Spindles and the location of all safety related devices (Emergency Stops, Guard Door switches….etc).
- Section 2.1 Function Table description.
- Section 2.3 Description of all Safety Related Devices.
- Section 4.5 Data Archiving.
- Section 4.6 Counter Signatures.

Notes :