SWD®

SAFETY DRIVE & WHEEL DRIVE

Instruction Manual

Version 2.0.x-b – Translated from French

SWD
Safety Wheel Drive

ez-Wheel®
A brand of the **IDEC Corporation Group**

Table of content

Table of figures

⚠ **Please read this document carefully before using the product for the first time.**

1. Introduction

1.1. Who is this manual intended for?

This manual is intended for industrial machine integrators.

Knowledge and understanding of power drive systems is necessary for implementing ***SWD®*** technology.

1.2. Terms of use

The terms used in this manual relate to the technical field of industrial machines and more specifically to fieldbus-controlled drive systems.

For an accurate reading of the manual, a good knowledge of the following reference systems is recommended:

- Machinery Directive (2006/42/CE)
- Safety of machinery – Electrical equipment of machines (EN 60204-1)
- Safety requirements – Adjustable speed electrical power drive systems (EN 61800-5)
- CANopen Protocol (EN 50325 et CiA/DS 301) et CANopen Safety (CiA 304)
- CANopen device profile for drives and motion control (CiA 402)

1.3. Additional resources

The following documents relating to the ***SWD®*** product are available from ez-wheel.com:

- Datasheets of ***SWD®*** products
- General ***SWD® Safety Wheel Drive*** range brochure
- 2D and 3D mechanical drawings for ***SWD®*** products

1.4. Declarations of conformity

SWD® products are developed in accordance with the regulatory requirements for commercialization.

The declarations of conformity for ***SWD® products have been drawn up by IDEC Corporation with the certification authority INERIS*** for the certified safety functions.

1.5. Important information about the manual

⚠	Important information – Read carefully
🖹	Configurable value
i	Additional information

1.6. Disclaimer

The technical information included in this manual is subject to change. No responsibility is accepted for the completeness, up-to-dateness or accuracy of the data and illustrations provided.

Texts and visuals included in this manual are the property of ***APEM SAS,*** *company of* ***IDEC Corporation***.

ez-Wheel and ***SWD Safety Wheel Drive*** are registered trademarks.

Designations may be trademarks and/or copyrights of their respective manufacturers, their use by third parties for their own purposes could conflict with the rights of their owners.

« INSTRUCTION MANUAL »
English Version 2024, translated from French

FIRST EDITION, FEBRUARY 2022

APEM SAS
IDEC Mobility Solutions division
LE MOULIN DE L'ABBAYE - 135 ROUTE DE BORDEAUX
16400 LA COURONNE - FRANCE

2. Safety instructions – Precautions regarding usage of *SWD® products*

⚠	Do not open. Do not expose to a heat source. Do not expose to fire. Do not insert metal pieces in the connectors. In any case, the product cannot be modified without explicit authorization of IDEC. Do not attempt to modify technical performances. Product use must remain within technical performance range specified by IDEC. Improper use and opening the product invalidates the warranty.

3. Description

Warning message applicable to *SWD®* products fitted with the parking brake option.

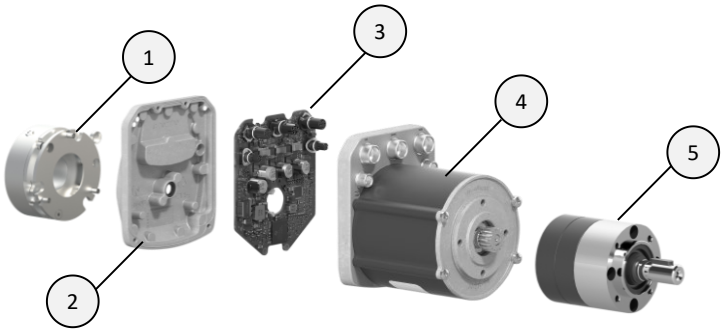
⚠ The surface of the brake is subject to high temperatures approaching 100°C. There is a risk of burn while handling the product after use.

A pictogram "Danger hot surface" is located on the technical label to locate the hazardous area:

3.1. Presentation and main characteristics

SWD® Core

Presentation:



1	Brake option
2	Electronic cover
3	Safety driver
4	BLDC Motor with PM (Permanent Magnet)
5	Planetary gearbox **1, 2 or 3 stage(s)**

Key features:

Motor shaft output	keyed shaft, Ø 14 mm
Typical use	Safety conveying or safety lifting
IP index	IP66 (Electronic cover)
Nominal voltage	24 VDC
Motor	BLDC PM
Nominal power	Up to 250 W (S1)
Transmission	Without, 1, 2 or 3-stages
Option(s)	Parking brake **/B**

Available transmissions:

Reference	Product ID	Stage(s)	Ratio (exact value)	Nominal speed (rpm)	Nominal torque (Nm)
EW2A-000N00x	ezSWDcore.0x/C	0	1	1400	1.7
EW2A-000N04x	ezSWDcore.4x/C	1	4 (3.71)	370	5.0
EW2A-000N14x*	**ezSWDcore.14x/C**	**2**	**14** (13.73)	**100**	**17**
EW2A-000N25x	ezSWDcore.25x/C		25 (25.01)	56	25
EW2A-000N50x	ezSWDcore.50x/C	3	50 (50.89)	28	25
EW2A-000N99x	ezSWDcore.100x/C		100 (99.51)	14	25

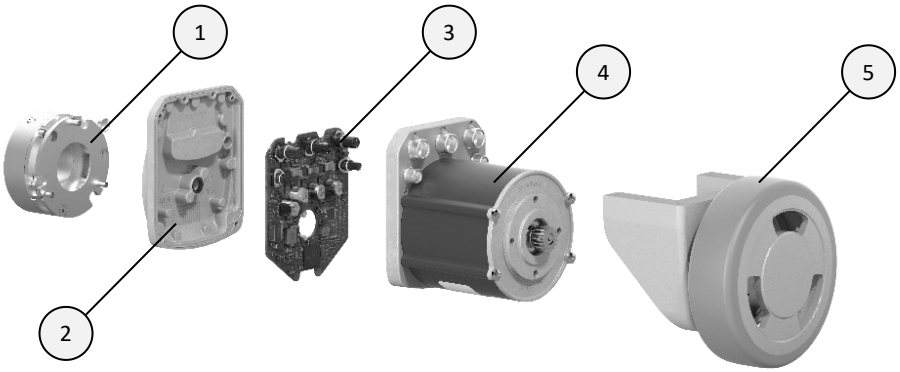
* Standard commercial reference

Key features for standard reference **EW2A-000N14x** / ezSWDcore.14x/C:

Speed	0 to 130 rpm
Nominal performance	17 Nm at 100 rpm
Peak torque	37 Nm
Transmission	Planetary gearbox – 2 stages ratio 14:1 – exact value $(63/17)^2$

SWD® 125

Presentation:



1	Brake option
2	Electronic cover
3	Safety driver
4	BLDC Motor with PM (Permanent Magnet)
5	Ø125 mm wheel, with integrated gearbox

Key features:

Wheel diameter	Ø125 mm
Type of tyre	PU 80 sh. A – Flat profile
Max. vertical load	250 kg
Typical use	Mobile robot, up to 500 kg at $2m.s^{-1}$

IP index	IP66 (Electronic cover)
Motor	BLDC PM
Nominal power	Up to 200 W (S1)
Transmission	1 or 2-stages
Option(s)	Parking brake **/B**

Available transmissions:

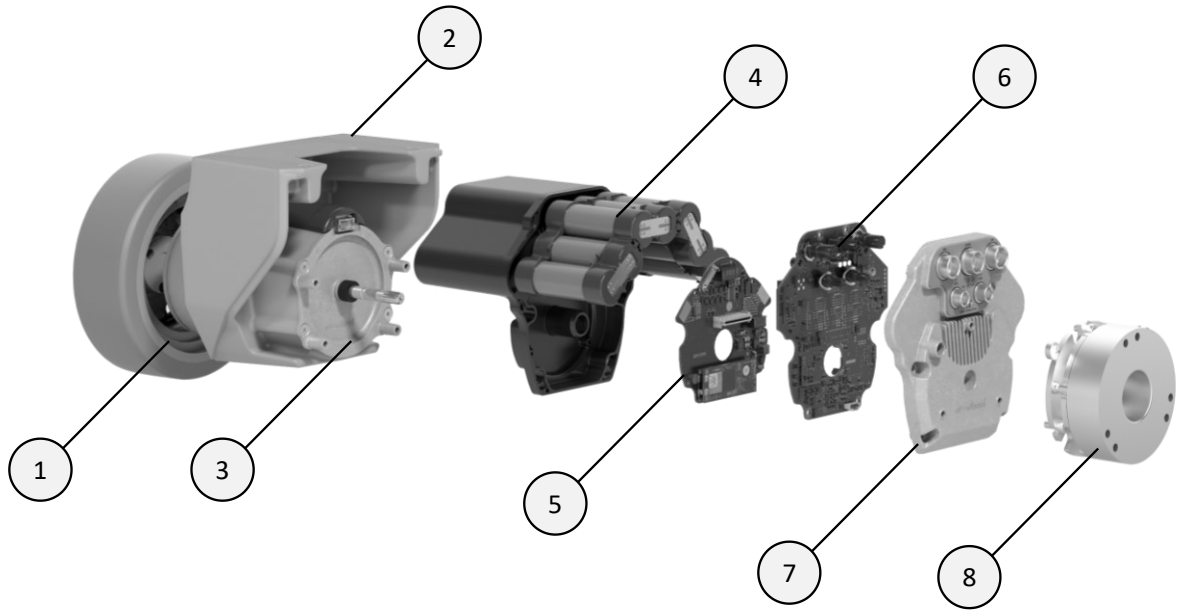
Reference	Product ID	Stage(s)	Ratio (exact value)	Nominal speed (rpm)	Nominal torque (Nm)
EW2A-125HN04x*	ezSWD125IM.4x/C	**1**	**4** (3.65)	**380**	**7.9**
EW2A-125HN14x	ezSWD125IM.14x/C	2	14 (13.53)	100	27

* Standard commercial reference

Key features for standard reference **EW2A-125HN04x** / ezSWD125IM.4x/C:

Speed	0 to 11 km/h
Pushing force	20 daN – Moves 500 kg
Nominal performance	7.9 Nm at 380 rpm
Peak torque	13 Nm
Transmission	Planetary – 1 stage ratio 4:1 – exact value (62/17)

SWD® 150



1	Ø 150 mm wheel, with integrated gearbox
2	Cast iron hub
3	Motor BLDC PM
4	Battery cells
5	BMS (Battery Management System)

6	Safety driver
7	Electronic cover
8	Brake option

Key features:

Wheel diameter	Ø 150 mm
Type of tyre	PU 92 sh. A – Flat profile
Max. vertical load	700 kg
Typical use	Mobile robots, up to 1.500 kg /motor at 1m.s^{-1}
IP index	IP66 (Electronic cover)
Nominal voltage	24 VDC
Motor	BLDC PM
Nominal power	185 W (S1)
Transmission	2-stages
Option(s)	Internal battery 100 Wh Parking brake**/B**

Available transmissions:

Ref.	Product ID	Stages	Ratio (exact value)	Nominal speed (rpm)	Nominal torque (Nm)
EW2A-150Hx14x*	ezSWD150IH.14x/Cx	**2**	**14** (13.73)	**100**	**23**
EW2A-150Hx25x	ezSWD150IH.25x/Cx		25 (25.01)	56	42

* Standard commercial reference

Key features for standard reference **EW2A-150Hx14x** / ezSWD150IH.14x/Cx:

Speed	0 to 3,7 km/h
Pushing force	60 daN – moves 1 500 kg
Nominal performance	23 daN at 100 rpm
Peak torque	45 Nm
Transmission	Planetary gearbox - 2 stages ratio 14:1 exact value (63/17)[2]

3.2. Safety functions

Motor torque disconnection	STO (Safe Torque Off) - up to SIL3/PLe/Cat4
Motor control	SMS (Safe Maximum Speed), SLS (Safe Limited Speed), SLSa[1],SDI - (Safe Direction) up to SIL2/PLd/ Cat3

[1] SLSa, Safe asymmetrical speed, ez-Wheel® specific, not specified by CEI 61800-5-2 norm.

Braking	SBC (Safe Brake Control) up to SIL2/PLd/Cat3, SBU[2] - (Safe Brake Unlock) up to SIL2/PLd/Cat3	
Encoder	30 ppr (motor shaft, before reduction) - SIL2/PLd/Cat3	
Interface	CANopen Safety® Safety Inputs, OSSDs compliant	CANopen safety

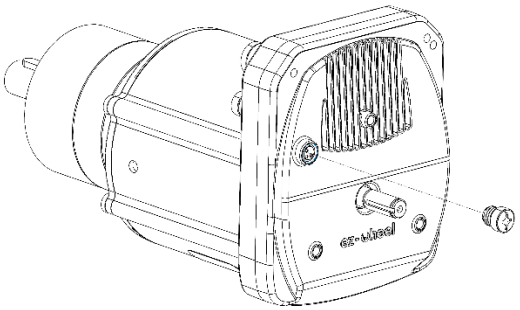
3.3. Terms of use

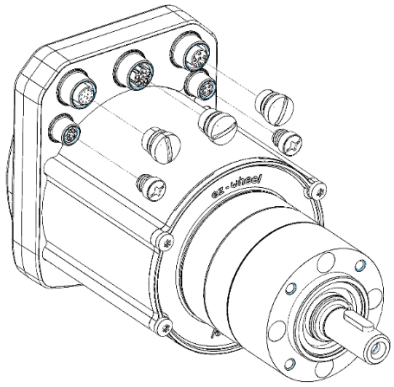
Temperatures	0 to +40°C
IP index	IP66 (Electronic cover)
Maintenance period	5 years[3]

⚠ The product is designed to be integrated into a machine. It is recommended to cover the machine to avoid direct access to the motorization by end-users.

Product sealing

To guarantee ***SWD®*** product life, we recommend using caps on unused connectors. Compliant caps are available on IDEC ez-Wheel product guide.

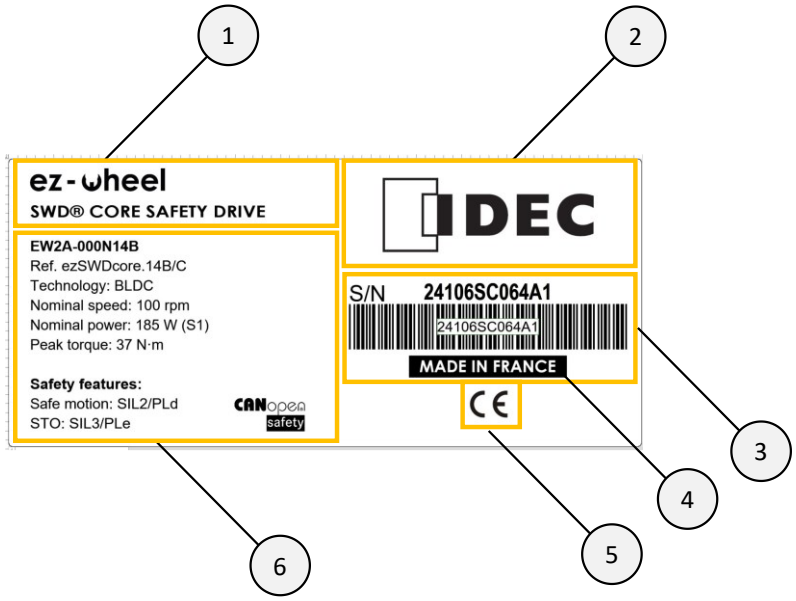




[2] SBU, Safe release of the brake, ez-Wheel® specific, not specified by CEI 61800-5-2 norm.
[3] Reference value, based on standard test condition, may differ according to specific use.

3.4. Product label

Product label example for ***SWD***® Core **EW2A-000N14B**:



1	Product Family and Range Identification
2	Manufacturer and trademark identification
3	Item code
4	Geographical origin of product
5	Regulatory pictogram
6	Commercial reference, Motor performance, Safety features

3.5. Applications

SWD® products are designed for speed-controlled moving applications, where risk assessment justifies the implementation of safe movement supervision.

- Mobile robots
- Pallet shuttle
- Lifting cart
- Conveyors

4. Synoptic

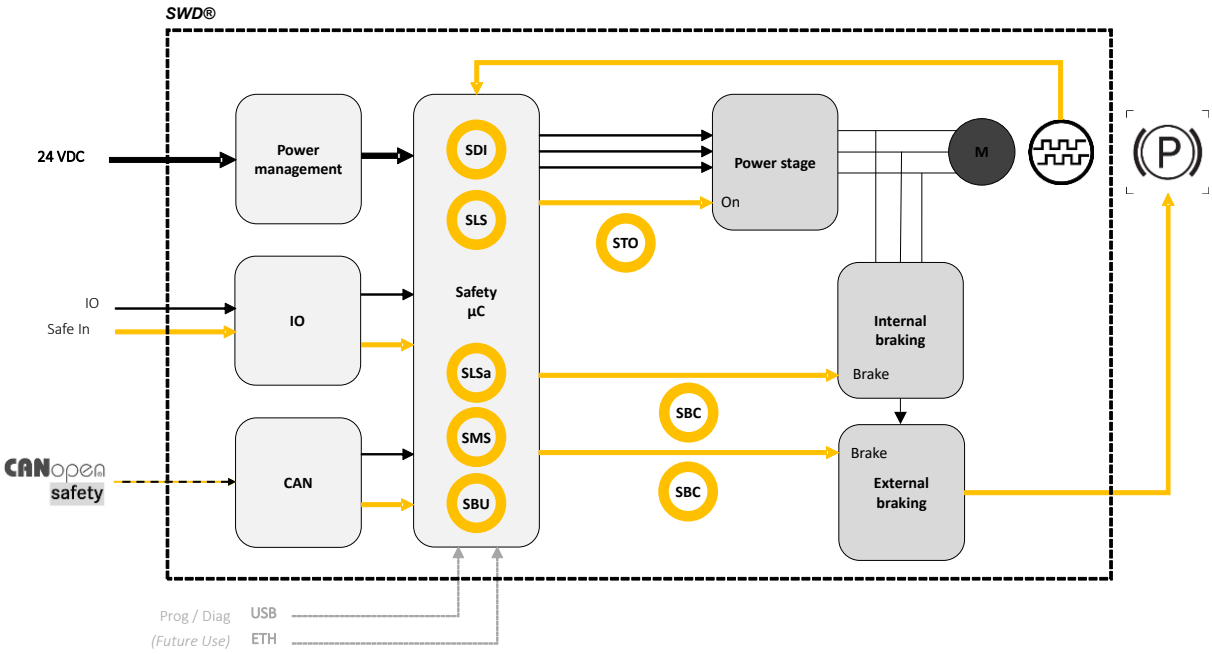


Figure 1 – General product overview

⚠ Functions SBC, SLSa, SMS and SBU are implemented from firmware version 'Firmware 2.0.1'

M	Motor et gearbox
	Safety encoder
P	Safety external brake

5. Interfaces

5.1. Overview

Mechanical assemblies' identification

SWD® Core

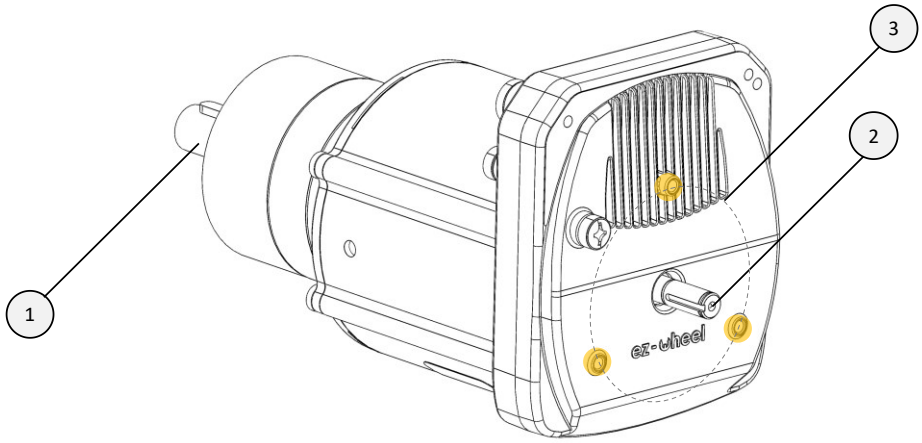


*Figure 2 – Overview of mechanical interfaces– **SWD® Core***

1	Gearbox mounting flange
2	Brake mounting shaft
3	Brake mounting pins

SWD® 125

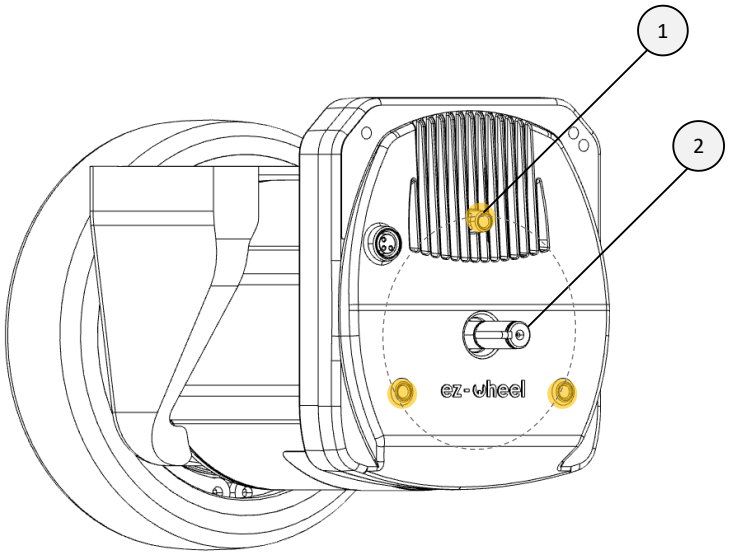


*Figure 3 – Overview of mechanical interfaces – **SWD® 125***

1	Brake mounting pins
2	Brake mounting shaft

SWD® 150

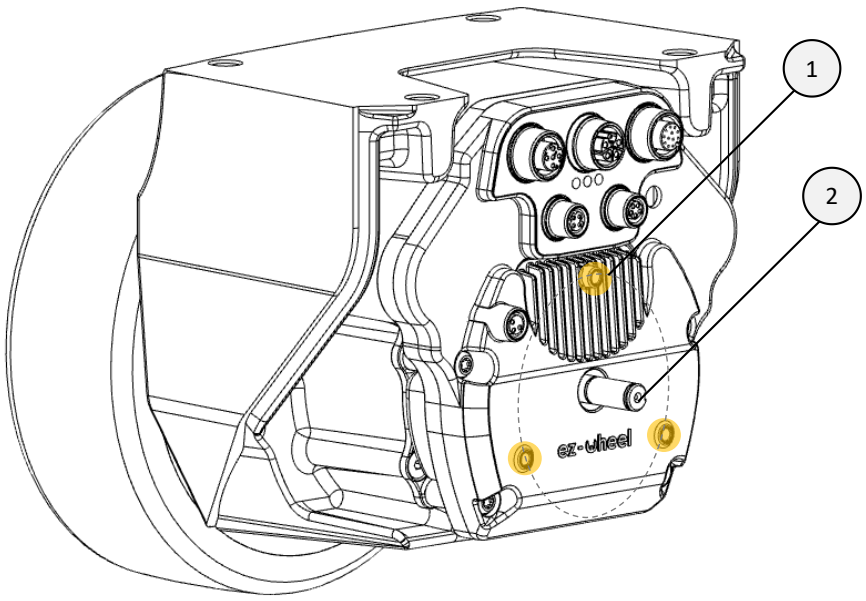


*Figure 4 – Overview of mechanical interfaces – **SWD® 150***

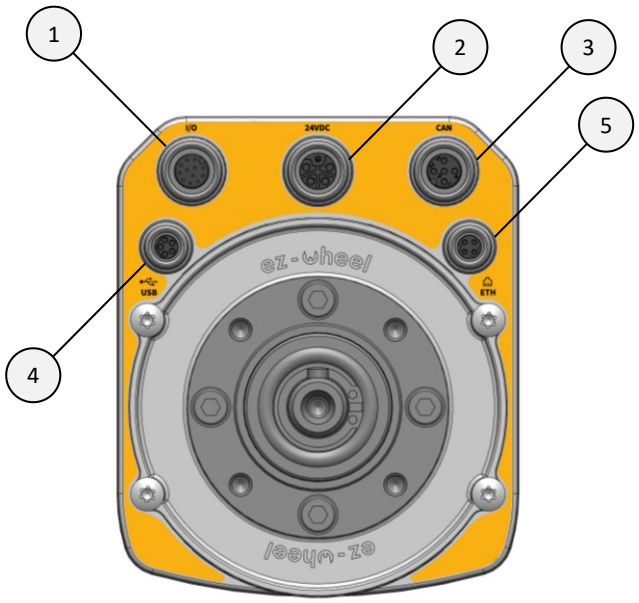
1	Brake mounting pins
2	Brake mounting shaft

Connectors and LEDs identification

SWD® Core

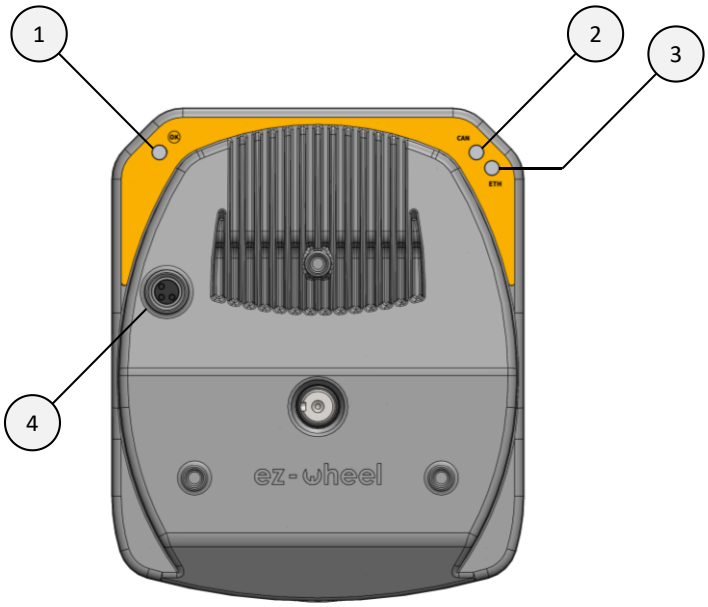
And,

SWD® 125



1	Connector I/O
2	Connector 24VDC
3	Connector CAN
4	Connector USB
5	Connector ETH *(Reserved for future use)*

Figure 5 – Connectors identification



1	**LED Status**
2	LED CAN
3	LED ETH *(Reserved for future use)*
4	Connector brake

Figure 6 – Connector brake and LEDs identification

SWD® 150

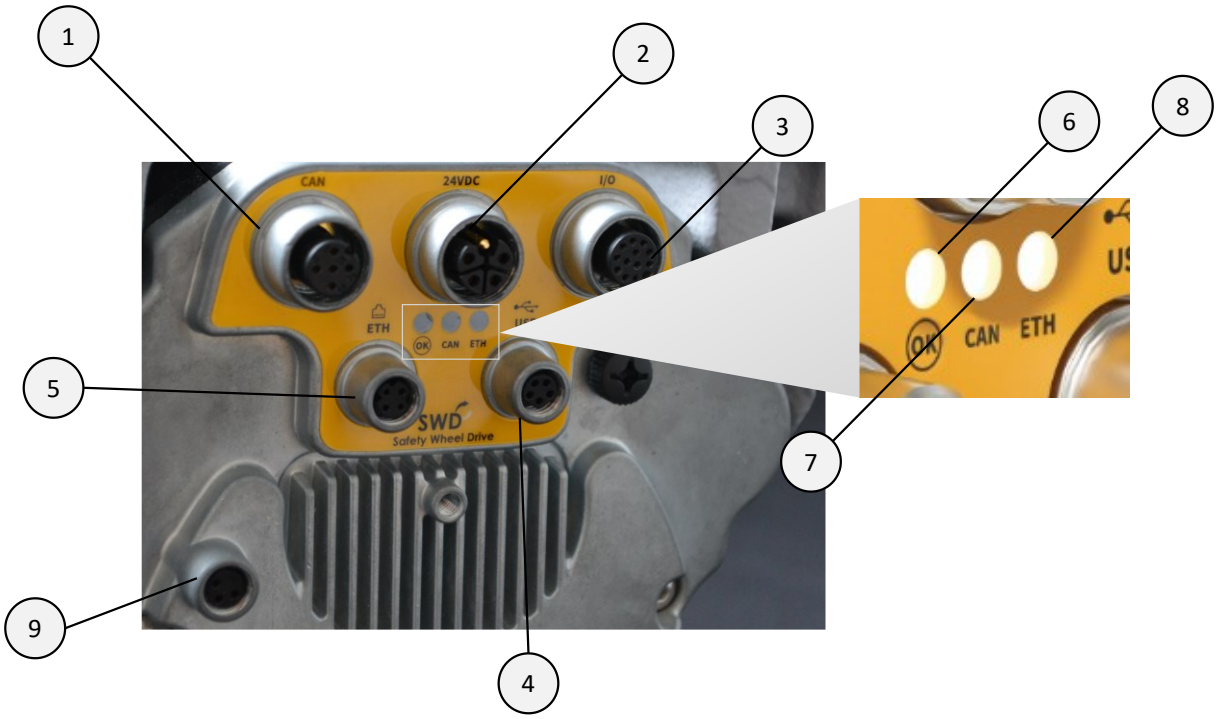


*Figure 7 – Connectors and LEDs identification – **SWD® 150***

1	Connector CAN
2	Connector 24VDC
3	Connector I/O
4	Connector USB
5	Connector ETH *(Reserved for future use)*
6	LED Status
7	LED CAN
8	LED ETH *(Reserved for future use)*
9	Connector brake

5.2. Connectors

The use of shielded cables is strongly recommended, as well as a total length of less than 30 meters for the CAN bus.

Connector layout plan

Face avant : Face arrière :

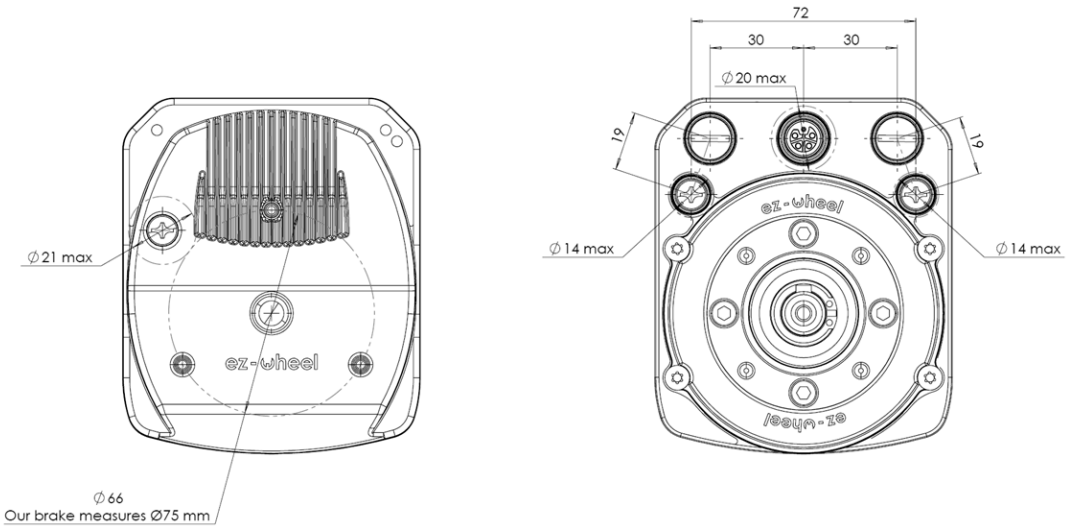
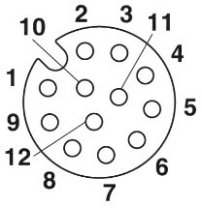


Figure 8 – Connector layout plan

I/O Connector

The I/O connector is M12, 12-pin, A-coded, female. It groups together the safety (INSafe) and product start-up (STO) inputs. It can also be used as a CAN interface (in addition to the dedicated CAN connector), and as a 24V power offset for third-party peripherals.

Pin #	Designation
1	CAN H
2	INSafe_4
3	CAN L
4	24 VDC output power supply (2A) – if activated =
5	STO_1: (0V: Safe State / 24 VDC: Drive enable)
6	GND output power supply (2A)
7	INSafe_1
8	STO_2: (0V: Safe State / 24 VDC: Drive enable)
9	INSafe_3
10	INSafe_2
11	GND IN (From external device)
12	ON: Pull to GND to turn it ON *(Reserved for future use)*



⚠ This connector cannot be used as a power supplier for the ***SWD®***

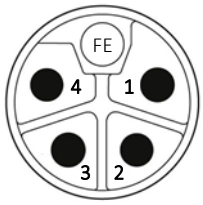
The following cables can be used to interface with the I/O connector:

- Compatible cables available in the IDEC ez-Wheel product guide

24 VDC Connector

The 24 VDC power supply connector is M12 Power, 5-pin, L-coded (4+FE), female. We recommend placing the power source as close as possible to the ***SWD®***, to limit power dissipation in the cables. A maximum length of five meters between the power source and the ***SWD®*** is recommended.

Pin #	Designation
1	+ Power supply 24 VDC (16A)
2	+ Power supply 24 VDC (16A)
3	GND Power supply (16A)
4	GND Power supply (16A)
FE	Mechanical mass connected to chassis



The following cables can be used to interface with the 24 VDC connector:

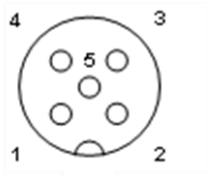
- Compatible cables available in the IDEC ez-Wheel product guide
- Phoenix Contact SAC-5P-M12MSL/ 1,5-280 FE SH – 1414884, 1.5m
- Phoenix Contact SAC-5P-M12MRL/ 1,5-280 FE SH - 1414851, 1.5m *(Only for **SWD®**-150)*

CAN Connector

The CAN connector is M12, 5-pole, A-coded, female. It can be used as a CANopen interface and as a 24V power supply for external devices, such as additional sensors.

Implementation complies with CiA 303-1 standard for standardized CAN interfaces.

Pin #	Designation
1	Mechanical mass
2	24 VDC output power supply (4A) – if activated = 🖹
3	GND output power supply (4A)
4	CAN H
5	CAN L



⚠ This connector cannot be used as a power supplier for the ***SWD®***

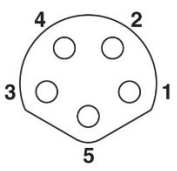
The following cables can be used to interface with the CAN connector:

- Compatible cables available in the IDEC ez-Wheel product guide
- Phoenix Contact SAC-5P-MR/ 2,0-923 CAN SCO – 1419044, 2m
- TE Connectivity 2273100, 1.5m

USB Connector

The USB connector is M8, 5-pin, B-coded, female. It is only used for updating and diagnosing the product. It must only be used with the explicit agreement of IDEC for safety reasons, otherwise all warranties will be voided.

Pin #	Designation
1	V_USB
2	USB D+
3	USB D-
4	GND USB
5	USB Id (NC)



The following cables can be used to interface with the USB connector:

- Compatible cables available in the IDEC ez-Wheel product guide
- Phoenix Contact SAC-5P-M 8MSB/ 1,5-115 – 1404461, 1.5m
- Phoenix Contact SAC-5P-M 8MS/ 2,0-920 – 1575712, 2.0m
- Phoenix Contact SAC-5P-M 8MR/920/... – 1575903 'Made to order.'

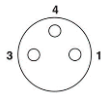
ETH Connector

⚠ *ETH connector is reserved for future use, it must not be connected.*

External brake connector

The external brake connector is M8, 3-pole, A-coded, female. It is used to connect an electromechanical brake, actuated by the SBC (Safe Brake Control) [4] output.

Pin #	Designation
1	SBC Command
3	GND_SBC
4	V_SBC *(BRAKE_LOCK_CHECK)*



- Pin1 is used to control the brake. If power is supplied (24VDC), the bake is released. If no power is supplied (0VDC), the brake is closed and holds motor shaft. The current consumption to keep it open is around 0,5A. The maximum current that can be provided by the SWD® is 4A, if the current exceeds this value an emergency message (EMCY) is sent with error code: '8011_h: External brake overcurrent error'.
- Pin3 is connected to electrical ground (GND).
- Pin4 is used to check that the voltage is the same on Pin1 and Pin4. The check occurs cyclically on PDS state 'OPERATION ENABLED' (when '2660_h Brake present' is set to True).

Consult your reseller for a brake compatible with your product.

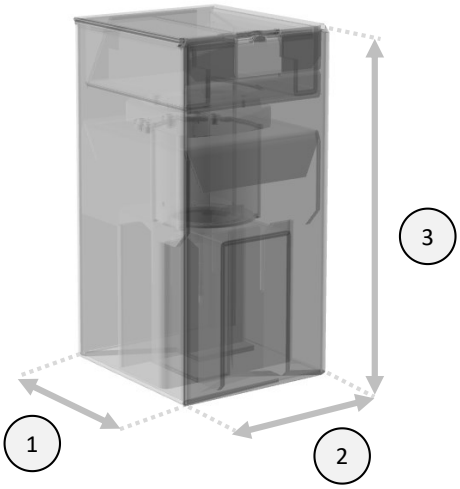
[4] Implemented from firmware version 'Firmware (2.0.x)'

6. Information and mechanical assembly

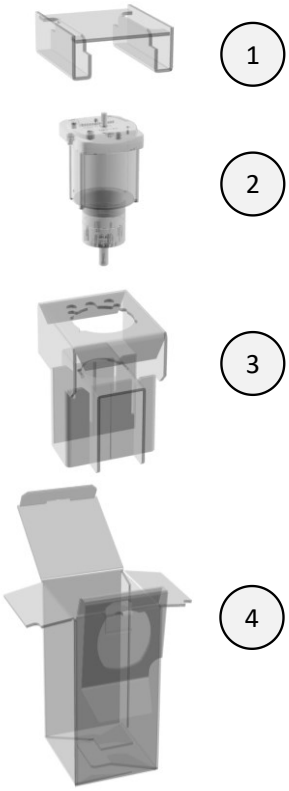
6.1. Dimensions, packaging, and pack contents

***SWD®* Core**

1	Depth: 170 mm
2	Width: 150 mm
3	Height: 306 mm



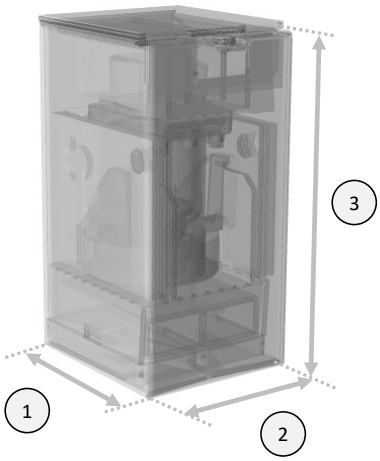
1	Upper wedge
2	Product ***SWD® Core***
3	Lower wedge
4	Conditioning



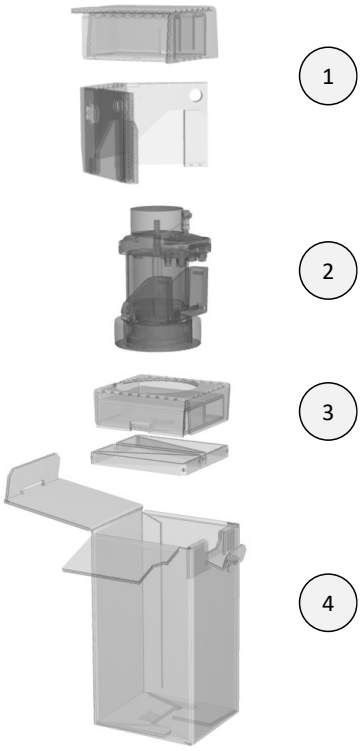
⚠ For engine-only logistics operations, use the original packaging.

SWD® 125

1	Depth: 170 mm
2	Width: 150 mm
3	Height: 310 mm



1	Upper wedge
2	Product ***SWD® 125***
3	Lower wedge
4	Conditioning



SWD® 150

Definition ongoing.

6.2. Overall dimensions and weight[5]

SWD® Core

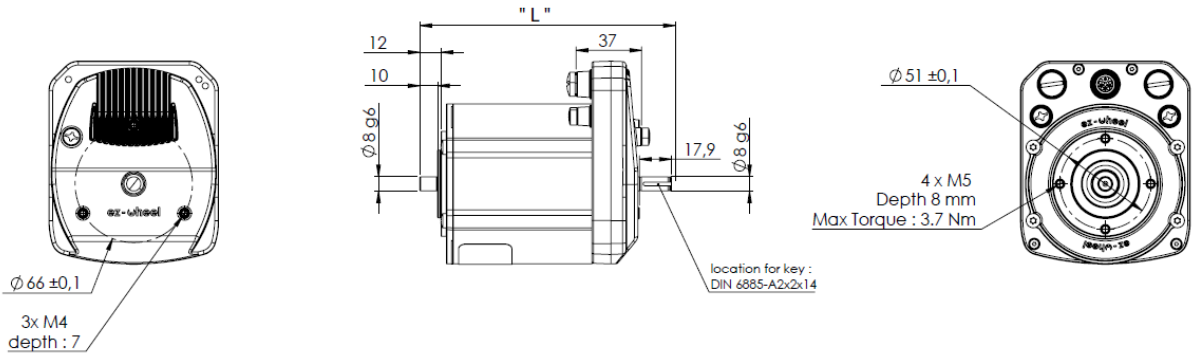


*Figure 9 – Overall dimensions – **SWD® Core** without gearbox*

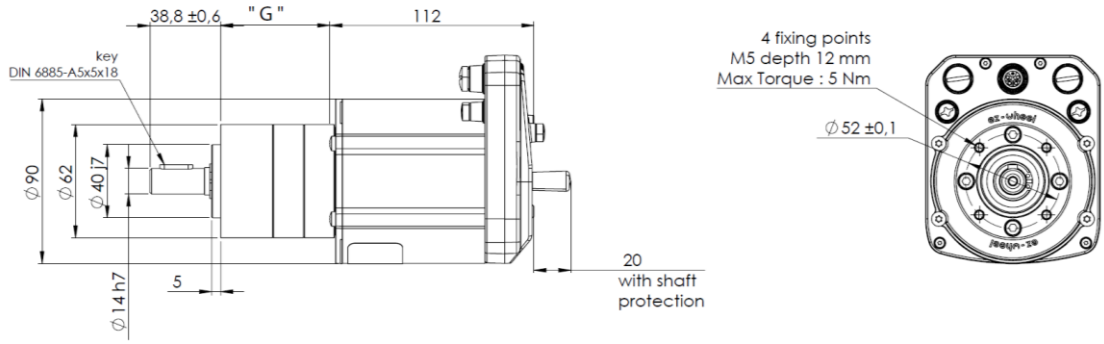


*Figure 10 – Overall dimensions – **SWD® Core** with gearbox(es)*

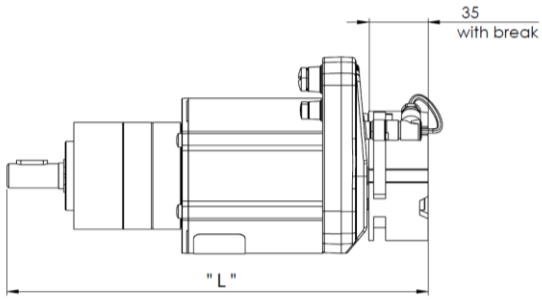


*Figure 11 – Overall dimensions - **SWD® Core** – with external brake*

[5] Indicative values +/- 10%

Gearbox Dim "G" ± 0.3 (mm)			With external brake	Dim "L" ± 2 (mm)	Weight ± 10% (kg)
1-stage	2-stages	3-stages			
-	-	-	×	144	2.7
42.9	-	-	×	214	3.3
42.9	-	-	✓	229	4.1
-	59.8	-	×	231	3.6
-	59.8	-	✓	246	4.4
-	-	76.8	×	248	3.9
-	-	76.8	✓	263	4.7

SWD® 125

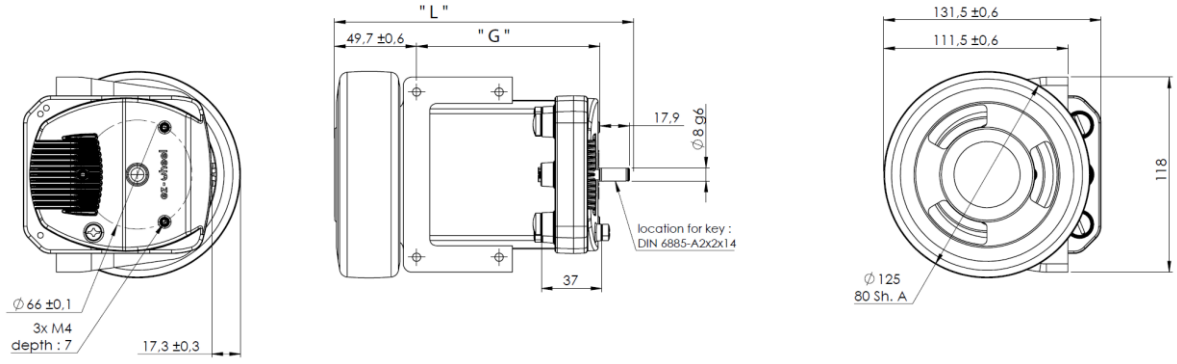


*Figure 12 – Overall dimensions – '**SWD® 125** 1-stage'*

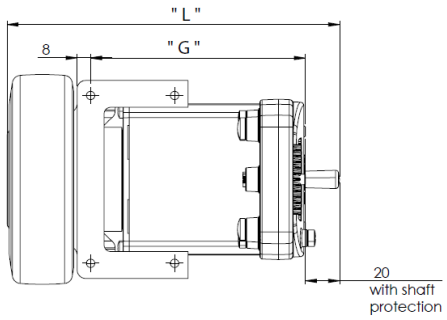


*Figure 13 – Overall dimensions – '**SWD® 125** 2-stages'*

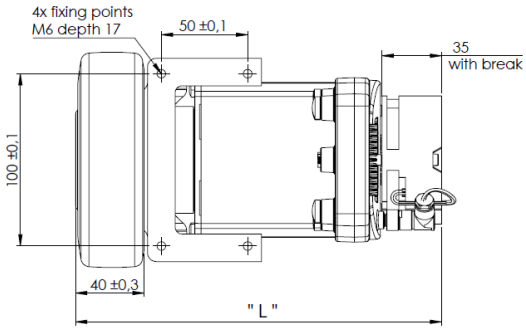


*Figure 14 – Overall dimensions – '**SWD® 125** 2-stages and external brake*

Gearbox Dim "G" ± 0.3 (mm)		With external brake	Dim "L" ± 2 (mm)	Weight ± 10% (kg)
1-stage	2-stages			
111	-	×	181	6.2
111	-	✓	196	7
-	128	×	198	6.5
-	128	✓	213	7.3

SWD® 150

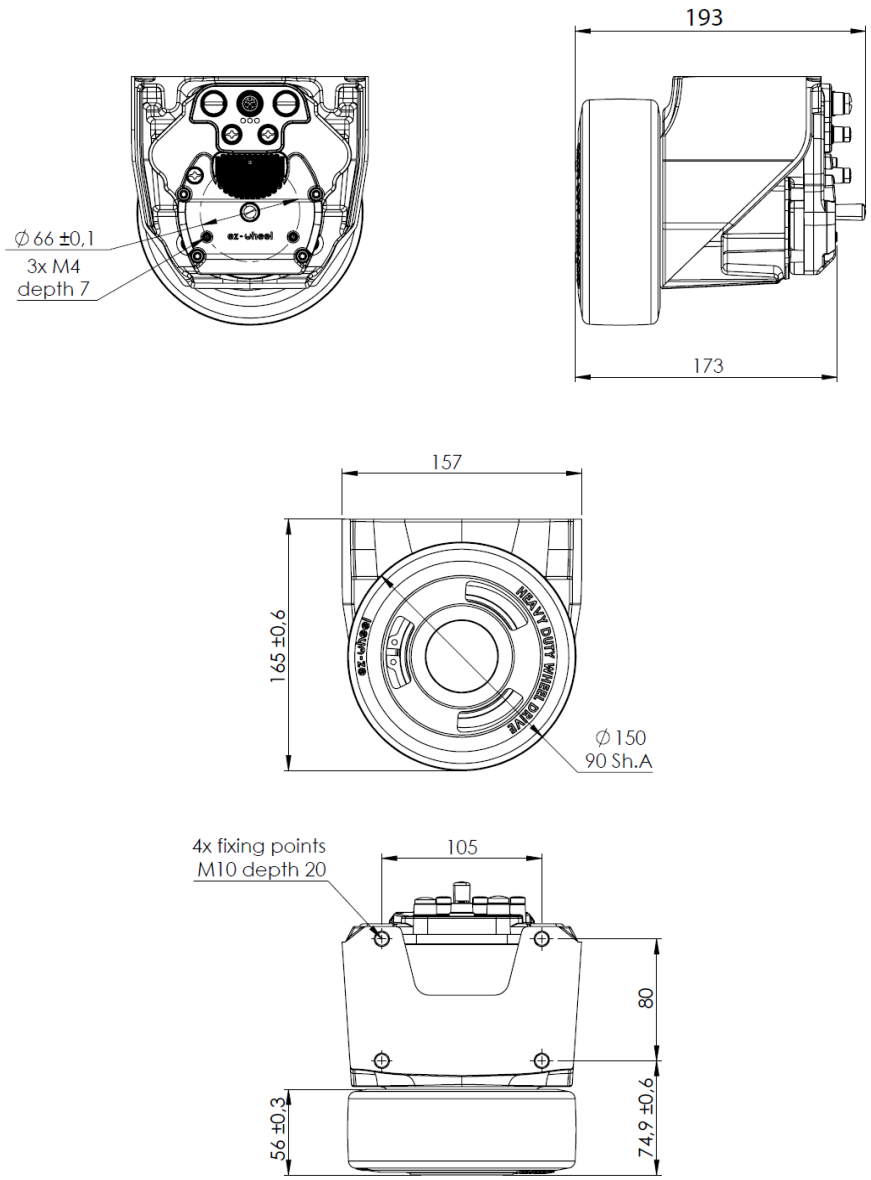


*Figure 15 – Overall dimensions – '**SWD® 150 2-stages**'*

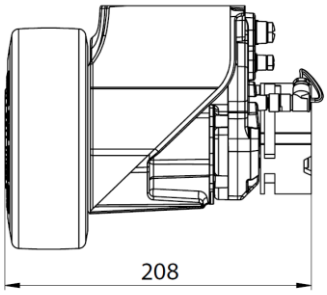


*Figure 16 – Overall dimensions – '**SWD®** 150 2-stages' with external brake*

***SWD® 150* versions**	**Weight ± 10% (kg)**
SWD® 150 with gearbox 2-stages	10.5
SWD® 150 with gearbox 2-stages and external brake	11.3

6.3. Mechanical assembly drawing

***SWD®* Core**

SWD® Core naked - Mounting recommendation (without gearbox)

For any specific mounting on the motor shaft, please contact us.

- **NEVER** tap or knock on the axle!
- **NEVER** exert forces more than the values indicated in paragraph 6.4 Restrictions on use.
  - CAUTION: fitting helical shafts may result in additional axial forces!
- **NEVER** make any mechanical changes to the shaft or bearing.
- **NEVER** tighten screws to the stop (blind holes).
- **NEVER** exceed the maximum tightening torque.
- The axial position of the motor shaft may vary slightly during operation (see *plan below).

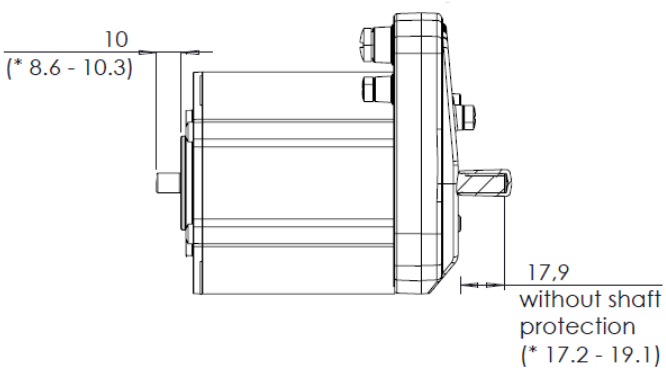


*Figure 17 – Motor shaft tolerance - **SWD®** Core (naked)*

Output characteristics

Max. radial load *(median of output shaft)*	330 N
Max. axial load *(in operation)*	220 N
Max. axial force on shaft when mounted	Consult your reseller

Depending on how the motor shaft is mounted, special tools may be required.

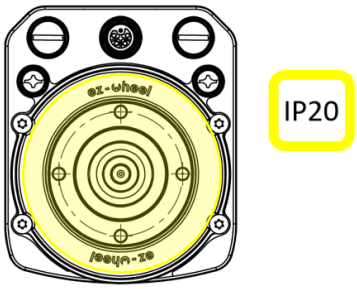
Consult your reseller for more information.

Special cases:
Please consult us for the maximum permissible loads for cases not defined above.

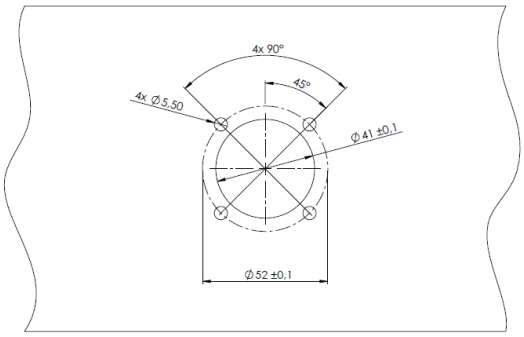
For example:

- Radial load not centred
- Combined axial and radial load

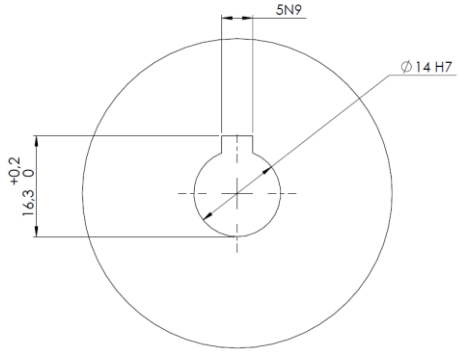
⚠ The front of the motor has an **IP20** protection rating



SWD® Core - Machine layout recommendation



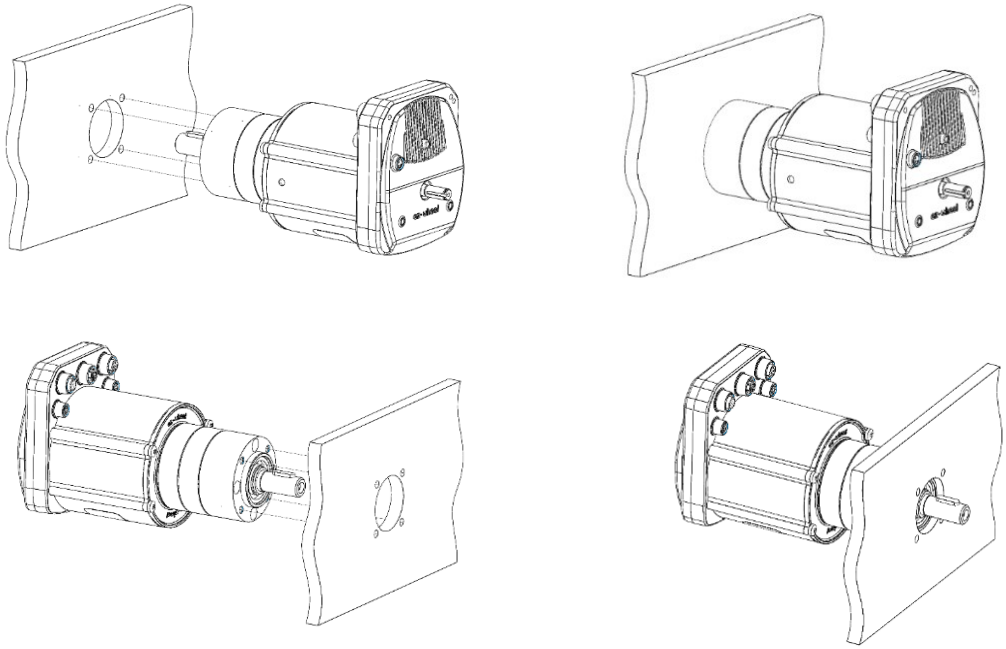
SWD® Core interface



Motor transmission interface

Present the ***SWD® Core*** product on the machine interface until it stops on the flat surface. Using 4 M5 screws (not supplied), tighten the product on the machine interface.

NEVER use a hammer to set up the product!



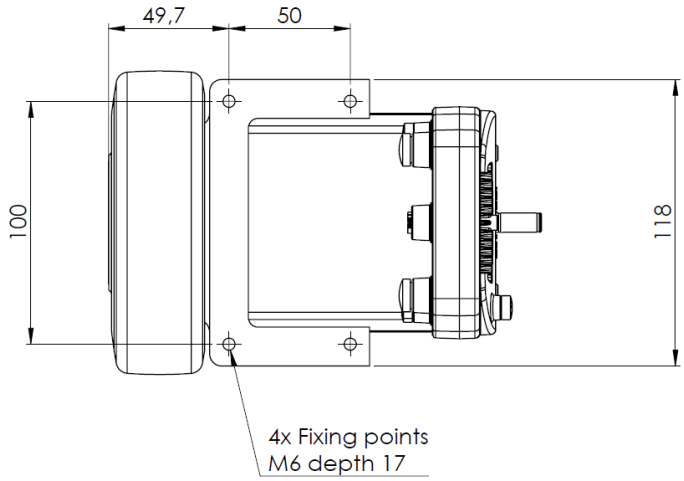
Installation requirements:

- Tightening torque for 4 - M5 screws (not supplied): 5 Nm.
- Length of thread engaged in the product: 8 to 10 mm.
- Use of thread lock and/or lock washers (not supplied).

For sealing higher than IP53, seal the shaft bearing by applying a gasket between the machine interface (x4) and the ***SWD® Core*** product (***SWD® Core*** side) and between the customer interface and the customer application (customer side). As this sealing is different for each application, we leave it up to each integrator to implement the solutions required for his application.

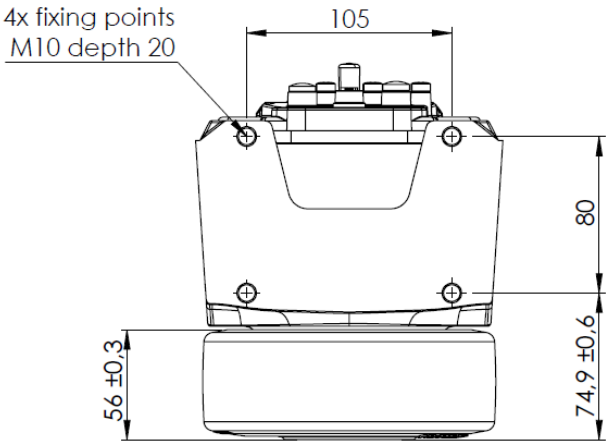
SWD® 125

SWD® 125 - Machine layout recommendation



SWD® 150

SWD® 150 - Machine layout recommendation



7. *SWD® Status indicator*

7.1. *SWD®* indicator display mode

The LEDs lights up according to the display states standardized in CiA 303-3.

LED state	Description
LED on	LED is on
-	LED is off
LED scintillating	LED blinks at 10 Hz (50 ms on / 50 ms off)
LED blinking	LED blinks at 2,5 Hz (200 ms on / 200 ms off)
LED single flash	LED flashes 1 time, followed by a long pause (200 ms on / 1000 ms off)
LED double flash	LED flashes 2 times, followed by a long pause (200 ms on / 200 ms off | 200 ms on / 1000 ms off)
LED triple flash	LED flashes 3 times, followed by a long pause (200 ms on / 200 ms off | 200 ms on / 200 ms off | 200 ms on / 1000 ms off)
LED quadruple flash	LED flashes 4 times, followed by a long pause (200 ms on / 200 ms off | 200 ms on / 200 ms off | 200 ms on / 200 ms off | 200 ms on / 1000 ms off)

During ***SWD®*** firmware updates, CAN LED is disabled.

7.2. Status LED display

SWD® Status LED is a two-color red or green LED, which indicates the status given by the CiA 402 standard status machine:

CiA 402 State		LED green	LED red
Not Ready to switch on	(On going)	LED flashes 1 time	-
Not Ready to switch on	(Initialisation error)	-	LED flashes 1 time
Switch on disabled		LED flashes 2 times	-
Ready to switch on		LED flashes 3 times	-
Switched on		LED blinks	-
Operation enabled		LED on	-
Operation enabled	(STO active)	-	LED on
Quick stop active		LED on	LED flashes 2 times
Fault reaction active		LED scintillating	LED scintillating
Fault		-	LED blinking

During ***SWD®*** firmware updates, Status LED is orange fixed (green + red), and red if the upload has failed[6].

[6] Only for ***SWD®*** products manufactured since October 2023

7.3. CAN LED display

The CAN bus status LED (CAN LED) is a two-color red or green LED, which indicates the operational status of the ***SWD®*** CANopen bus:

CANopen state		LED green	LED red
Bus Off		-	-
Initialisation	(Ongoing)	-	-
Initialisation	(Initialisation error)	-	LED blinking
Pre-Operational		LED blinking	-
Operational		LED on	-
Stopped		LED single flash	-

8. Power supply

8.1. 32A Power supply

A two times 16 amps power supply (32A) is possible via the 24 VDC connector. This transmits the power required to operate the ***SWD®***.

Various causes of power supply errors are possible. For instance, if the voltage is too high or too low. Or due to an excessive current.

Warning thresholds must be exceeded for a certain period before they are triggered (timeout). Error thresholds, on the other hand, are triggered immediately, followed by an STO command.

According to the configuration of the 'error behaviour' object (1029h), an error changes the NMT state of the node, unlike an alert, which does not.

Thresholds values are as follow:

Type	Level	Limits	Timeout	EMCY message
Under-voltage	Alert	16 Volts	1000 ms	✓
	Error	14 Volts	-	✓
Over-voltage	Alert	32 Volts	1000 ms	✓
	Error	34 Volts	-	✓
Over-current	Alert	25 Amps	1000 ms	×
	Error	30 Amps	-	×

Some of these errors send an emergency message on the bus of type EMCY, as described in the table above. When the error level is reached, an STO is activated, and the NMT state can be assigned according to the configuration of the 'error behaviour' object (1029h), by default, the motor switches to pre-operational state.

8.2. 2A and 4A Power supply

A 24 VDC power supply on CAN connector of 4A and on I/O connector of 2A, are configurable. This can be used to supply power to third-party peripherals, via a single cable between the peripheral and the ***SWD®***.

The 'control_ext_can_alim' (2400 01_h) and 'control_ext_canio_alim' (2400 02_h) objects are used to configure the 24V power supply on these connectors.

2400: 01-02_h: control_ext	Deactivated	Activated
Value	**0**	**1**

It is possible that an error blocks the correct operation of the power supply, for example a current requested too high. Objects ‘status_ext_can_alim’ (2401 01_h) and ‘status_ext_canio_alim’ (2401 02_h) are used to find out the status of the power supply on these connectors.

2401: 01-02_h: status_ext	Deactivated	Activated
Value	**0**	**1**

The object returns '0' if an error has occurred, otherwise '1' if the power supply is working properly.

Index	Sub-index	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2400h	01h	control_ext_can_alim	U8	0h	RW	-	-	0	255	1
2400h	02h	control_ext_can_alim	U8	1h	RW	-	-	0	255	1
2401h	01h	status_ext_can_alim	U8	0h	RW	-	-	0	255	1
2401h	02h	status_ext_canio_alim	U8	0h	RW	-	-	0	255	1

- By default, the power supply on the CAN IO connector is activated at start-up.
- If the configuration is saved, it will be applied after a reboot, and the power supply configuration will be activated on this/these connector(s).

9. CAN bus and CANopen protocol

9.1. Bus specification

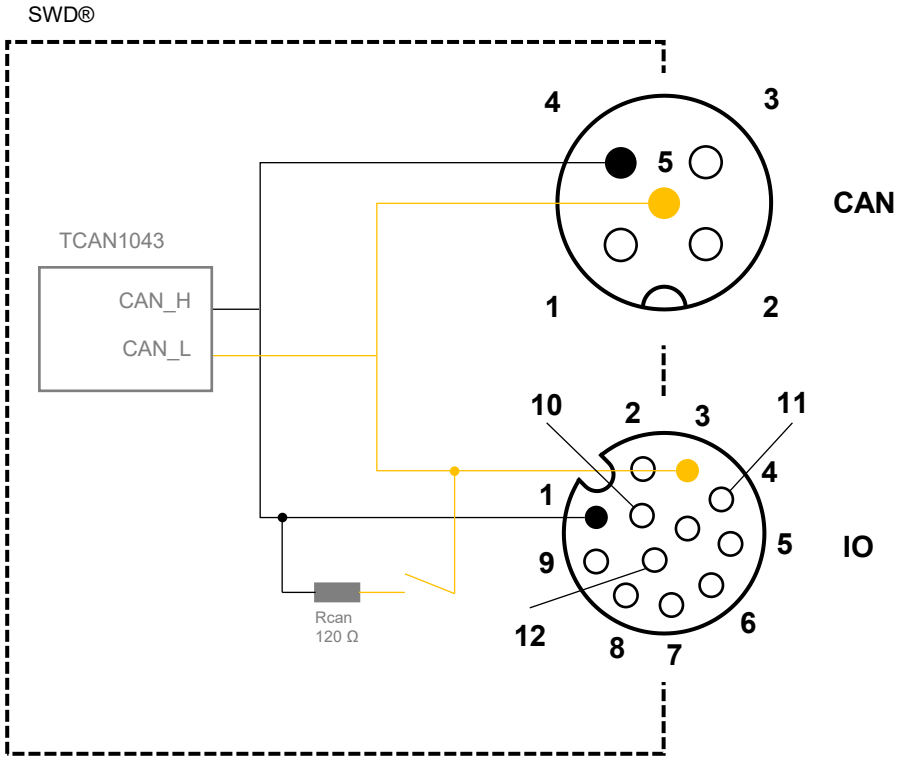


Figure 18 – CAN bus internal wiring diagram

9.2. Bitrate of the CAN node (Baudrate)

SWD® default baud rate is 1,000,000 bps. This value is configurable.

Baud rate configuration can be achieved using two different methods (in NMT Pre-Op mode):

- With ***SWD®*** dictionary object 'Configure Bit Timing Parameters', 2100 :00_h
- With LSS[7] protocol (Layer Settings Services)

Configuration with ***SWD®*** object dictionary is performed in 3 steps:

- Write the corresponding value to the selected Baudrate to the 'Configure Bit Timing Parameters' dictionary object 2100 :00_h
- Save configuration of communication parameters (cf. 9.10)
- Load new configuration using NMT command 'Reset communication'

When saved, ***SWD®*** CANopen identifier is persistent to power loss, and will be used at next start-up.

[7] Implemented from software version 'Firmware' (1.2.x)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2100_h	0	Configure Bit Timing Parameters	U8	0_h	RW	-	-	0	4	1

	Value	Baudrate (bps)
2100_h: Configure Bit Timing Parameters	**0**	**1.000.000**
	1	**800.000**
	2	**500.000**
	3	**250.000**
	4	**125.000**

9.3. CAN node identifier (Node-ID)

The CANopen node identifier identifies a device on the bus. Each node identifier is associated with default CAN message identifiers.

Default ***SWD®*** identifier is 10_h, this value is configurable.

Node-ID configuration can be achieved using two different methods (in NMT Pre-Op mode):

- With ***SWD®*** dictionary object 'Node ID', 2101 :00_h
- With LSS[8] protocol (Layer Settings Services)

Configuration with ***SWD®*** object dictionary is performed in 3 steps:

- Write the new node-ID value in object dictionary 'Node ID', 2101 :00_h
- Save configuration of communication parameters (cf. 9.10)
- Load new configuration using NMT command 'Reset communication'

When saved, ***SWD®*** CANopen identifier is persistent to power loss, and will be used at next start-up.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2101_h	0	Node Id	U8	10_h	RW	-	-	1	127	1

⚠ Changing the ***SWD®*** identifier has no impact on message configuration. When configuring the ***SWD®***, the user must ensure that the communication parameters (PDO, SDO, etc.) are reconfigured.

⚠ Only the Heartbeat and SDO server 1 identifiers are updated automatically when the Node Id is reconfigured.

9.4. Termination resistor

When the drive is located at the end of the CAN bus, a termination resistor is required. The value of this resistor must be determined in relation to the application's bus topology.

The termination resistor can be set up in 2 simple ways:

- Add a terminating resistor to one of the CAN bus connectors between CAN H and CAN L.
- By activating ***SWD®*** internal termination resistor

[8] Implemented from software version 'Firmware' (1.2.x)

SWD® includes an internal termination resistor whose activation is configurable by software.

⚠ *The internal resistor is activated in the default configuration.*
⚠ *During motor drive start-up or reset, the resistor is deactivated until the configuration is applied.*

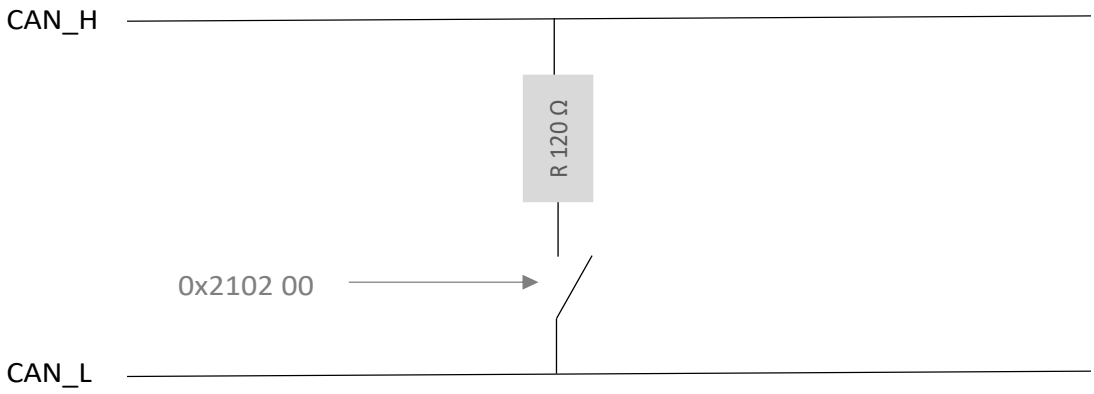


Figure 19 – Internal termination resistor activation

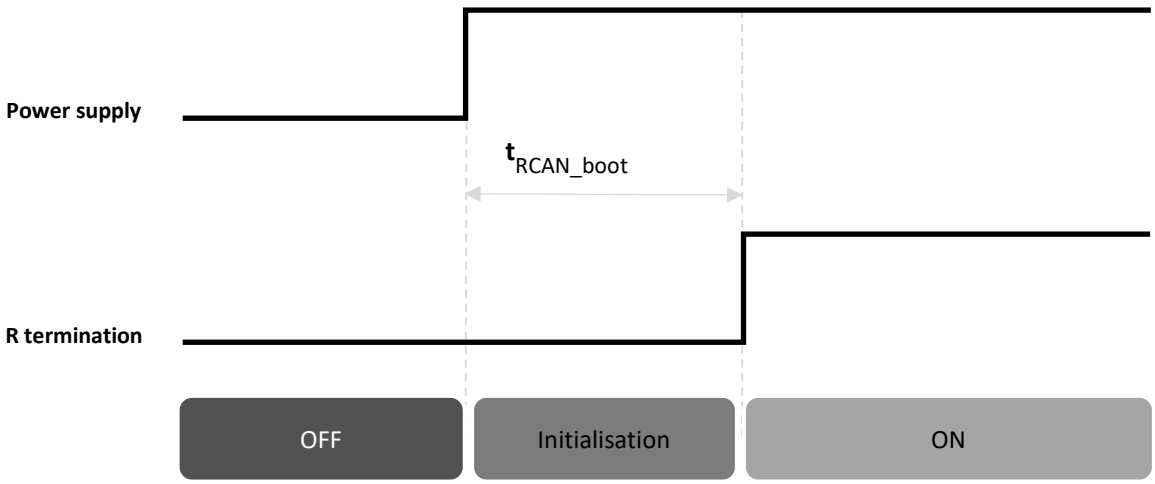


Figure 20 – CAN bus initialisation

		Min	Max	Value	Unit
R_{can}	Internal termination resistor			120	Ω
t_{RCAN_boot}	Time for setting up the internal resistance configuration at start-up		<500		ms

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2102_h	00_h	R Termination	BOOL	01_h	RW	NO	-	0	1	1

2102_h: R Termination	Deactivated	Activated
Value	**0**	**1**

9.5. *SWD®* Identity object

SWD® identification data is described in object 1018_h in this format:

Index	Sub-index	Name	Value	Data type
1018_h	00_h	Number of entries	04_h	Unsigned8
	01_h	Vendor-ID	$0000\ 0515_h$	Unsigned32
	02_h	Product code	-[9]	Unsigned32
	03_h	Revision number	-[10]	Unsigned32
	04_h	Serial number	-[11]	Unsigned32

9.6. NMT protocol (Network ManagemenT) and state machine

A CANopen node respects a state machine, which corresponds to its start-up and operating states. State transitions can be performed automatically, or in response to a request from the CANopen Master node.

The NMT (Network Management) protocol allows this state machine to react, and to get NMT status of a CANopen node:

[9] Specific ID according to the ***SWD®*** device used (e.g. ezSWDcore.14/C (EW2A-000N14A) has ID 3). Refer to the product itself.

[10] Bit 31-16 is the major revision number. Bit 15-0 the minor revision number. Cf. release notes.

[11] Specific serial number, refer to the product itself. Caution encoded format differs from the sticker.

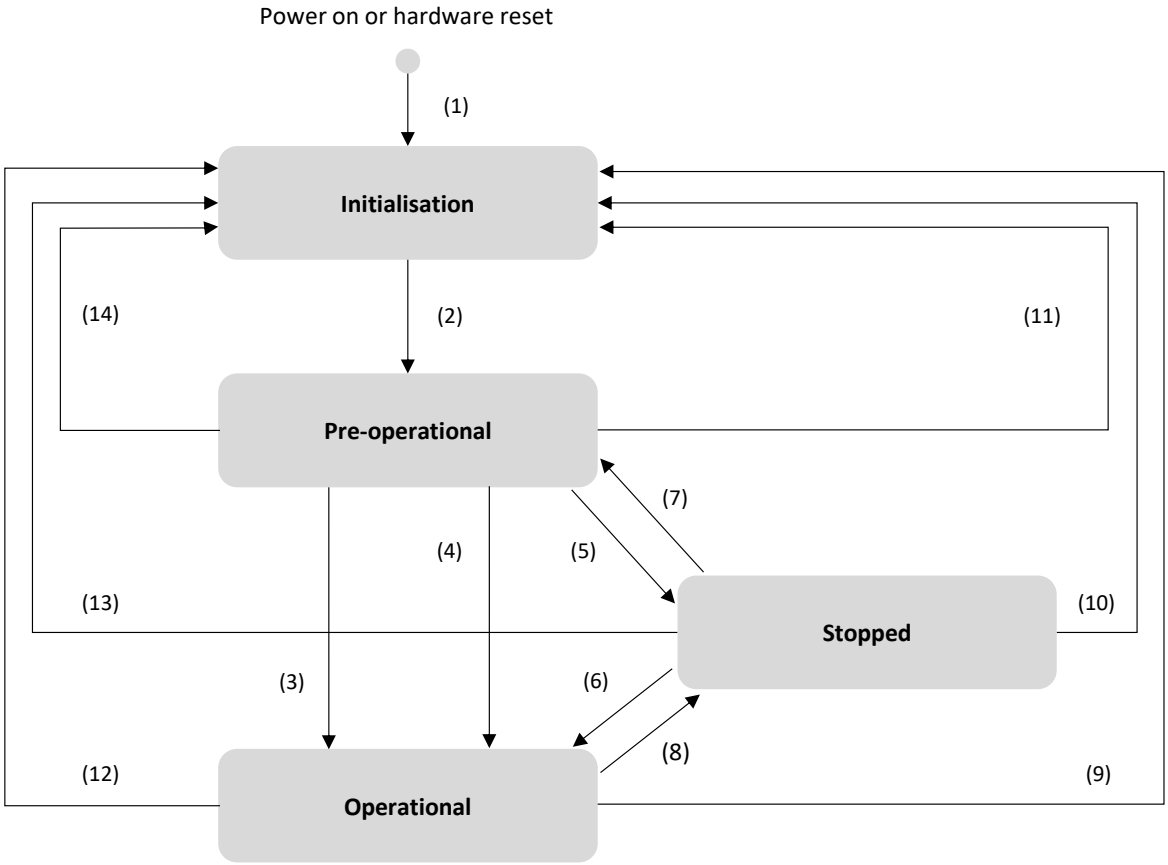


Figure 21 – NMT state machine diagram of a CANopen device

(1)	At Power on the NMT state initialization is entered autonomously
(2)	NMT state initialization finished – enter NMT state Pre-operational automatically
(3)	NMT service start remote node indication or by local control
(4), (7)	NMT service enter pre-operational indication
(5), (8)	NMT service stop remote node indication
(6)	NMT service start remote node indication
(9), (10), (11)	NMT service reset node indication
(12), (13), (14)	NMT service reset communication indication

The availability of the protocols depends on the state of the node, the table below presents this availability:

	Pre-operational	Operational	Stopped
PDO		✓	
SDO	✓	✓	
NMT	✓	✓	✓
EMCY	✓	✓	
SRDO		✓	
SYNC	✓	✓	
Heartbeat	✓	✓	✓
Timestamp	Not supported		

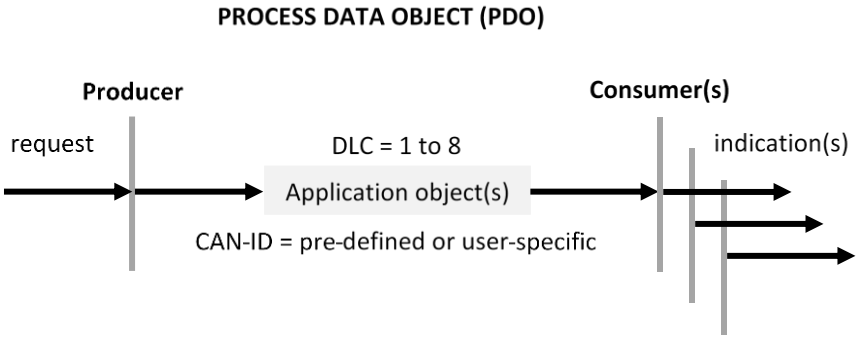
9.7. PDO (Process Data Object)

PDOs are messages used to exchange data in producer/consumer mode. A producer CANopen node sends a message which will be consumed by one or more consumer nodes. For the producer: the PDO is a TPDO (T: Transmitted), the same message is an RPDO (R: Received) for the consumer(s).

A PDO message is characterised by communication parameters including:

- Its identifier on the CAN bus (CAN-ID)
- Its transmission or reception modes
- Its activation

And mapping parameters, describing all the data carried by the message.



SWD® supports up to 8 transmit PDOs (TPDO) and 8 receive PDOs (RPDO).

Communication parameters

PDO communication parameters define whether a message is activated, its identity on the bus (CAN-ID), and its transmission or reception conditions.

For each of a node's messages, a CANopen dictionary lists its communication parameters. RPDO communication parameters are accessible from indexes 1400_h to 1407_h, and TPDO communication parameters from indexes 1800_h to 1807_h.

The configuration structure for communication parameters has the following format:

Index	Sub-index	Description	Data type
RPDOs: 1400_h to 1407_h **TPDOs: 1800_h to 1807_h**	00_h	Number of entries	Unsigned8
	01_h	COB-ID	Unsigned32
	02_h	Transmission type	Unsigned8
	03_h	Inhibit time	UNSIGNED16
	05_h	Event timer	UNSIGNED16

COB-ID (:01_h)

The COB-ID input can be used to specify:

- CAN-ID identity of the message on the bus
- Validity of the message

⚠ ***SWD®*** *supports 11-bit CAN-ID ('CAN base frame') message identifiers and does not implement the sending of TPDOs on RTR request.*

For RPDOs, the COB-ID format is:

31	30	29	28	11	10	0
Valid	RTR -	Frame 0_h	29-bit CAN-ID -		**11-bit CAN-ID**	
MSB						LSB

Bit(s)	Value	Description
Valid	**0_b** **1_b**	**PDO exists / is valid.** **PDO does not exist / is not valid**
RTR	-	*Ignored*
Frame	*0_b*	*11-bit CAN-ID valid (CAN base frame)*
29-bit CAN-ID	**-**	*Ignored*
11-bit CAN-ID	**x**	**11-bit CAN-ID of the CAN base frame**

For TPDOs, the COB-ID format is:

31	30	29	28	11	10	0
Valid	RTR 1_h	Frame 0_h	29-bit CAN-ID -		**11-bit CAN-ID**	
MSB						LSB

Bit(s)	Value	Description
Valid	**0_b** **1_b**	**PDO exists / is valid.** **PDO does not exist / is not valid**
RTR	*1_b*	*No RTR allowed on TPDO (RTR shall not be used with CANopen)*
Frame	*0_b*	*11-bit CAN-ID valid (CAN base frame)*
29-bit CAN-ID	**-**	*Ignored*
11-bit CAN-ID	**x**	**11-bit CAN-ID of the CAN base frame**

Transmission type (:02_h)

The transmission type parameter of a PDO specifies the transmission mode, and the triggering mode. The ***SWD®*** supports synchronous message triggering (SYNC) only.

Message transmission is triggered by reception of the synchronization message (SYNC). The trigger condition can be configured according to the number of (SYNC) messages received before transmission.

Value	Description	Supported
00h	Acyclic synchronous: *Triggered when the SYNC message is received and one of the mapped process data has changed its value after the last transmission.*	☐
01h	Cyclic synchronous (every sync)	☑
02h	Cyclic synchronous (every 2nd SYNC)	
03h	Cyclic synchronous (every 3rd SYNC)	
04h	Cyclic synchronous (every 4th SYNC)	
(…)	(…)	
F0h	Cyclic synchronous (every 240th SYNC)	
F1h	Reserved	
(…)		
FBh		
FCh	Synchronous RTR only: *not recommended anymore*	☐
FDh	Asynchronous RTR only: *not recommended anymore*	
FEh	Asynchronous: *Triggered by an internal event (e.g., change-of-state of one of the mapped process data or elapsing of the event-timer or any other event). The device manufacturer specifies the internal event triggering the TPDO transmission.*	
FFh	Asynchronous: *As before, but the CiA profile specifies the internal event triggering the TDPO transmission.*	

The transmission type specifies the transmission speed, based on the transmission period of the synchronisation message (SYNC).

- A transmission type equal to '00h', means that the message must be transmitted after the SYNC occurs, but acyclically (non-periodically), only if a data item mapped in a PDO has changed between two synchronization messages (SYNC).
- A transmission type equal to '01h', means that the message must be transmitted after reception of each synchronization message (SYNC).
- A transmission type between '01h' et 'F0h' (N), means that the message must be transmitted after every Nth SYNC object.

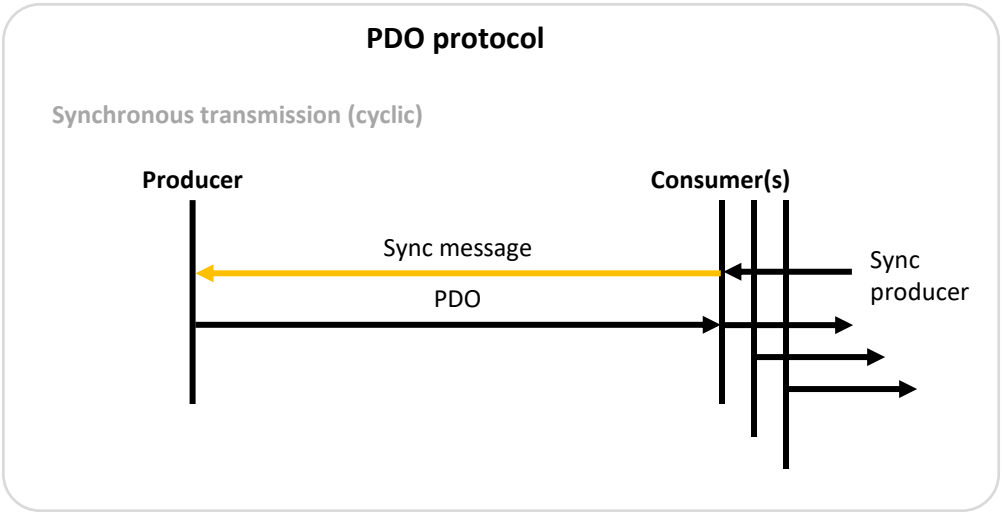


Figure 22 – PDO description

Inhibit time (:03)

The inhibit-time sub-parameter provides the time in ms, when this TPDO is allowed sending it again. The null value disables the inhibit time. It is not allowed to change the value while the PDO exists (bit 31 of sub-index 01_h is set to 0_b)

Event timer (:05)

The event timer sub-parameter specifies the period in ms of the TPDO transmission respectively the time-out (missing) of a RPDO. The null value disables the event-timer.

A RPDO uses this timer for deadline monitoring. If the timers expire before receiving a newer RPDO, it triggers an emergency message (EMCY), with error code: '8206: PDO_event_timer', and the node goes into error state.

Mapping parameters

PDO mapping configuration allows data included in a PDO to be pointed to an object in the dictionary, either on the transmit or receive side.

SWD® supports user reconfiguration of PDO mapping.

The data are concatenated in the data field of the CAN message:

- In reception mode (RPDO), mapping is used to decode the data received in the RPDO, and therefore to update the data in the ***SWD®*** dictionary. RPDO mapping is configured in objects 1600_h to 1607_h.
- In emission mode (TPDO), mapping is used to encode the data to be transmitted in the TPDO, which has been updated in the ***SWD®*** dictionary. TPDO mapping is configured in objects $1A00_h$ to $1A07_h$.

The configuration structure for mapping parameters has the following format:

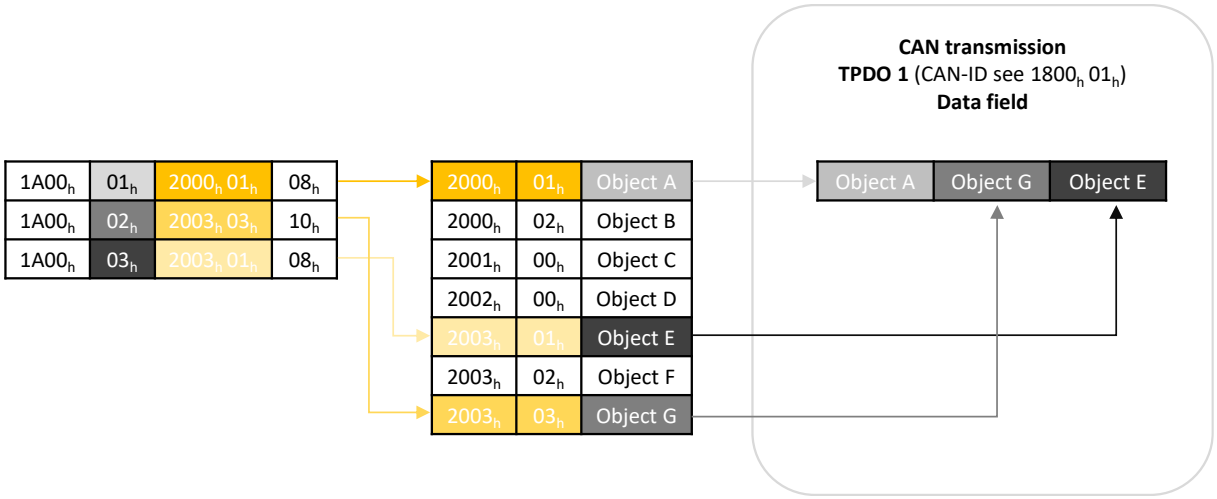


Figure 23 – TPDO mapping

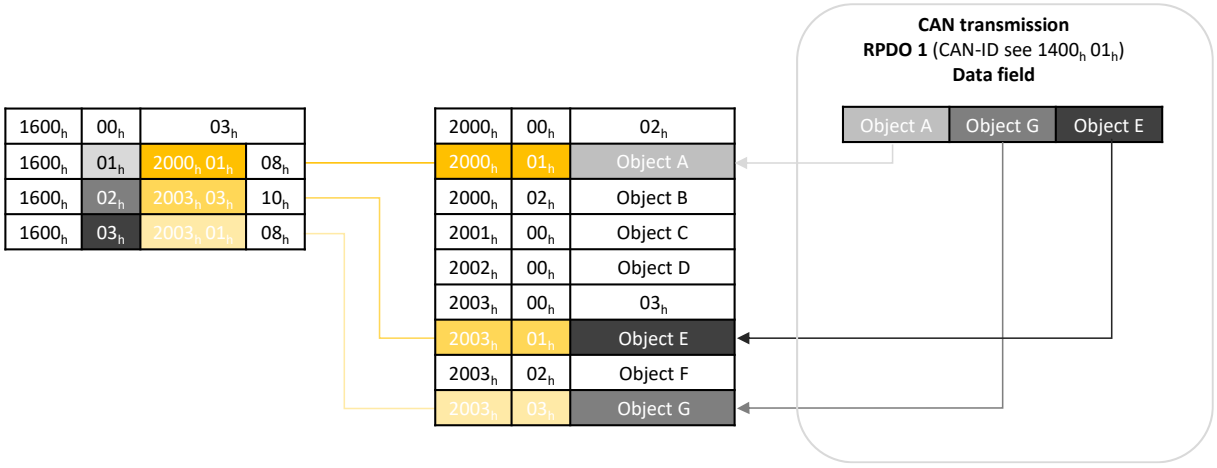


Figure 24 – RPDO mapping

Default values

Valid *(COB-ID bit 31)*	RPDOs	Mapping
☑	**1**	Controlword
☐	**2**	-
☐	**3**	-
☑	**4**	Controlword & vl target velocity
☐	**5**	-
☐	**6**	-
☐	**7**	-
☐	**8**	-

Valid *(COB-ID bit 31)*	TPDOs	Mapping
☑	**1**	Statusword
☐	**2**	Statusword
☑	**3**	Statusword & position value
☑	**4**	Statusword & velocity actual value
☐	**5**	Statusword
☐	**6**	Statusword
☐	**7**	Statusword
☐	**8**	Statusword

RPDO 1 : Controlword

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameters										
1400h	00h	Number of entries	U8	02h	RO	NO	-	02h	02h	1
	01h	COB-ID	U32	0000 0210h	RW	NO	-	200h	27Fh	4
	02h	Transmission type	U8	01h	RW	NO	-	0h	0FFh	1
	03h	Inhibit time	U16	00h	RO	NO	-	00h	00h	2

	05_h	Event timer	U16	00_h	RW	NO	ms	00_h	$FFFF_h$	2
Data mapping										
1600_h	00_h	Number of mapped	U8	01_h	RW	NO	-	00_h	08_h	1
	01_h	Mapping Entry 1	U32	$6040\ 0010_h$ Controlword	RW	NO	-	-	-	4
	02_h	Mapping Entry 2	U32	00_h	RW	NO	-	-	-	4
	03_h	Mapping Entry 3	U32	00_h	RW	NO	-	-	-	4
	04_h	Mapping Entry 4	U32	00_h	RW	NO	-	-	-	4
	05_h	Mapping Entry 5	U32	00_h	RW	NO	-	-	-	4
	06_h	Mapping Entry 6	U32	00_h	RW	NO	-	-	-	4
	07_h	Mapping Entry 7	U32	00_h	RW	NO	-	-	-	4
	08_h	Mapping Entry 8	U32	00_h	RW	NO	-	-	-	4

RPDO 4: Controlword & Target velocity

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameters										
1403_h	00_h	Number of entries	U8	02_h	RO	NO	-	02_h	-	1
	01_h	COB-ID	U32	$000\ 0510_h$	RW	NO	-	500_h	$57F_h$	4
	02_h	Transmission type	U8	01_h	RW	NO	-	0_h	$0FF_h$	1
	03_h	Inhibit time	U16	00_h	RO	NO	-	00_h	00_h	2
	05_h	Event timer	U16	00_h	RW	NO	ms	00_h	$FFFF_h$	2
Data mapping										
1603_h	00_h	Number of mapped	U8	02_h	RW	NO	-	01_h	08_h	1
	01_h	Mapping Entry 1	U32	$6040\ 0010_h$ Controlword	RW	NO	-	-	-	4
	02_h	Mapping Entry 2	U32	$6042\ 0010_h$ VI target velocity	RW	NO	-	-	-	4
	03_h	Mapping Entry 3	U32	00_h	RW	NO	-	-	-	4
	04_h	Mapping Entry 4	U32	00_h	RW	NO	-	-	-	4
	05_h	Mapping Entry 5	U32	00_h	RW	NO	-	-	-	4
	06_h	Mapping Entry 6	U32	00_h	RW	NO	-	-	-	4
	07_h	Mapping Entry 7	U32	00_h	RW	NO	-	-	-	4
	08_h	Mapping Entry 8	U32	00_h	RW	NO	-	-	-	4

TPDO 1: Statusword

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes

Communication parameters										
1800_h	00_h	Number of entries	U8	02_h	RO	NO	-	02_h	-	1
	01_h	COB-ID	U32	4000 0190_h	RW	NO	-	180_h	180_h + $7F_h$	4
	02_h	Transmission type	U8	01_h	RW	NO	-	00_h	$0FF_h$	1
Data mapping										
$1A00_h$	00_h	Number of mapped	U8	01_h	RW	NO	-	00_h	08_h	1
	01_h	Mapping Entry 1	U32	6041 0010_h Statusword	RW	NO	-	-	-	4
	02_h	Mapping Entry 2	U32	00_h	RW	NO	-	-	-	4
	03_h	Mapping Entry 3	U32	00_h	RW	NO	-	-	-	4
	04_h	Mapping Entry 4	U32	00_h	RW	NO	-	-	-	4
	05_h	Mapping Entry 5	U32	00_h	RW	NO	-	-	-	4
	06_h	Mapping Entry 6	U32	00_h	RW	NO	-	-	-	4
	07_h	Mapping Entry 7	U32	00_h	RW	NO	-	-	-	4
	08_h	Mapping Entry 8	U32	00_h	RW	NO	-	-	-	4

TPDO 3: Statusword & current position

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameters										
1802_h	00_h	Number of entries	U8	02_h	RO	NO	-	02_h	-	1
	01_h	COB-ID	U32	4000 0390_h	RW	NO	-	380_h	380_h + $7F_h$	4
	02_h	Transmission type	U8	01_h	RW	NO	-	00_h	$0FF_h$	1
Data mapping										
$1A02_h$	00_h	Number of mapped	U8	02_h	RW	NO	-	00_h	08_h	1
	01_h	Mapping Entry 1	U32	6041 0010_h Statusword	RW	NO	-	-	-	4
	02_h	Mapping Entry 2	U32	6064 0020_h position value	RW	NO	-	-	-	4
	03_h	Mapping Entry 3	U32	00_h	RW	NO	-	-	-	4
	04_h	Mapping Entry 4	U32	00_h	RW	NO	-	-	-	4
	05_h	Mapping Entry 5	U32	00_h	RW	NO	-	-	-	4
	06_h	Mapping Entry 6	U32	00_h	RW	NO	-	-	-	4
	07_h	Mapping Entry 7	U32	00_h	RW	NO	-	-	-	4
	08_h	Mapping Entry 8	U32	00_h	RW	NO	-	-	-	4

TPDO 4 : Statusword & current velocity

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes

Communication parameters										
1803$_h$	00$_h$	Number of entries	U8	02$_h$	RO	NO	-	02$_h$	-	1
	01$_h$	COB-ID	U32	4000 0490$_h$	RW	NO	-	480$_h$	480$_h$ + 7F$_h$	4
	02$_h$	Transmission type	U8	01$_h$	RW	NO	-	00$_h$	0FF$_h$	1
Data mapping										
1A03$_h$	00$_h$	Number of mapped	U8	02$_h$	RW	NO	-	00$_h$	08$_h$	1
	01$_h$	Mapping Entry 1	U32	6041 0010$_h$ Statusword	RW	NO	-	-	-	4
	02$_h$	Mapping Entry 2	U32	606C 0020$_h$ Velocity actual value	RW	NO	-	-	-	4
	03$_h$	Mapping Entry 3	U32	00$_h$	RW	NO	-	-	-	4
	04$_h$	Mapping Entry 4	U32	00$_h$	RW	NO	-	-	-	4
	05$_h$	Mapping Entry 5	U32	00$_h$	RW	NO	-	-	-	4
	06$_h$	Mapping Entry 6	U32	00$_h$	RW	NO	-	-	-	4
	07$_h$	Mapping Entry 7	U32	00$_h$	RW	NO	-	-	-	4
	08$_h$	Mapping Entry 8	U32	00$_h$	RW	NO	-	-	-	4

PDOs configuration method

The following method is used to change the mapping or communication parameters of PDOs in the pre-operational NMT state:

1. Set node to pre-operational state (Pre-Op) by sending NMT command 'Enter Pre-Op'.
2. Disable PDO by setting COB-ID bit 31 'valid' to 1$_b$ (sub-index 01h, PDO communication parameter).
3. Disable mapping by setting sub-index 00h of the mapping parameters to 00$_h$.
4. Modify mapping by changing the values of the corresponding sub-indices.
5. Enable mapping by setting sub-index 00$_h$ to the number of mapped objects.
6. Create the PDO by setting bit 31 'valid' to 0$_b$ in the associated COB-ID.
7. Save configuration of communication parameters (cf. 9.10).
8. Load new configuration via NMT command 'Reset communication'.
9. Start node by sending NMT 'Start' command.

9.8. Emergency (EMCY)

Presentation

The emergency message service (EMCY) is based on the producer/consumer principle. When the ***SWD®*** detects an error, it acts as a producer, transmitting a message on the CAN bus indicating the error code. Other nodes can act as consumers, by reading the message containing the error code.

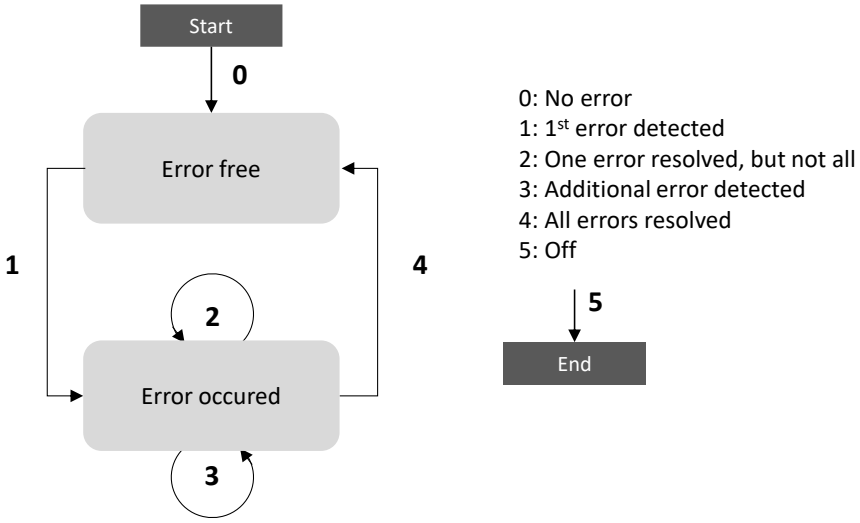


Figure 25 – EMCY error state machine

0	After initialisation, The ***SWD®*** goes into error-free state. No emergency message is sent.
1	The ***SWD®*** detects a 1st error. It goes into error state. An emergency message can be sent with the error code and associated error register. The error code corresponds to the first two bytes of the message, and the error register to the third byte. The error code is set in the object '1003_h: Predefined error field'. The error register is set in object '1001_h: Error Register'.
2	An error disappears, but one or more errors are still present. An emergency message containing error code 0000_h (error reset) may be transmitted together with the remaining errors in the error register and in the manufacturer-specific error field.
3	A new error appears. The ***SWD®*** remains in error state and can transmit an emergency message with error code and associated error register.
4	All errors have disappeared. The ***SWD®*** goes into error-free state and transmits an emergency object with the error code "reset error / no error".
5	Reset or power off.

The list of errors causing an emergency/no message (EMCY) to be sent, are listed below in the error code table.

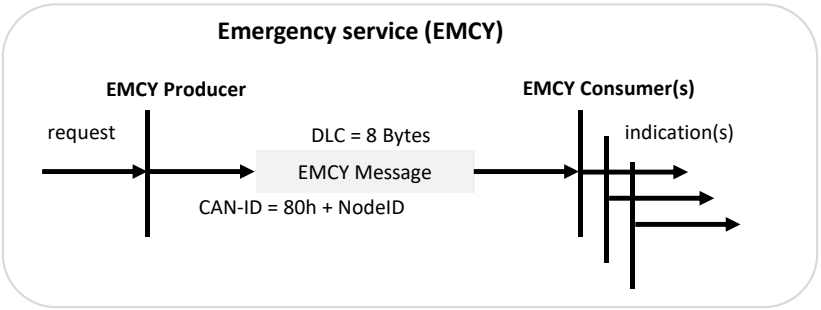
The link between error state (EMCY) and CANopen node state (NMT) is defined by the 'Error behavior' object (1029_h). If an error or alert is detected in 'Operational' mode, it can be configured to switch the machine to 'pre-Operational', 'Stopped' or to maintain its NMT state. An error can occur in CAN bus communication, and an error or alert can appear in the application.

⚠ *When using multiple devices, it is strongly recommended to set the 'Application error' object to 1. In this case, an application error will not cause the node to enter the NMT 'pre-Operational' state and will not stop the emission of SRDOs safety messages.*

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
1029h	00h	Number of entries	U8	03h	RO	NO	-	-	-	1
1029h	01h	Communication error	U8	00h	RW	NO	-	-	-	1
1029h	02h	Application error	U8	01h	RW	NO	-	-	-	1
1029h	03h	Application warning	U8	01h	RW	NO	-	-	-	1

Value	**00h**	When this error occurs: the node enters the NMT 'pre-Operational' state
	01h	When this error occurs: no change in NMT state of the node
	02h	When this error occurs: the node enters the NMT 'Stopped' state

Emergency messages (EMCY) are transmitted in the following format:



EMCY Message		
Error code	**Error register**	**Manufacturer-specific**
2 Bytes	1 Byte	5 Bytes

Error code

Reset	Description	EMCY
0000h	Reset error or no error	☐

Generic	Description	EMCY
1000 h	Generic error	☐

Current	Description	EMCY
2221h	Motor overcurrent error - Error level	☐
2222h	Motor overcurrent error - Warning level	☐

Voltage	Description	EMCY
3211h	DC overvoltage error - Error level	☑
3212h	DC overvoltage fault - Warning level	☑
3221h	DC undervoltage error - Error level	☑
3222h	DC undervoltage error - Warning level	☑

Temperature	Description	EMCY
4000h	Internal error temperature too high	☐
4210h	Internal error Power stage temperature too high	☑
4310h	Internal error Drive temperature too high	☑

Software	Description	EMCY
6000h	CANopen software error	☑
6020h	CANopen software error: CRC safety mapping / configuration	☑

Manufacturer	Description	EMCY
7100h	Power supply error on CAN or IO connector	☑
7121h	Motor blocked error	☑

Monitoring	Description	EMCY
8001h	Fault on safety input STO_1	☑
8002h	Fault on safety input STO_2	☑
8003h	INSafe_1 safety input failure	☑
8004h	INSafe_2 safety input failure	☑
8005h	INSafe_3 safety input failure	☑
8006h	INSafe_4 safety input failure	☑
8007h	Internal encoder coherence failure	☑
8008h	Internal failure switching-off driver	☑

8009h	STO signal consistency failure	☑
800Ah	Internal brake activation failure	☑
800Bh	Internal brake management failure	☑
800Ch	Internal failure disconnecting power supply to driver	☑
800Dh	Internal failure on DC power	☑
800Eh	Internal STO management failure	☑
8010h	Error external brake disconnected	☑
8011h	External brake overcurrent error	☑
8012h	SBU activated	☑
8013h	Error activating SBU	☑
8014h	Error inconsistent diagnosis on N_CC_STATE	☑
8015h	Error diagnosing MCU_NBRAKE	☑
8016h	Error diagnosing BRAKE_LOCK_CHECK	☑
8017h	Error in external brake activation command	☑
8018h	Error internal brake interrupt check	☑
8050h	Generic driver error	☑

Communication	Description	EMCY
8120h	CAN bus error - error passive	☑
8140h	CAN bus error - return from bus off	☑

Protocol	Description	EMCY
8201h	SRDO protocol error - SCT not respected	☑
8202h	SRDO protocol error - SRVT not respected	☑
8203h	SRDO protocol error - data inconsistency	☑
8204h	SRDO protocol error - message missing	☑
8205h	SRDO protocol error - wrong message size	☑
8206h	RPDO protocol error – 'Event timer' timeout	☑

Error register

This object is an 8-bit field, indicating error classes. Each bit corresponds to a class:

Bit	M/O	Signification
0	M	Generic error
1	O	Current
2	O	Voltage
3	O	Temperature
4	O	Communication error
5	O	*Device specific*
6	O	*Reserved*
7	O	*Manufacturer specific*

Error class and history

Object 1001_h contains current error class.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
1001_h	0	Error Register	U8	0	RO	TPDO	-	0	FF_h	1

Object 1003_h contains the list of errors. The oldest error is stored in the object with the highest sub-index. The most recent error is stored in the object with sub-index 1. The number of errors stored in the table corresponds to the value stored in sub-index 0.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
1003_h	0	Number of errors	U8	0	RO	-	-	0	FE_h	1
	1..FE_h	Standard error field	U32	-	RO	-	-	-	-	4

EMCY COB-ID

The COB-ID value of the emergency message (EMCY) can be set from object 1014_h. The default value is (80h + NodeID):

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
1014_h	0	COB-ID EMCY message	U32	80h + NodeID	RW	-	-	-	-	1

The COB-ID format is:

31	**30**	**29**	**28**	**11**	**10**	**0**
Valid	RTR 0_h	Frame 0_h	29-bit CAN-ID -		**11-bit CAN-ID**	
MSB						LSB

EMCY COB-ID description table:

Bit(s)	Value	Description
Valid	**0_b** **1_b**	EMCY **exists / is valid.** EMCY **does not exist / is not valid**
RTR	*0_b*	*reserved*
Frame	*0_b*	*11-bit CAN-ID valid (CAN base frame)*
29-bit CAN-ID	**-**	*Ignored*
11-bit CAN-ID	**x**	**11-bit CAN-ID of the CAN base frame**

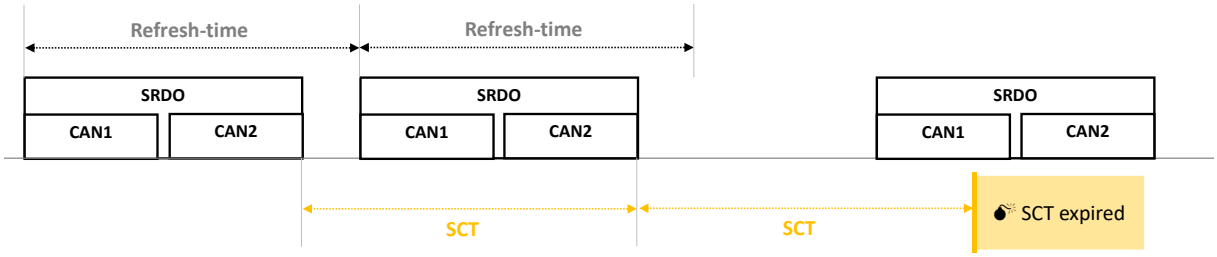
9.9. SRDO (Safety-Relevant Data Object)

The SRDO protocol included in CANopen Safety enables the transmission of safety-related data. Data exchange between two safety-related nodes using the CANopen Safety protocol is guaranteed up to Safety Integrity Level 3 (SIL 3). SRDOs consist of two CAN messages (CAN1 and CAN2). The second message contains the information of the 1st message, but inverted bit by bit.

Periodicity control between messages (SCT):

Cyclic transmission speed and refresh time are monitored. The SCT (Safe-guard Cycle Time) defines the maximum time between two periodic transmissions of an SRDO. If the time elapsed between two SRDO receptions exceeds the SCT, the SRDO consumer signals the SCT event. The SRLD (Safety-Related Logical Device) must then switch to a safe state.

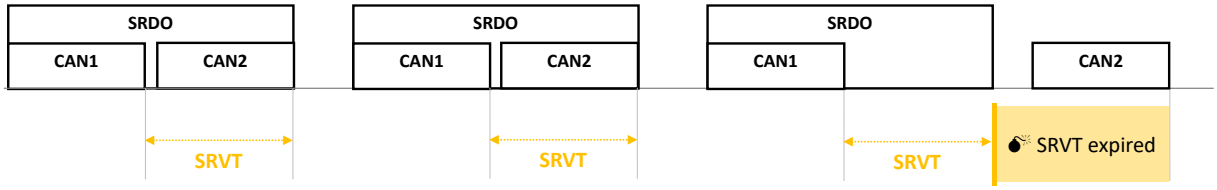
The figure below illustrates the SCT mechanism:



Time control between CAN frames (SRVT)

An SRDO message includes two CAN frames (CAN1 and CAN2) whose reception time between the two frames is also monitored. The SRVT (Safety-Related object Validation Time) defines this duration. If the time elapsed between two CAN frames (CAN1 and CAN2) is greater than the SRVT, the SRDO consumer signals the SRVT event. The SRLD (Safety-Related Logical Device) must then switch to the safe state.

The figure below illustrates the SRVT mechanism:



Data consistency checks

The data contained in the two CAN frames (CAN1 and CAN2) must be consistent with each other. The second message contains the bit-by-bit complement of this data.

Communication parameters

Index	Sub-index	Description	Data type
SRDO1: 1301h **SRDO2: 1302h** **SRDO 9 à 16:** **1309h to 1310h**	00h	Number of entries	Unsigned8
	01h	Direction	Unsigned8
	02h	SCT	Unsigned16
	03h	SRVT	Unsigned8
	04h	Transmission type	Unsigned8
	05h	COB-ID 1	Unsigned32
	06h	COB-ID 2	Unsigned32

Time (SCT et SRVT)

In transmission, the SCT value defines the period at which the SRDO message is to be sent.

At the receiving end, the SCT and SRVT times are used to configure monitoring. The default values are 50ms and 20ms respectively. Times are expressed in ms and the value 0 is forbidden.

⚠ SRVT must always be lower than SCT, otherwise an error will occur.

Message direction:

Value	Description
00_h	Invalid (does not exist)
01_h	(TX) Transmission
02_h	(RX) Reception
03_h	Reserved
(…)	
FF_h	

Transmission type:

Value	Description	Supported
00_h	Acyclic synchronous (every sync if data has changed)	☐
01_h	Cyclic synchronous (every sync)	☐
02_h	Cyclic synchronous (every 2nd SYNC)	☐
03_h	Cyclic synchronous (every 3rd SYNC)	☐
04_h	Cyclic synchronous (every 4th SYNC)	☐
(…)	(…)	☐
$F0_h$	Cyclic synchronous (every 240th SYNC)	☐
FC_h	Reserved	
(…)		
FD_h		
FE_h	Asynchronous: *Triggered by an internal event*	☑
FF_h	Asynchronous: *As before, but the CiA profile specifies the internal event SRDO transmission.*	☐

ℹ Any change to the transmission type generates an 'abort code'.

Default values

SRDO	Valid	Direction	Mapping
1	☑	RX (02_h)	Safety control word 1 (6620 0108_h)
2	☑	TX (01_h)	Safety status word 1 à 8 (6621 0108_h -> 6621 0808_h)
3	Reserved		
(…)			
8			
9	☐	RX	Safety control word 2 (6620 0208_h)
10	☐	TX	Safe position actual value i32 (6611 0020_h) + Safe velocity actual value i32 (6613 0020_h)[12]

[12] *Implemented from firmware version 1.1.4*

11	☐	RX	Safety control word 3 (6620 0308$_h$)
12	☐	RX	Safety control word 4 (6620 0408$_h$)
13	☐	RX	Safety control word 5 (6620 0508$_h$)
14	☐	RX	Safety control word 6 (6620 0608$_h$)
15	☐	RX	Safety control word 7 (6620 0708$_h$) + Safety control word 8 (6620 0808$_h$)
16	☐	TX	Safety control word Safein 1 (2620 0208$_h$)

SRDO 1 RX

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
1301$_h$	00$_h$	Number of entries	U8	06$_h$	RO	NO	-	-	-	1
	01$_h$	Information direction	U8	2 (RX)	RW	NO	-	0 / 2 (RX)		1
	02$_h$	SCT	U16	50	RW	NO	ms	1	U16	2
	03$_h$	SRVT	U16	20	RW	NO	ms	1	U16	2
	04$_h$	Transmission type	U8	FE$_h$	RW	NO		FE$_h$	FE$_h$	1
	05$_h$	COB ID 1	U32	0000 00FF$_h$ + (2 x node-ID)	RW	NO				4
	06$_h$	COB ID 2	U32	0000 0100$_h$ + (2 x node-ID)	RW	NO				4
Mapping										
1381$_h$	00$_h$	Number of entries	U8	02$_h$	RO	NO	-	00$_h$	08$_h$	1
	01$_h$	Safety Controword 1	U8	6620 0108$_h$ SCW_1	RO	NO	-	-	-	4
	02$_h$	Safety Controword 1 inverted	U8	6622 0108$_h$ SCW_1_inv	RO	NO	-	-	-	4

SRDO 2 TX

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
1302h	00$_h$	Number of entries	U8	06$_h$	RO	NO	-	-	-	1
	01$_h$	Information direction	U8	1 (TX)	RW	NO	-	0 / 1 (TX)		1
	02$_h$	SCT	U16	25	RW	NO	ms	1	U16	2
	03$_h$	SRVT	U16	20	RW	NO	ms	1	U16	2
	04$_h$	Transmission type	U8	FE$_h$	RW	NO		FE$_h$	FE$_h$	1
	05$_h$	COB ID 1	U32	0000 0103$_h$	RW	NO				4
	06$_h$	COB ID 2	U32	0000 0104$_h$	RW	NO				4
Mapping										
1382$_h$	00$_h$	Number of entries	U8	16$_h$	RO	NO	-	00$_h$	08$_h$	1
	01$_h$	Safety Statusword 1	U32	6621 0108$_h$	RO	NO	-	-	-	4

	02_h	Safety Statusword 1 inverted	U32	6623 0108$_h$	RO	NO	-	-	-	4
	03_h	Safety Statusword 2	U32	6621 0208$_h$	RO	NO	-	-	-	4
	04_h	Safety Statusword 2 inverted	U32	6623 0208$_h$	RO	NO	-	-	-	4
	05_h	Safety Statusword 3	U32	6621 0308$_h$	RO	NO	-	-	-	4
	06_h	Safety Statusword 3 inverted	U32	6623 0308$_h$	RO	NO	-	-	-	4
	07_h	Safety Statusword 4	U32	6621 0408$_h$	RO	NO	-	-	-	4
	08_h	Safety Statusword 4 inverted	U32	6623 0408$_h$	RO	NO	-	-	-	4
	09_h	Safety Statusword 5	U32	6621 0508$_h$	RO	NO	-	-	-	4
	$0A_h$	Safety Statusword 5 inverted	U32	6623 0508$_h$	RO	NO	-	-	-	4
	$0B_h$	Safety Statusword 6	U32	6621 0608$_h$	RO	NO	-	-	-	4
	$0C_h$	Safety Statusword 6 inverted	U32	6623 0608$_h$	RO	NO	-	-	-	4
	$0D_h$	Safety Statusword 7	U32	6621 0708$_h$	RO	NO	-	-	-	4
	$0E_h$	Safety Statusword 7 inverted	U32	6623 0708$_h$	RO	NO	-	-	-	4
	$0F_h$	Safety Statusword 8	U32	6621 0808$_h$	RO	NO	-	-	-	4
	10_h	Safety Statusword 8 inverted	U32	6623 0808$_h$	RO	NO	-	-	-	4

SRDO 9 RX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
1309_h	00_h	Number of entries	U8	06_h	RO	NO	-	-	-	1
	01_h	Information direction	U8	0 (inactive)	RW	NO	-	0 / 2 (RX)		1
	02_h	SCT	U16	50	RW	NO	ms	1	U16	2
	03_h	SRVT	U16	20	RW	NO	ms	1	U16	2
	04_h	Transmission type	U8	FE_h	RW	NO		FE_h	FE_h	1
	05_h	COB ID 1	U32	0000 00FF_h + (2 x node-ID)	RW	NO				4
	06_h	COB ID 2	U32	0000 0100_h + (2 x node-ID)	RW	NO				4
Mapping										
1389_h	00_h	Number of entries	U8	02_h	RO	NO	-	00_h	08_h	1
	01_h	Safety Controword 2	U32	6620 0208$_h$	RO	NO	-	-	-	4
	02_h	Safety Controword 2 inverted	U32	6622 0108$_h$	RO	NO	-	-	-	4

SRDO 10 TX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
$130A_h$	00_h	Number of entries	U8	06_h	RO	NO	-	-	-	1

	01_h	Information direction	U8	0 (inactive)	RW	NO	-	0 / 1 (TX)		1
	02_h	SCT	U16	25	RW	NO	ms	1	U16	2
	03_h	SRVT	U16	20	RW	NO	ms	1	U16	2
	04_h	Transmission type	U8	FE_h	RW	NO		FE_h	FE_h	1
	05_h	COB ID 1	U32	$0000\ 0107_h$	RW	NO				4
	06_h	COB ID 2	U32	$0000\ 0108_h$	RW	NO				4
Mapping										
$138A_h$	00_h	Number of entries	U8	04_h	RO	NO	-	00_h	08_h	1
	01_h	Safe position actual value	U32	$6611\ 0020_h$	RO	NO	-	-	-	4
	02_h	Safe position actual value inverted	U32	$661A\ 0020_h$	RO	NO	-	-	-	4
	03_h	Safe velocity actual value inverted	U32	$6613\ 0020_h$	RO	NO	-	-	-	4
	04_h	Safe velocity actual value inverted	U32	$661C\ 0020_h$	RO	NO	-	-	-	4

SRDO 11 RX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
$130B_h$	00_h	Number of entries	U8	06_h	RO	NO	-	-	-	1
	01_h	Information direction	U8	0 (inactive)	RW	NO	-	0 / 2 (RX)		1
	02_h	SCT	U16	50	RW	NO	ms	1	U16	2
	03_h	SRVT	U16	20	RW	NO	ms	1	U16	2
	04_h	Transmission type	U8	FE_h	RW	NO		FE_h	FE_h	1
	05_h	COB ID 1	U32	$0000\ 0109_h$	RW	NO				4
	06_h	COB ID 2	U32	$0000\ 010A_h$	RW	NO				4
Mapping										
$138B_h$	00_h	Number of entries	U8	02_h	RO	NO	-	00_h	08_h	1
	01_h	Safety Controword 3	U32	$6620\ 0308_h$	RO	NO	-	-	-	4
	02_h	Safety Controword 3 inverted	U32	$6622\ 0308_h$	RO	NO	-	-	-	4

SRDO 12 RX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
$130C_h$	00_h	Number of entries	U8	06_h	RO	NO	-	-	-	1
	01_h	Information direction	U8	0 (inactive)	RW	NO	-	0 / 2 (RX)		1
	02_h	SCT	U16	50	RW	NO	ms	1	U16	2

	03_h	SRVT	U16	20	RW	NO	ms	1	U16	2
	04_h	Transmission type	U8	FE_h	RW	NO		FE_h	FE_h	1
	05_h	COB ID 1	U32	0000 $010B_h$	RW	NO				4
	06_h	COB ID 2	U32	0000 $010C_h$	RW	NO				4
Mapping										
$138C_h$	00_h	Number of entries	U8	02_h	RO	NO	-	00_h	08_h	1
	01_h	Safety Controword 4	U32	6620 0408_h	RO	NO	-	-	-	4
	02_h	Safety Controword 4 inverted	U32	6622 0408_h	RO	NO	-	-	-	4

SRDO 13 RX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
$130D_h$	00_h	Number of entries	U8	06_h	RO	NO	-	-	-	1
	01_h	Information direction	U8	0 (inactive)	RW	NO	-	0 / 2 (RX)		1
	02_h	SCT	U16	50	RW	NO	ms	1	U16	2
	03_h	SRVT	U16	20	RW	NO	ms	1	U16	2
	04_h	Transmission type	U8	FE_h	RW	NO		FE_h	FE_h	1
	05_h	COB ID 1	U32	0000 $010D_h$	RW	NO				4
	06_h	COB ID 2	U32	0000 $010E_h$	RW	NO				4
Mapping										
$138D_h$	00_h	Number of entries	U8	02_h	RO	NO	-	00_h	08_h	1
	01_h	Safety Controword 5	U32	6620 0508_h	RO	NO	-	-	-	4
	02_h	Safety Controword 5 inverted	U32	6622 0508_h	RO	NO	-	-	-	4

SRDO 14 RX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
$130E_h$	00_h	Number of entries	U8	06_h	RO	NO	-	-	-	1
	01_h	Information direction	U8	0 (inactive)	RW	NO	-	0 / 2 (RX)		1
	02_h	SCT	U16	50	RW	NO	ms	1	U16	2
	03_h	SRVT	U16	20	RW	NO	ms	1	U16	2
	04_h	Transmission type	U8	FE_h	RW	NO		FE_h	FE_h	1
	05_h	COB ID 1	U32	0000 $010F_h$	RW	NO				4
	06_h	COB ID 2	U32	0000 0110_h	RW	NO				4

Mapping										
138E$_h$	00$_h$	Number of entries	U8	02$_h$	RO	NO	-	00$_h$	08$_h$	1
	01$_h$	Safety Controword 6	U32	6620 0608$_h$	RO	NO	-	-	-	4
	02$_h$	Safety Controword 6 inverted	U32	6622 0608$_h$	RO	NO	-	-	-	4

SRDO 15 RX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
130F$_h$	00$_h$	Number of entries	U8	06$_h$	RO	NO	-	-	-	1
	01$_h$	Information direction	U8	0 (inactive)	RW	NO	-	0 / 2 (RX)		1
	02$_h$	SCT	U16	50	RW	NO	ms	1	U16	2
	03$_h$	SRVT	U16	20	RW	NO	ms	1	U16	2
	04$_h$	Transmission type	U8	FE$_h$	RW	NO		FE$_h$	FE$_h$	1
	05$_h$	COB ID 1	U32	0000 0111$_h$	RW	NO				4
	06$_h$	COB ID 2	U32	0000 0112$_h$	RW	NO				4
Mapping										
138F$_h$	00$_h$	Number of entries	U8	04$_h$	RO	NO	-	00$_h$	08$_h$	1
	01$_h$	Safety Controword 7	U32	6620 0708$_h$	RO	NO	-	-	-	4
	02$_h$	Safety Controword 7 inverted	U32	6622 0708$_h$	RO	NO	-	-	-	4
	03$_h$	Safety Controword 8	U32	6620 0808$_h$	RO	NO	-	-	-	4
	04$_h$	Safety Controword 8 inverted	U32	6622 0808$_h$	RO	NO	-	-	-	4

SRDO 16 TX (inactive)

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Communication parameter										
1310h	00$_h$	Number of entries	U8	06$_h$	RO	NO	-	-	-	1
	01$_h$	Information direction	U8	0 (inactive)	RW	NO	-	0 /1 (TX)		1
	02$_h$	SCT	U16	25	RW	NO	ms	1	U16	2
	03$_h$	SRVT	U16	20	RW	NO	ms	1	U16	2
	04$_h$	Transmission type	U8	FE$_h$	RW	NO		FE$_h$	FE$_h$	1
	05$_h$	COB ID 1	U32	0000 0113$_h$	RW	NO				4
	06$_h$	COB ID 2	U32	0000 0114$_h$	RW	NO				4
Mapping										
1390$_h$	00$_h$	Number of entries	U8	2$_h$	RO	NO	-	00$_h$	08$_h$	1
	01$_h$	Safety Statusword INSafe	U32	2620 0208$_h$	RO	NO	-	-	-	4

	02_h	Safety Statusword INSafe inverted	U32	$2622\ 0208_h$	RO	NO	-	-	-	4

9.10. Store and restore configurations

The ***SWD®*** supports saving user parameters and restoring factory settings. This is managed when the node is stopped (Stopped) or in pre-operational state (Pre-operational).

User parameters are saved when the 'save' command is received in the 'Store Parameter' object (1010_h):

Store Parameters 1010_h				
Signature	MSB			LSB
/ISO8859/ character	e	v	a	s
Value	65_h	76_h	61_h	73_h

Saved parameters are divided into several groups. Saving is performed either for all product parameters, or for a specific group. This depends on which sub-index the 'save' command is written to:

Sub-index	Group of data	Supported
1	Save all Parameters	☑
2	Save Communication Parameters	☐
3	Save Application Parameters	☐
4	Save Manufacturer Parameters	☐
5	Save LSS Parameters	☐
6	Save Motor Calibration and Internal Parameters	☐
7	Save Test Results	☐

The default configuration is restored by writing 'load' to the 'Restore Default parameters' object (1011_h):

Restore Default Parameters 1011_h				
Signature	MSB			LSB
/ISO8859/ character	d	a	o	l
Value	64_h	61_h	$6F_h$	$6C_h$

The parameters displayed are divided into several groups. Parameters are returned either for all product parameters, or for a specific group. This depends on which sub-index the 'load' command is written to:

Sub-index	Group of data	Supported
1	Restore all Default Parameters	☑
2	Restore Communication Default Parameters	☑
3	Restore Application Default Parameters	☑
4	Restore Manufacturer Parameters	☐
5	Restore LSS Parameters	☐
6	Restore Motor Calibration and Internal Parameters	☐

7	Restore Test Results	☐

Communication parameters include objects from index 1000_h to $1FFF_h$.
Manufacturer-specific parameters and therefore ***SWD®*** parameters, from index 2000_h to $5FFF_h$.
Parameters specific to the profile and therefore to the motor drive, from index 6000_h to $9FFF_h$.

The list of objects is available to download from the website, in the resources tab.

Versions in html format (readable in your web browser), and a version in EDS (Electronic Data Sheet) format for machines and PLCs are available.

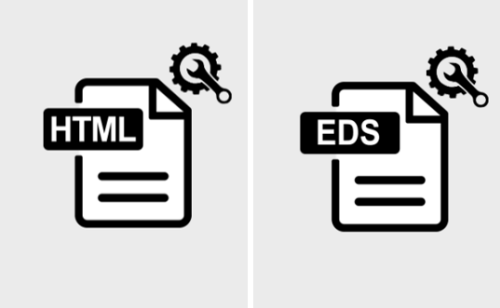


*Figure 26 - **SWD®** CANopen Object Dictionary, download icons*

https://ez-wheel.com/en/resources

10. CiA 402: Motor control unit profile

CiA profile 402 specifies a generic motor interface (PDS: Power Drive System). This interface handles various operating modes, such as position, speed, torque, etc.

An operating mode is associated with parameters that must be configured to start the state machine. State machine control depends on the chosen operating mode. It is used to control the motor.

The ***SWD®*** drive implements the *Velocity mode*.

10.1. State machine

The CiA 402 profile state machine is used to control the operation of the drive integrated into the ***SWD®***:

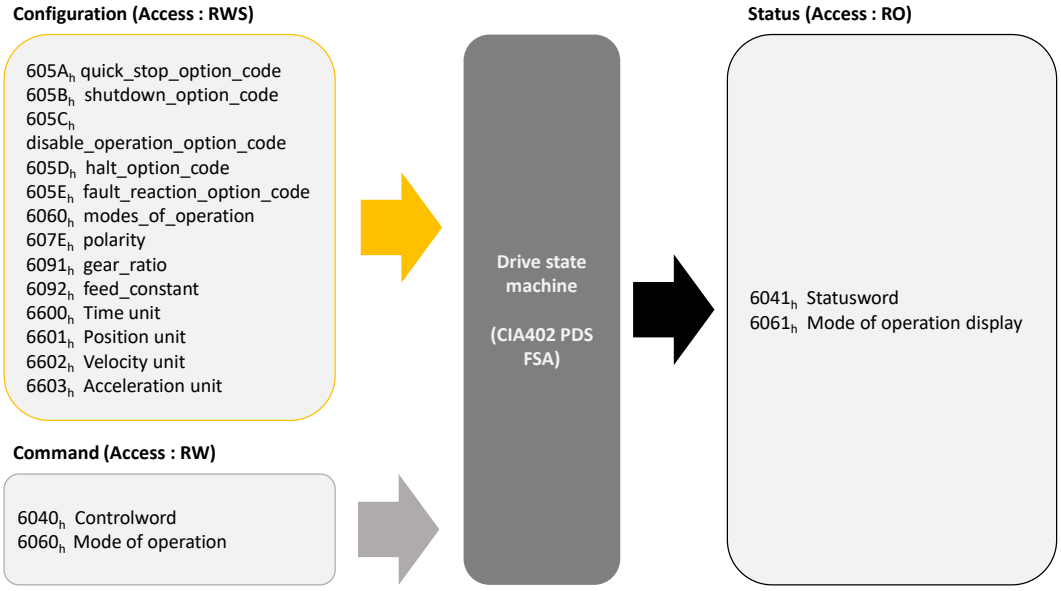


Figure 27 – Drive status management interface

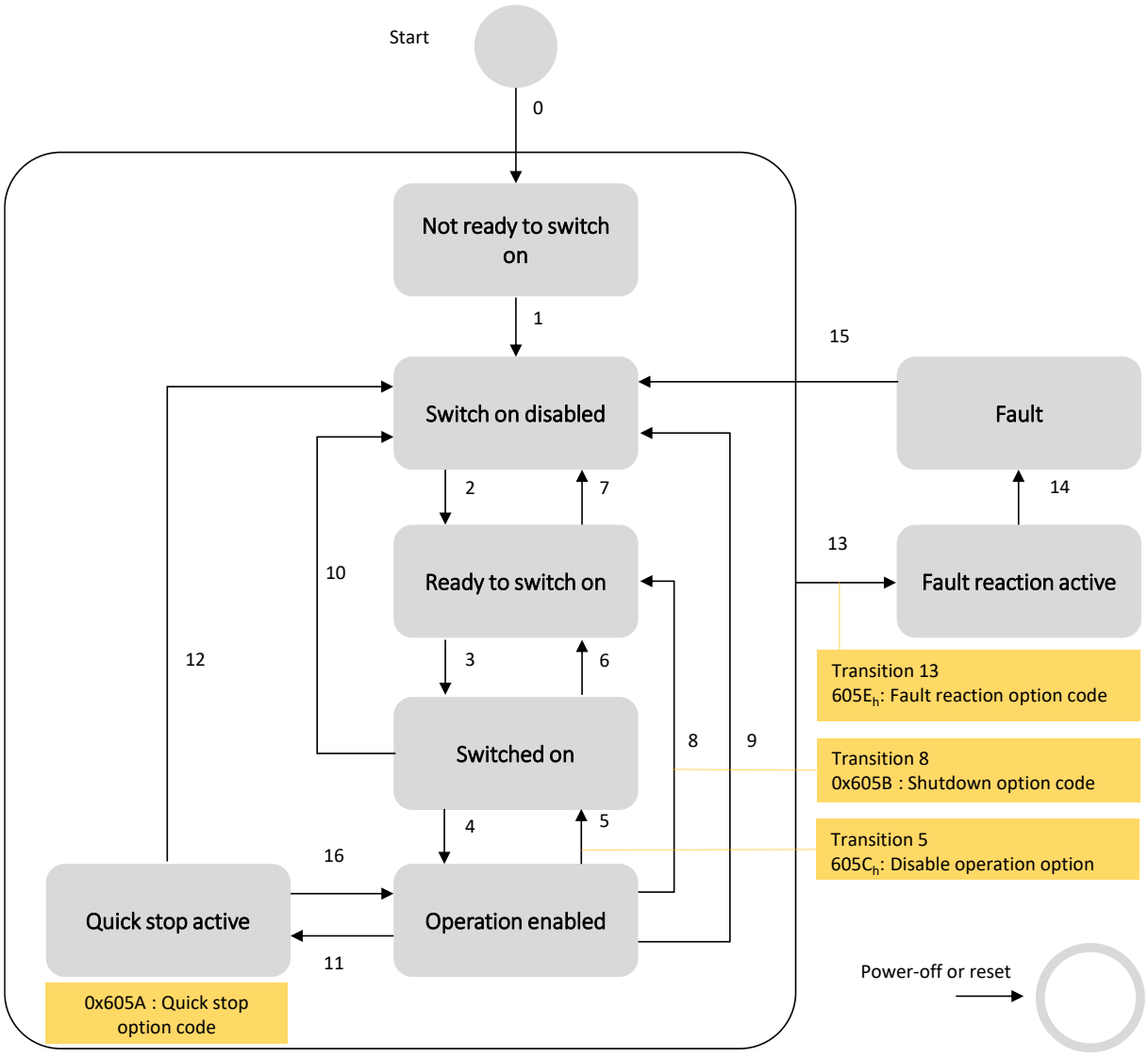


Figure 28 – CiA 402, state machine and transitions

⚠ *Reading the CiA-402 object '6041$_h$: Statusword', provides the status of the state machine. This makes it possible to determine possible transitions.*

Transition	Events	Actions
0	Automatic transition after power-on or reset application	Drive device self-test and/or self - initialization shall be performed
1	Automatic transition	Communication shall be activated
2	Shutdown command from control device or local signal	None
3	Switch on command received from control device or local signal	The high-level power shall be switched on, if possible
4	Enable operation command received from control device or local signal	The drive function shall be enabled and all internal set-points cleared
5	Disable operation command received from control device or local signal	The drive function shall be disabled
6	Shutdown command received from control device or local signal	The high-level power shall be switched off, if possible
7	Quick stop or disable voltage command from control device or local signal	None
8	Shutdown command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible
9	Disable voltage command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible
10	Disable voltage or quick stop command from control device or local signal	The high-level power shall be switched off, if possible
11	Quick stop command from control device or local signal	The quick stop function shall be started.
12	Automatic transition when the quick stop function is completed and quick stop option code is 1, 2, 3 or 4, or disable voltage command received from control device (depends on the quick stop option code)	The drive function shall be disabled, and the high-level power shall be switched off, if possible
13	Fault signal	The configured fault reaction function shall be executed
14	Automatic transition	The drive function shall be disabled; the high-level power shall be switched off, if possible
15	Fault reset command from control device or local signal *(↑ rising edge)*	A reset of the fault condition is carried out, if no fault exists currently on the drive device; after leaving the fault state, the fault reset bit in the Controlword shall be cleared by the control device
16	Enable operation command from control device, if the quick stop option code is 5, 6, 7, or 8	The drive function shall be enabled

CiA standard 402-2 defines a set of supported functions for controlling machine states. A summary table of supported function states for each machine stage is available below:

The following functions are supported:

- *Active brake, (configurable with object 2050 00_h 'cia402_use_internal_brake'[13]).*
- *Electronic board under voltage*
- *Active engine control*
- *Configuration authorised*

Their On/Off states are as follows:

Function	FSA states							
	Not ready to switch on	Switch on disabled	Ready to switch on	Switched on	Operation enabled	Quick stop active	Fault reaction active	Fault
Active brake	**Yes**	**Yes**	**Yes**	**Yes**	**Yes/No**	**Yes/No**	**Yes**	**Yes**
Card under voltage	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Active engine control	**No**	**No**	**No**	**No**	**Yes**	**Yes**	**Yes**	**No**
Authorised configuration	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**	**No**	**Yes**

Figure 29 – Configuration of functions according to CiA 402 state machine states

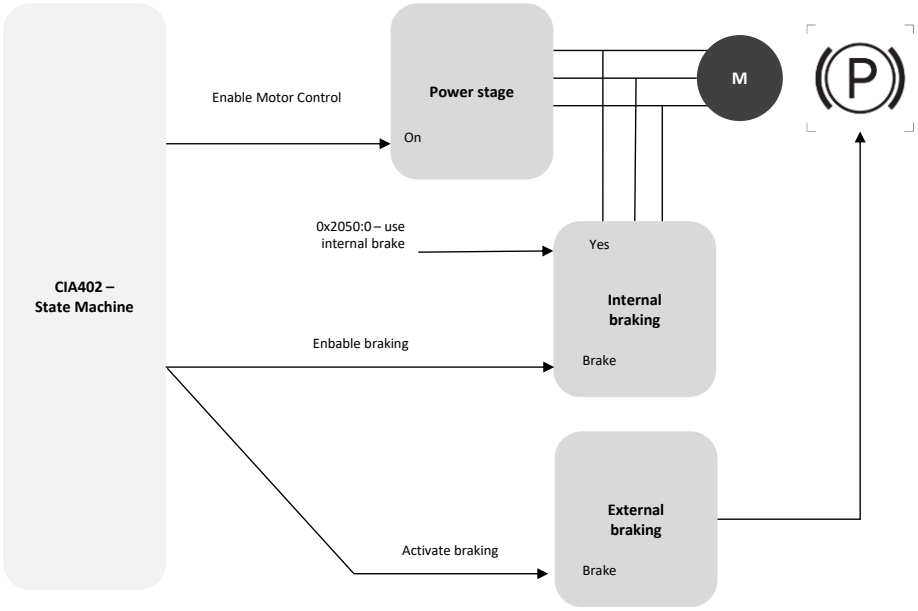


Figure 30 – Motor control overview

	Brake present	Brake not present
Internal brake	2050 00_h cia402_use_internal_brake = true	2050 00_h cia402_use_internal_brake = false
External brake[14]	Mechanical assembly and connection to brake connector	Not mounted / connected

[13] Configuration possible under development, not active by default

[14] An external brake can only be connected and activated as of software version 'Firmware 2.0.x'.

6040h Controlword

The '6040h Controlword' object is used to control the CiA 402 PDS state machine and request the transition from one state to another. The transition is made by writing the 'Controlword', or sometimes automatically, for example in the event of an error (if configured).

Some transitions can perform specific, configurable actions, such as a safe shutdown with automatic deceleration ramp.

15	11	10	9	8	7	6	4	3	2	1	0
ms		r	oms	h	fr	oms		eo	qs	ev	so
MSB											LSB

Key	
ms	manufacturer-specific
r	reserved
oms	operation mode specific
h	halt
fr	fault reset
eo	enable operation
qs	quick stop
ev	enable voltage
so	switch on

Command	Controlword bitfield					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Switch on + enable operation	0	1	1	1	1	3 + 4 (NOTE)
Disable voltage	0	X	X	0	X	7, 9, 10, 12
Quick stop	0	X	X	1	X	7, 10, 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4, 16
Fault reset	↑ rising edge	X	X	X	X	15

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
6040h	0	Controlword	U16	0	RW	R	-	-	-	2

6041h Statusword

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms		oms		ila	tr	rm	ms	w	sod	qs	ve	f	oe	so	rtso
MSB															LSB

Key	
ms	manufacturer-specific
oms	operation mode specific
ila	internal limit active
tr	target reached
rm	remote
w	warning
sod	switch on disabled
qs	quick stop
ve	voltage enable
f	fault
oe	operation enabled
so	switched on
rtso	ready to switch on

Ila - internal limit active

The drive sets the 'Ila' bit to 1 if the speed setpoint is outside the speed range authorized for the product (See 6046_h - vl_velocity_min_max_amount).

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
6041_h	0	Statusword	U16	0	RO	T	-	-	-	2

Start-up example

SWD® state	Status word (typical)	Control word for next state
Not ready to switch on	$00\ 00_h$	Automatic
Switch on disabled	$00\ 40_h$	$00\ 06_h$
Ready to switch on	$00\ 21_h$	$00\ 07_h$
Switched on	$00\ 23_h$	$00\ 0F_h$
Operation enabled	$00\ 27_h$	

NB: Controlword value for a fault reset is $00\ 80_h$ (↑Rising edge of bit 7).

Operating modes

The ***SWD®*** drive implements the 'velocity' mode, which enables speed control of the motor. Supported operating modes are available by reading the "6502_h Supported drive mode" object.

31	16	15	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer-specific		reserved		cstca	cst	csv	csp	ip	hm	r	tq	pv	vl	pp
MSB														LSB

	Key
pp	profile position mode
vl	velocity mode
pv	profile velocity mode
tq	torque profile mode

r	reserved
hm	homing mode
ip	interpolated position mode
csp	cyclic sync position mode
csv	cyclic sync velocity mode
cst	cyclic sync torque mode
cstca	cyclic sync torque mode with commutation angle
r(eserved)	reserved

The operating mode defines the behaviour when the ***SWD®*** is in the 'operation enable' state.

The object "Modes of operation" (6060_h) is used to select the operating mode. The object "Modes of operation display" (6061_h) displays the current ***SWD®*** operating mode. The value associated with the operating mode is shown below:

Value	Definition	Supported
-128 to -1	Manufacturer-specific operation modes	☐
0	No mode change/no mode assigned	☐
+1	Profile position mode (pp)	☐
+2	**Velocity mode (vl)**	☑
+3	Profile velocity mode (pv)	☐
+4	Torque profile mode (tq)	☐
+5	Reserved (r)	☐
+6	Homing mode (hm)	☐
+7	Interpolated position mode	☐
+8	Cyclic sync position mode	☐
+9	Cyclic sync velocity mode	☐
+10	Cyclic sync torque mode	☐
+11	Cyclic sync torque mode with commutation angle	☐
+12 to +127	Reserved	☐

ℹ By default, velocity mode is activated.

The default operating mode can be saved in non-volatile memory:

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
6502_h	0	Supported operating mode	U32	2	RO	NO	Table	2	2	4
6060_h	0	Operating mode request	U32	2	RW	NO	Table	0	2	4
6061_h	0	Operating mode	U32	2	RO	NO	Table	0	2	4

10.2. 'Velocity mode' (vl) operation

Introduction

The 'velocity' operating mode controls the speed of the ***SWD®*** motor.

Speed is controlled by an acceleration or deceleration ramp, which limits the current load and mechanical stress on the motor.

Load-dependent force calculation is performed automatically, and a feedback loop ensures that the speed setpoint is followed without exceeding the acceleration or deceleration limits.

The velocity mode consists of a transfer function whose various stages produce internal or external values, some of which can be inspected by reading CANopen objects (described in the next subsection).

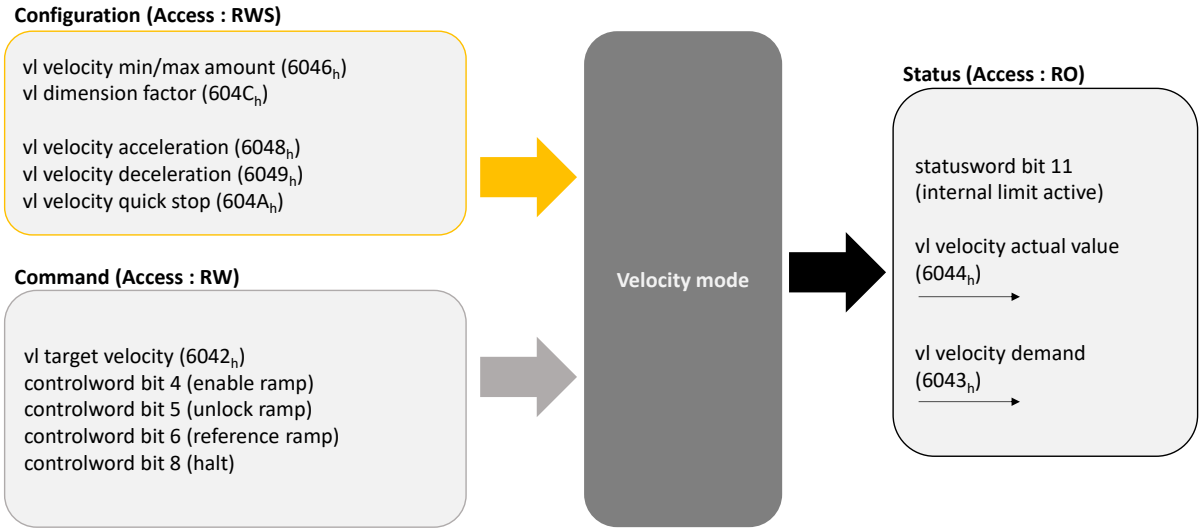


Figure 31 – 'Speed control' interface

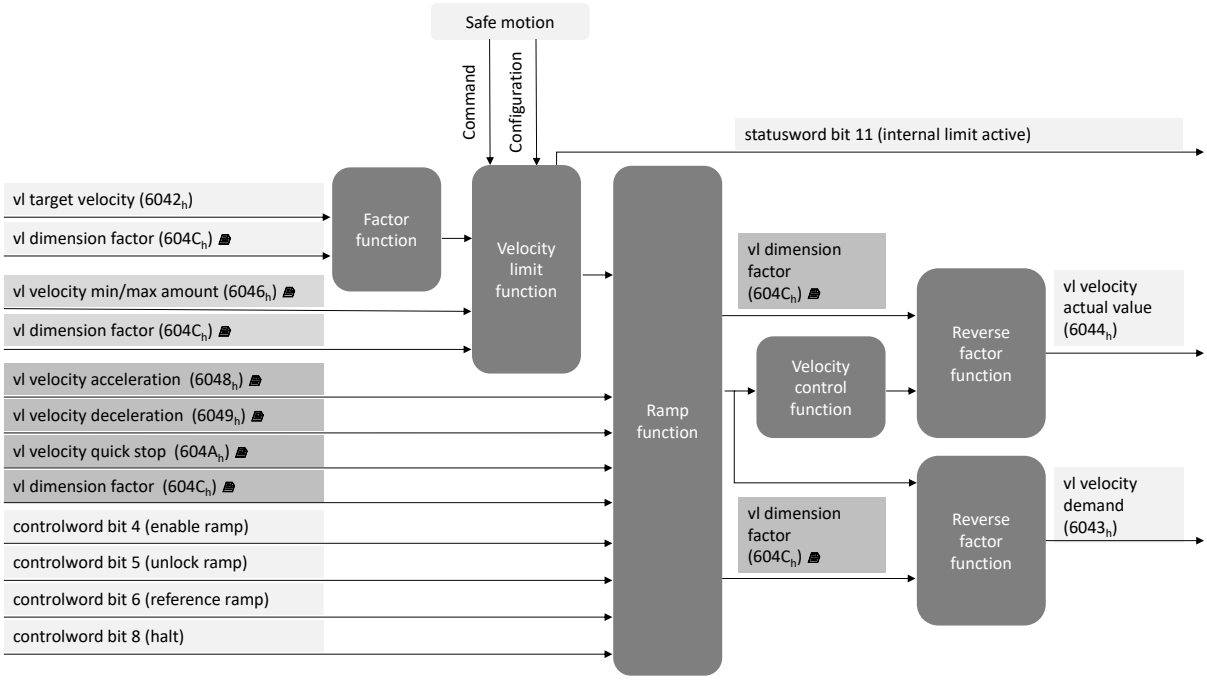


Figure 32 – 'Speed control' architecture

6042h: Speed setpoint

The speed setpoint is transmitted via the dictionary input 'vl_target_velocity' (6042h).

By default, speeds are expressed in revolutions per minute at the motor shaft, regardless of the mechanical gearbox ratio.

⚠ The speed unit used depends on the Dimension factor configuration (604Ch).

6064h: Position

Position is expressed in motor encoder increments and is available in the dictionary entry 'position_value' (6064h).

The ***SWD®***'s built-in encoder resolution is 30 increments per motor revolution. Positive or negative counting direction can be configured via the 'polarity' object (607Eh).

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
6064h	0	position_value	I32	0	R0	TPDO	inc			4	Yes

607Eh: Polarity

By configuration, it is possible to change the direction of motor rotation corresponding to a positive speed. Similarly, it is possible to change the direction in which position increments are incremented.

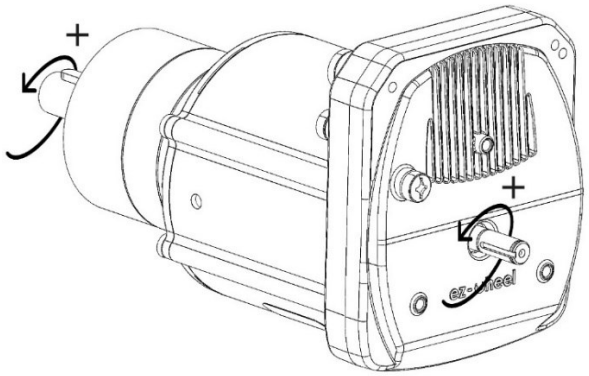


Figure 33 – Positive direction of rotation (+)

7	6	5	0
Position polarity	Velocity polarity	Reserved (0)	
MSB			LSB

Polarity bits are coded as follows: 0b = multiply by 1 and 1b = multiply by -1

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
607Eh	0	Polarity	U8	0	RW	RPDO	-	0	1	1

⚠ *The 'polarity' object has no impact on the positive direction of rotation used for the SDIp and SDIn safety functions.*

604C$_h$: vl_dimension_factor

It is possible to apply a dimension factor to the expression of speeds, so that they are expressed in another unit. The dimension factor can be configured using the following objects:

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
604C$_h$	1	vl_dimension_factor_numerator	I32	1	RW	RPDO	-	I32 excepted 0		4
604C$_h$	2	vl_dimension_factor_denominator	I32	1	RW	RPDO	-	I32 excepted 0		4

$$Speed_{RPM} = \frac{Numerator}{Denominator} \times Speed$$

When it is not equal to 1, the dimension factor applies to all the following parameters:

6046	vl_velocity_min_max_amount
6048	vl_velocity_acceleration
6049	vl_velocity_deceleration
604A	vl_velocity_quick_stop
6693	SLS velocity limit u32
66D5	SDI velocity zero window u32

Speed Limitations

The setpoint speed limitation block limits the speed range accessible in the setpoint. It acts symmetrically on positive and negative setpoints.

In the case of a setpoint greater in absolute value than the 'velocity_max_amount' setpoint, the limiting output setpoint will be equal to plus or minus 'velocity_max_amount'.

In the case of a setpoint lower in absolute value than the 'velocity_min_amount' setpoint, the limiting output setpoint will be equal to zero.

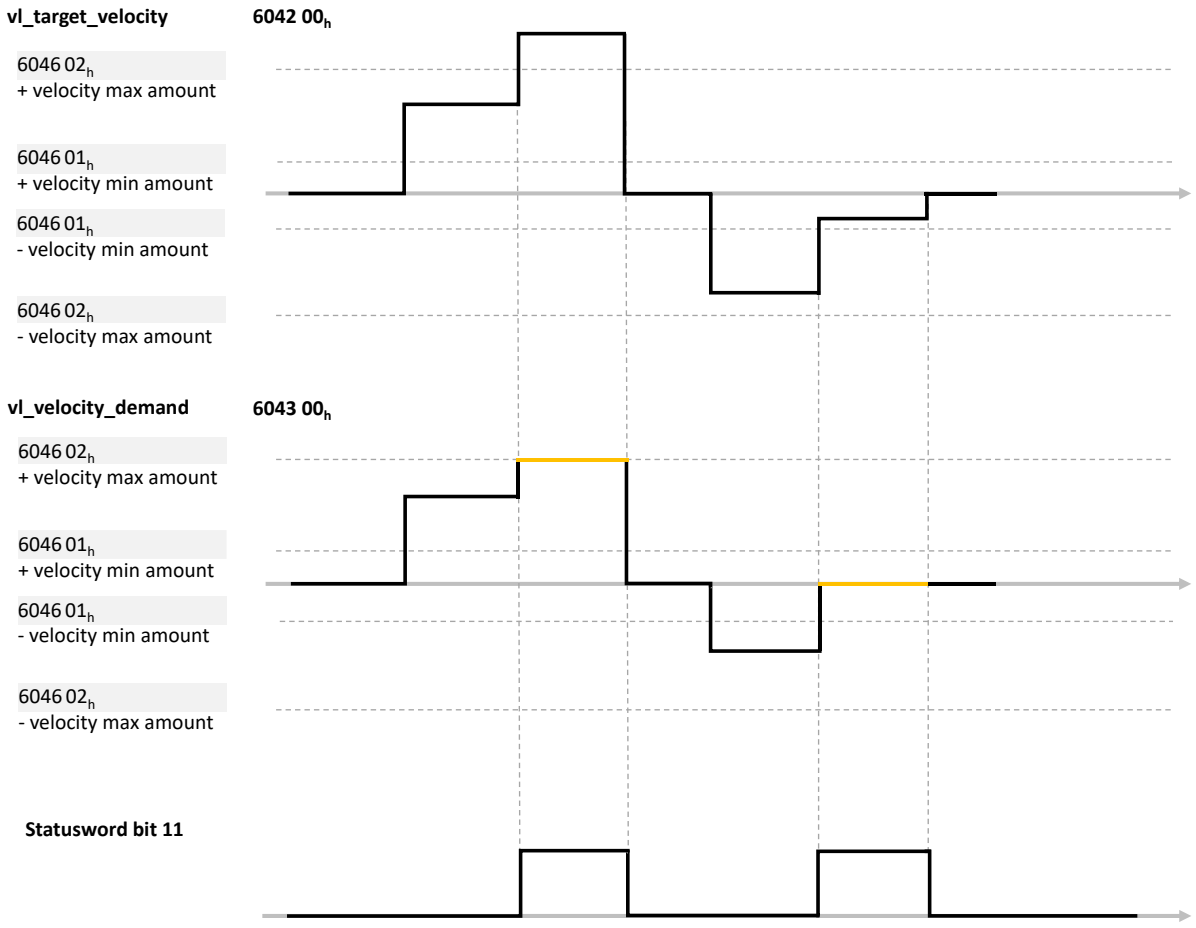


Figure 34 – Speed limit management

⚠ Limits are set in motor rpm, i.e., before the GearBox.
⚠ The speed unit used depends on the 'vl_dimension_factor' configuration (604C$_h$).

The user can modify the limits used by the drive:

- During runtime, limitation values are taken into consideration when switching to 'operation enable'.
- By configuration, by saving the limits to be used by default at product startup.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
6046$_h$	1	vl_velocity_min_amount	U32	5	RW	RPDO	RPM	30	1800	4	Yes
6046$_h$	2	vl_velocity_max_amount	U32	2000	RW	RPDO	RPM	30	1800	4	Yes

Speed limitation and safety functions

Additional setpoint limitations are applied when safety functions are activated. This limitation of the setpoint is not in itself a safety function but provides an early way of dealing with the constraints associated with a safety function.

Active function	Output command
STO	Disconnection of motor torque
SMS	Maximum speed is set regardless of the requested safety-related functions
SLS	Speed is restricted below SLS speed set. Same behavior as 'velocity max amount'.
SLSa	Speed is restricted below SLSa speed independently for each direction.
SDI p	Positive speed commands above nZero_SDI are forced to null speed.
SDI n	Negative speed commands below -nZero_SDI are forced to null speed.

The ramps used for this limitation are the acceleration (6048_h) and deceleration (6049_h) ramps.

The 11th bit of the Statusword is also set in the case of setpoint limitation related to a safety function.

⚠ *Limiting the setpoint does not guarantee that the motor speed will remain within the limits imposed by the safety function, and that an error management response will not be triggered. A setpoint consistent with the active safety functions must be set at application level.*

Ramps

The ramp function smoothes the setpoint by limiting setpoint variations during acceleration or deceleration.

The output of the ramp function is the data used as motor control input.

There are 3 ramp configurations:

- An acceleration ramp used when the setpoint increases in absolute value
- A nominal deceleration ramp used when the setpoint decreases in absolute value
- A fast deceleration ramp

Each ramp is configured by setting 2 variables in the dictionary:

- A speed variation 'delta_speed' expressed by default in rpm (Cf. $604C_h$)
- A time variation delta_temps expressed in seconds

The ramp is then calculated as follows:

$$Ramp = \frac{delta_{speed}}{delta_{time}}$$

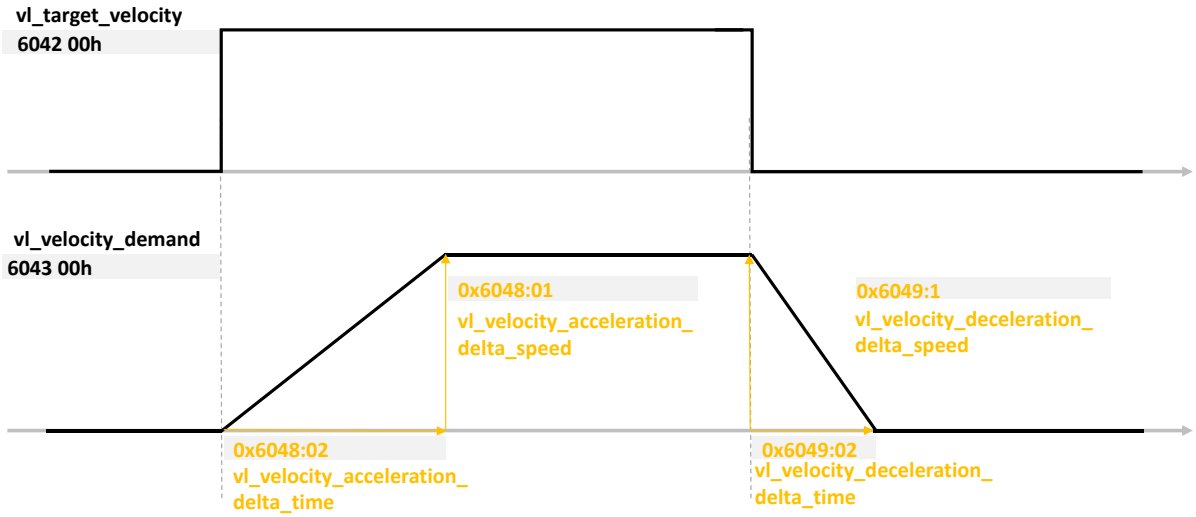


Figure 35 – Acceleration/deceleration ramp management (1/2)

When the setpoint direction is reversed, the nominal deceleration ramp is used to return to zero speed. The acceleration ramp is then used to reach the new setpoint.

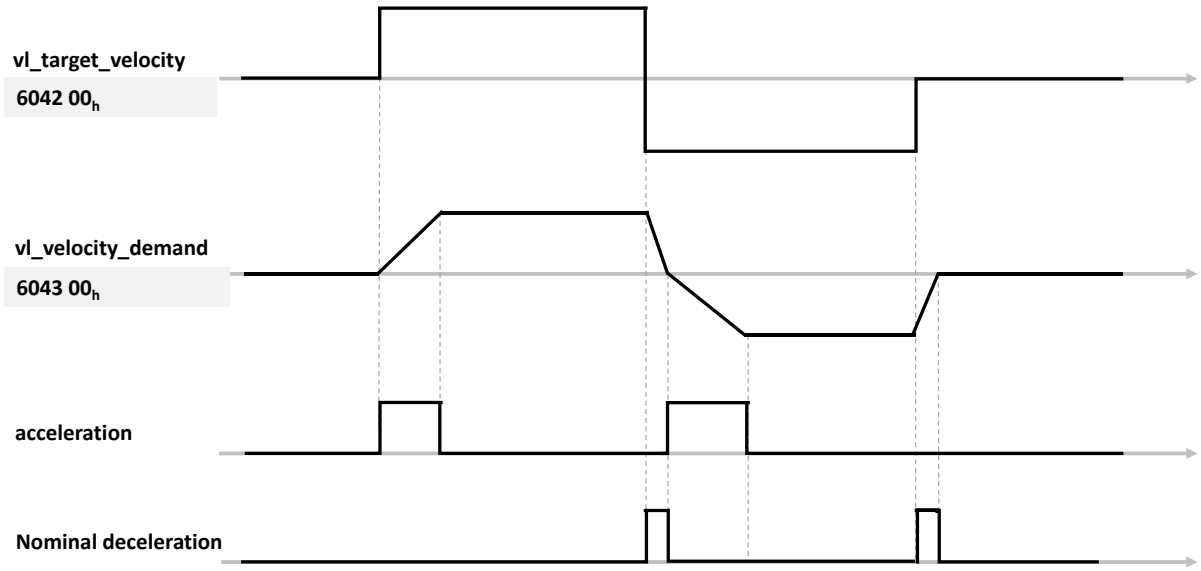


Figure 36 – Managing acceleration/deceleration ramps (2/2)

Ramp function control

The ramp function can be controlled via the controlword bit field.

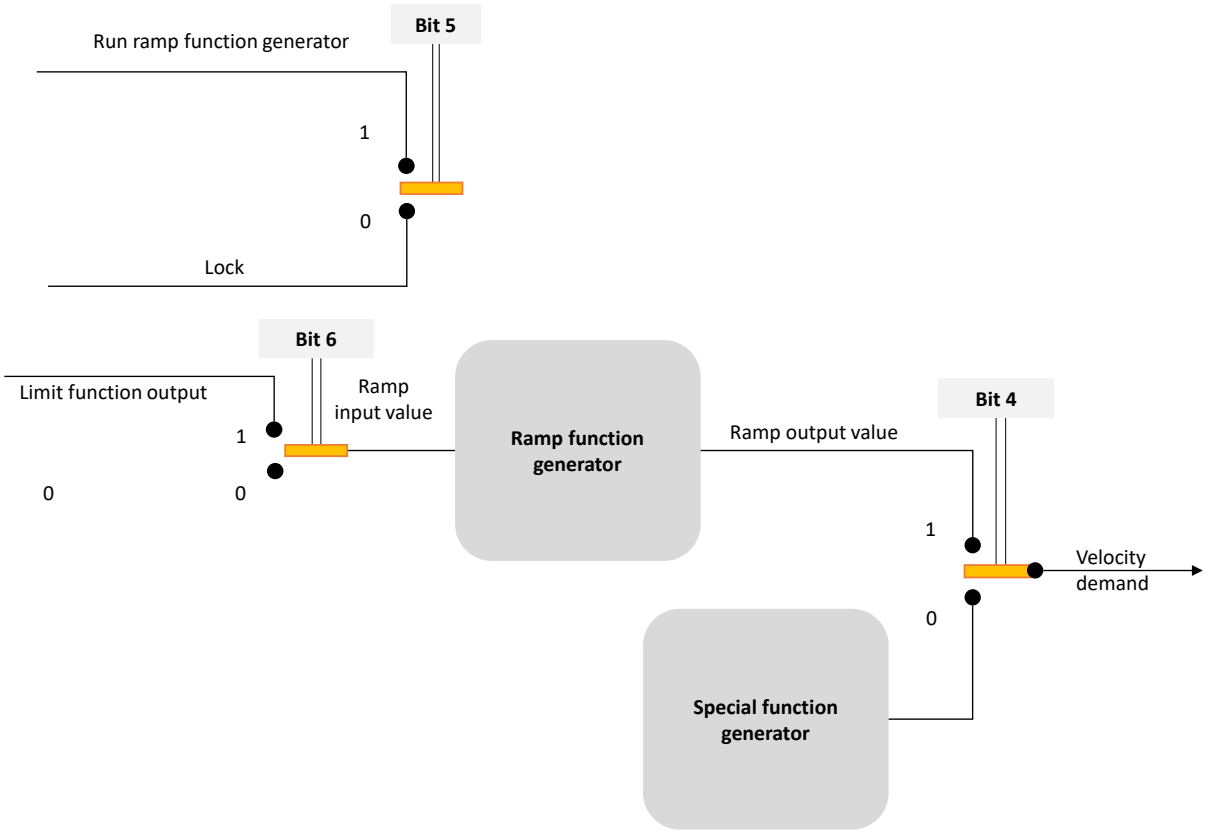


Figure 37 – Use of controlword bits in velocity mode

Bit	Bitfield coding in velocity mode	
4	Activate ramp	0: Ramp module is deactivated, bloc output is identical to the input 1: Ramp module is activated
5	Unlock ramp	0: Ramp output bloc is fixed 1: Ramp output bloc follows the input applying ramps
6	Ramp reference	0: Ramp input bloc is forced to null, only deceleration ramp is used 1: Ramp input bloc corresponds to the speed limitation bloc
8	Halt	0: No command, nominal behaviour 1: Motor stops, used ramp depends on '605D Halt option code'

Acceleration ramp

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
6048h	1	vl_velocity_ acceleration_ delta_speed	U32	500	RW	RPDO	RPM[15]	1	10000	4	Yes
6048h	2	vl_velocity_ acceleration_ delta_time	U16	1	RW	RPDO	s	1	100	2	Yes

Deceleration ramp

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
6049h	1	vl_velocity_ deceleration_ delta_speed	U32	500	RW	RPDO	RPM[16]	1	10000	4	Yes
6049h	2	vl_velocity_ deceleration_ delta_time	U16	1	RW	RPDO	s	1	100	2	Yes

Deceleration ramp in 'Quick stop' mode

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
604Ah	1	vl_velocity_ quick_stop_ delta_speed	U32	1000	RW	RPDO	RPM[17]	1	10000	4	Yes
604Ah	2	vl_velocity_ quick_stop_ delta_time	U16	1	RW	RPDO	s	1	100	2	Yes

[15] This unit depends on the configuration of object 'velocity_unit' ($604C_h$).
[16] This unit depends on the configuration of object 'velocity_unit' ($604C_h$).
[17] This unit depends on the configuration of object 'velocity_unit' ($604C_h$).

Speed control

The 'speed control' block monitors the setpoint generated by the motor ramp block.

This block is based on a PID which aims to avoid speed error.

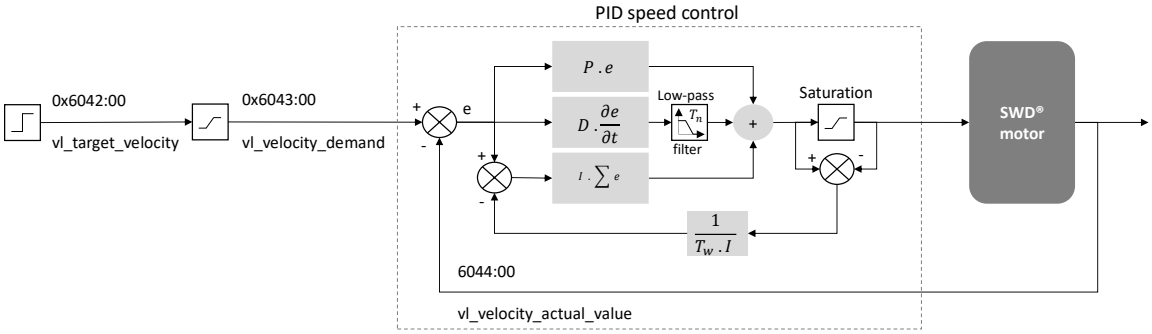


Figure 38 – Speed controller

The PID parameters are accessible in the CANopen dictionary and are updated when the CiA 402 state machine switches to the 'Operation Enable' state.

⚠ *Coefficients can be saved in non-volatile memory.*
⚠ *If a user configuration has been saved, it is used when the drive starts up.*

The error 'e' is expressed in the PID in mdeg.s^{-1} at the motor shaft before gearbox.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2155$_h$	0	motctrl_speed_pid_p	U32	-[18]	RW	NO	10^{-8}	0	4294967295	4
2156$_h$	0	motctrl_speed_pid_i	U32		RW	NO	10^{-8}	0	4294967295	4
2157$_h$	0	motctrl_speed_pid_d	U32		RW	NO	10^{-8}	0	4294967295	4
2158$_h$	0	motctrl_speed_pid_tw	U32		RW	NO	10^{-3}	0	4294967295	4
2159$_h$	0	motctrl_speed_pid_tn	U32		RW	NO	10^{-3}	0	4294967295	4

In addition to the PID, the controller incorporates an "anti-windup" action on the integral action, and a time filter on the derivative action, both of which can be set using the 'motctrl_speed_pid_tw' and 'motctrl_speed_pid_tn' parameters.

The minimum speed setpoint is 5 RPM.

[18] Default values differs according to the gearbox ratios of the ***SWD®*** products. You can refer directly to the product objects dictionary.

Gear ratio .4: *P = 10, I = 100, D = 0, Tw = 600, Tn = 316*
Gear ratio .14: *P = 150, I = 400, D = 50, Tw = 600, Tn = 316*

10.3. Configuration

The configuration data for the motor controller are considered when the 'Operation enable' state is reached.

Modifications in the dictionary are not possible in 'Operation enable', 'Quick stop active' and 'Fault reaction active' states.

6007h Abort connection option code

You can define the action to be applied if the CANopen connection is lost. The ***SWD®*** considers a loss of the CAN bus in the following cases:

- Bus-off
- ***SWD®*** in *NMT STOPPED and* in *NMT INITIALISATION*

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
6007h	0	Abort connection option code	I16	2	RO[19]	NO	Table	0	3	2

Value	Definition
-32 768 to -1	Manufacturer-specific
0	No action
+1	Fault signal
+2	Disable voltage command
+3	Quick stop command
+4 to +32 767	Reserved

605Ah Quick stop option code

You can define the action to be applied when a QuickStop command is received.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
605Ah	0	Quick stop option code	I16	6	RO[20]	NO	Table	0	6	2

Value	Definition
+1	Slow down on slow down ramp and transit into switch on disabled
+2	Slow down on quick stop ramp and transit into switch on disabled
+5	Slow down on slow down ramp and stay in quick stop active
+6	Slow down on quick stop ramp and stay in quick stop active

[19] Configuration under development
[20] Configuration under development

605B$_h$ Shutdown option code

You can define the action to be applied when a QuickStop command is received.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
605B$_h$	0	shutdown option code	I16	0	RO[21]	NO	Table	0	2	2	Yes

Value	Definition
0	Disable drive function (switch-off the drive power stage)
+1	Slow down with slow down ramp; disable of the drive function

605C$_h$ Disable operation option code

It is possible to define the action applied at transition 8: from 'operation enable' to 'switched on state'.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
605B$_h$	0	shutdown option code	I16	0	RO[22]	NO	Table	0	1	2	Yes

Value	Definition
0	Disable drive function (switch-off the drive power stage)
+1	Slow down with slow down ramp; disable of the drive function

605D$_h$ Halt option code

You can define the action to be taken when a 'Halt' command is received.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
605D$_h$	0	Halt option code	I16	1	RO[23]	NO	Table	1	2	2

Value	Definition
+1	Slow down on slow down ramp and stay in operation enabled
+2	Slow down on quick stop ramp and stay in operation enabled

605E$_h$ Fault reaction option code

You can define the action to be applied when a PDS (Power Drive System) internal error is detected.

[21] Configuration under development
[22] Configuration under development
[23] Configuration under development

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
605E$_h$	0	fault_reaction_option_code	I16	0	RO[24]	NO	Table	-	-	2

Value	Definition
0	Disable drive function, motor is free to rotate
+1	Slow down on slow down ramp
+2	Slow down on quick stop ramp

[24] Configuration under development

11. Safety functions

11.1. *SWD®* Safety functions

The ***SWD®*** includes safe motor shutdown functions, as well as advanced safe speed supervision functions. This exclusive integrated solution simplifies machine architecture by removing the need for external supervision devices and safety logic.

The motor stop functions are as follows:

- STO (Safe Torque Off): safe disconnection of motor torque
- SBC[25] (Safe Brake Control): safe braking engagement
- SBU[26] (Safe Brake Unlock): freewheel mode, safe brake disengagement

Speed supervision functions are based on a safety encoder integrated into the ***SWD®***. They are listed below:

- SDI (Safe Direction): safe control of direction of rotation
- SLS (Safely Limited Speed): safe speed limitation
- SLSa[27] (Safely Limited Speed asymmetric): safe speed limitation, with different thresholds depending on the direction of the motor's rotation.
- SMS[28] (Safe Maximum Speed): Maximum speed limitation independent of activated safety functions.

These safety functions are implemented in accordance with IEC 61800-5-2 (functional safety requirements for the development of safety drives) and CiA/DS 402-4 (safety functions for motor controllers).

	Safety functions	Possible settings
STO	Safe disconnection of motor torque	Rearm: ▪ automatic ▪ by request acknowledgment STO associated with braking, see SBC configuration
SBC	Safe braking engagement	Joint activation with STO: ▪ Internal ***SWD®*** braking ▪ External electromechanical brake ▪ Both brakes simultaneously
SBU	Safe braking disengagement	-
SDI	Reliable control of direction of rotation	Direction of rotation prohibited: ▪ in positive or negative direction ▪ with a tolerance threshold

[25] Implemented from 'Firmware' (2.0.x)
[26] Implemented from 'Firmware' (2.0.x)
[27] Implemented from 'Firmware' (2.0.x)
[28] Implemented from 'Firmware' (2.0.x)

SLS	Safe speed limitation	Overtaking speed: ▪ 8 switchable thresholds ▪ adjustable triggering time
SLSa	Safe speed limitation, in one direction only	Overtaking speed, in one direction only: ▪ 8 switchable thresholds ▪ adjustable triggering time
SMS	Safe limitation of maximum authorized speed, regardless of activated safety functions	Overtaking speed, regardless of safety functions triggered ▪ a threshold limit for positive maximum speeds ▪ a threshold limit for negative maximum speeds

11.2. Typical use of the safety functions

SWD® safety functions are designed for speed-controlled load movement applications where risk assessment justifies the implementation of safe movement supervision.

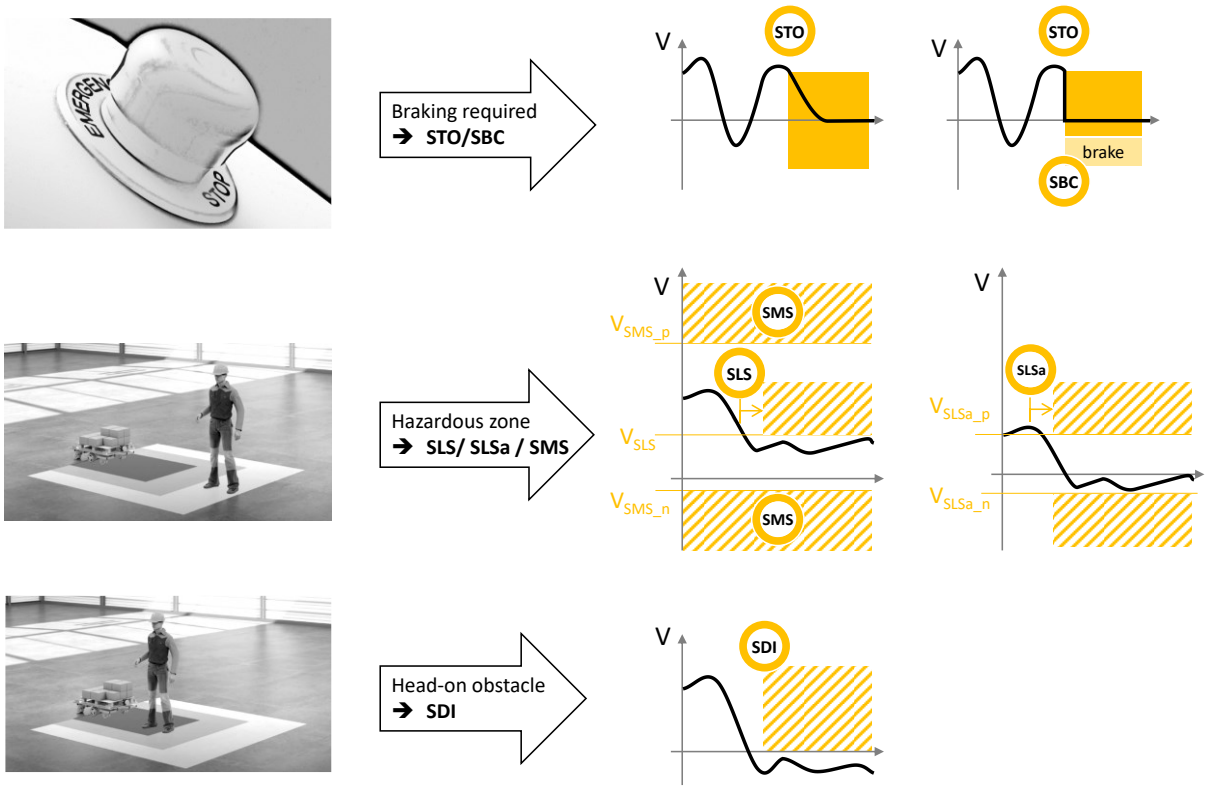


Figure 39 – Use of the safety functions

11.3. Safety level summary

	ISO 13849-1:2015			EN IEC 61508[29]				IEC 62061[30]	IEC 61800-5-2[31]	NF EN 60204-1[32]
Safety function	**Category**	**PL**	**PFH_D /h**	**SIL**	**PFH /h**	**PFD_{AVG} / Year**	**SFF**	**SIL CL**	**SIL**	**Category**
STOP safety functions										
Safe Torque Off (STO) STO1 et STO2 inputs	Category 4	PL e	1,42E-9	SIL 3	8,9E-9	3,9E-5	100%	SIL3	SIL3	Category 0
Safe Torque Off (STO) CANopen Safety	Category 2	PL d	2,29E-7	SIL 2	8,9E-9	3,9E-5	99,7%	SIL 2	SIL 2	Category 0
Safety Safe Torque Off (STO) SafeInput	Category 2	PL d	2,29E-7	SIL 2	10,3E-9	4,5E-5	99,6%	SIL 2	SIL 2	Category 0
Safety Safe Torque Off (STO) Pair of SafeInput	Category 3	PL d	4,29E-8	SIL 2	11,6E-9	5,1E-5	99,6%	SIL 2	SIL 2	Category 0
Rearm of STO SafeInput	Category 2	PL d	2,29E-7	SIL 2	10,3E-9	4,5E-5	99,6%	SIL 2	SIL 2	Category 0
Rearm of STO CANopen safety	Category 2	PL d	2,29E-7	SIL 2	8,9E-9	3,9E-5	99,7%	SIL 2	SIL 2	Category 0
Monitoring safety functions										
Safe Maximum speed (SMS)	Category 2	PL d	2,29E-7	SIL 2	8,9E-9	3,9E-5	99,7%	SIL 2	SIL 2	NA
Safely limited speed (SLS) CANopen Safety	Category 2	PL d	2,29E-7	SIL 2	8,9E-9	3,9E-5	99,7%	SIL 2	SIL 2	NA
Safely limited speed (SLS) SafeInput	Category 2	PL d	2,29E-7	SIL 2	10,3E-9	4,5E-5	99,6%	SIL 2	SIL 2	NA
Safely limited speed (SLS) Pair of SafeInput	Category 3	PL d	2,29E-7	SIL 2	11,6E-9	5,1E-5	99,6%	SIL 2	SIL 2	NA
Safely limited speed asymmetric (SLSa) CANopen Safety	Category 2	PL d	2,29E-7	SIL 2	8,9E-9	3,9E-5	99,7%	SIL 2	SIL 2	NA
Safely limited speed asymmetric (SLSa) SafeInput	Category 2	PL d	2,29E-7	SIL 2	10,3E-9	4,5E-5	99,6%	SIL 2	SIL 2	NA

[29] EN IEC 61508: December 2001 and April 2010

[30] NF EN 62061 (July 2005) + NF EN 62061/A1 (2013-05-10) + NF EN 62061/A2 (2015-12-25)

[31] IEC 61800 part 5-2 ed1 2007 and ed2 2016: Adjustable speed electrical power drive systems

[32] NF EN 60204-1:2018 “Safety of machinery - Electrical equipment of machines - Part 1: General requirements “ - IEC 60204-1:2016, modified

Safely limited speed asymmetric (SLSa) Pair of SafeInput	Category 3	PL d	2,29E-7	SIL 2	11,6E-9	5,1E-5	99,6%	SIL 2	SIL 2	NA
Safe direction (SDI) CANopen Safety	Category 2	PL d	2,29E-7	SIL 2	8,9E-9	3,9E-5	99,7%	SIL 2	SIL 2	NA
Safe direction (SDI) SafeInput	Category 2	PL d	2,29E-7	SIL 2	10,3E-9	4,5E-5	99,6%	SIL 2	SIL 2	NA
Safe direction (SDI) Pair of SafeInput	Category 3	PL d	2,29E-7	SIL 2	11,6E-9	5,1E-5	99,6%	SIL 2	SIL 2	NA
Safe speed	Category 3	PL d	4,29E-8	SIL2	8,9E-9	3,9E-5	99,6 %	SIL 2	SIL 2	NA
Output safety functions										
Safe Brake Control (SBC2) external STO1 et STO2 inputs	Category 2	PL d	2,29E-7	SIL 3	9,2E-9	4,0E-5	99,7%	SIL 3	SIL 3	Category 0
Safe Brake Control (SBC2) CANopen Safety	Category 2	PL d	2,29E-7	SIL 2	8,9E-9	3,9E-5	99,7%	SIL 2	SIL 2	Category 0
Safe Brake Control (SBC2) SafeInput	Category 2	PL d	2,29E-7	SIL 2	10,3E-9	4,5E-5	99,6%	SIL 2	SIL 2	Category 0
Safe Brake Control (SBC2) Pair of SafeInput	Category 2	PL d	2,29E-7	SIL 2	11,6E-9	5,1E-5	99,6%	SIL 2	SIL 2	Category 0
Safe Brake Control (SBC3) internal STO1 et STO2 inputs	Category 2	PL d	2,29E-7	SIL 3	10,8E-9	4,7E-5	99,6%	SIL 3	SIL 3	Category 0
Safe Brake Control (SBC3) CANopen Safety	Category 2	PL d	2,29E-7	SIL 2	10,5E-9	4,6E-5	99,6%	SIL 2	SIL 2	Category 0
Safe Brake Control (SBC3) SafeInput	Category 2	PL d	2,29E-7	SIL 2	11,8E-9	5,2E-5	99,6%	SIL 2	SIL 2	Category 0
Safe Brake Control (SBC3) Pair of SafeInput	Category 2	PL d	2,29E-7	SIL 2	13,2E-9	5,8E-5	99,5%	SIL 2	SIL 2	Category 0

STO functions are certified for **category 0** shutdown in compliance with standard **EN 60204-1.**

This mode corresponds to disconnection of the motor power supply for the STO function, which is then in freewheel mode.

A 14-year $MTTF_D$ is used for ***SWD®-Core & SWD®-125***

A 12-year $MTTF_D$ is used for ***SWD®-150.***

11.4. Activating a safety function

Safety functions can be triggered by:

- A safety input on the ***SWD®*** I/O connector (SafeIN).
- A safety controlword from the CANopen dictionary object (Safety controlwords).
- In reaction to the detection of an internal ***SWD®*** error.

ℹ The control status of a safety function is read-only in the CANopen dictionary and is processed by the ***SWD®***.

Types of sensors enabling activation

Sensors		Description
Electro-mechanical switch/safety switch	EMSS	Electromechanical switches with equivalent double contacts / safety switches without signal processing element
Safety sensors with monitored semiconductor output	OSSD	Safety sensors with dual-channel semiconductor outputs and cross-circuit monitoring
Safety sensors with test input	SCSD	Safety sensors with test input, signal processing and test output

Safety functions configuration

Safety controlwords' are used to control a safety function from CANopen Safety. Safety status words are used to retrieve the status of a safety function from CANopen Safety.

It is necessary to map the desired safety functions to these 'Safety words'. Commands must be mapped in 'scw_mapping_' and 'status' in 'ssw_mapping'.

The list of 'commands' and 'status' that can be mapped are:

Safety Function	Code Control	Code Status
RESTART_ACK	6630 0000$_h$	6631 0000$_h$
ERROR_ACK	6632 0000$_h$	6633 0000$_h$
STO	6640 0000$_h$	6644 0000$_h$
SBC_1	6660 0100$_h$	6667 0100$_h$
SBC_2	6660 0200$_h$	6667 0200$_h$
SBC_3	6660 0300$_h$	6667 0300$_h$
SBU	3040 0000$_h$	3041 0000$_h$
SMSp	66AA 0100$_h$	66A8 0100$_h$
SMSn	66AC 0100$_h$	66A8 0100$_h$
SLS [1]	6690 0100$_h$	669F 0100$_h$
SLS [2]	6690 0200$_h$	669F 0200$_h$
SLS [3]	6690 0300$_h$	669F 0300$_h$
SLS [4]	6690 0400$_h$	669F 0400$_h$
SLS [5]	6690 0500$_h$	669F 0500$_h$
SLS [6]	6690 0600$_h$	669F 0600$_h$
SLS [7]	6690 0700$_h$	669F 0700$_h$
SLS [8]	6690 0800$_h$	669F 0800$_h$
SLSa [1]	3050 0100$_h$	3059 0100$_h$

SLSa [2]	3050 0200$_h$	3059 0200$_h$
SLSa [3]	3050 0300$_h$	3059 0300$_h$
SLSa [4]	3050 0400$_h$	3059 0400$_h$
SLSa [5]	3050 0500$_h$	3059 0500$_h$
SLSa [6]	3050 0600$_h$	3059 0600$_h$
SLSa [7]	3050 0700$_h$	3059 0700$_h$
SLSa [8]	3050 0800$_h$	3059 0800$_h$
SDIp [1]	66D0 0100$_h$	66DE 0100$_h$
SDIp [2]	66D0 0200$_h$	66DE 0200$_h$
SDIn [1]	66D1 0100$_h$	66DF 0100$_h$
SDIn [2]	66D1 0200$_h$	66DF 0200$_h$

Activation with CANopen Safety

ℹ *To update the value of controlwords in a security context, it is necessary to use SRDO safety messages.*

To activate a safety function via CANopen, you need to use a safety control word(s). Each Safety Controlword can control up to 8 safety functions, with each bit corresponding to the activation of a command. It is also possible to retrieve the status of a safety function by using safety Statusword(s). Each Safety Statusword can be used to retrieve up to 8 safety function states, with each component bit corresponding to the activation state of a function.

Each Safety Controlword bit is associated with a command by means of a mapping configuration operation.

Each bit of the Safety Statusword is associated with a status by means of a configuration operation in this mapping.

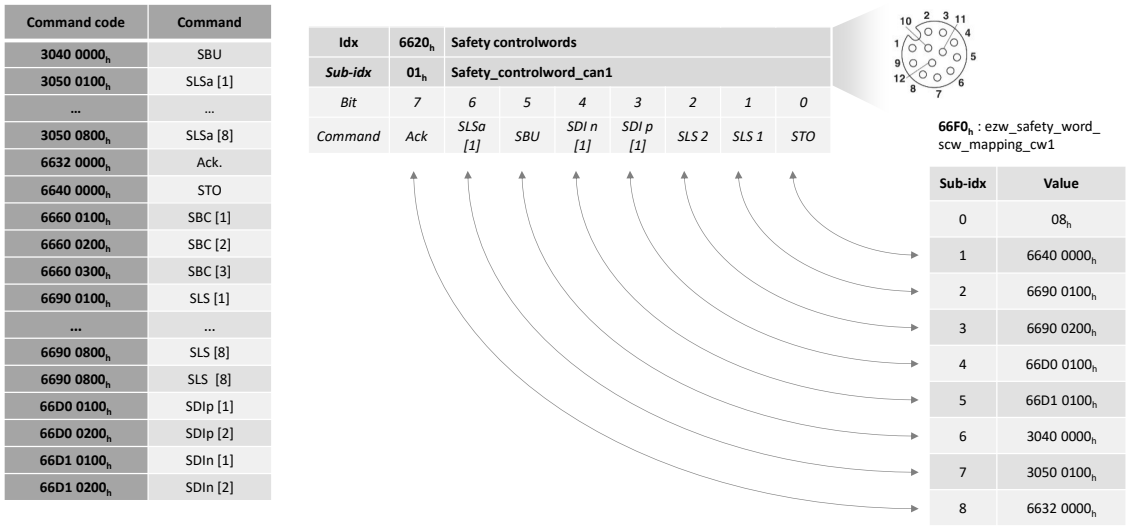


Figure 40 – Example of safety command configuration via CANopen

Activation with safety inputs

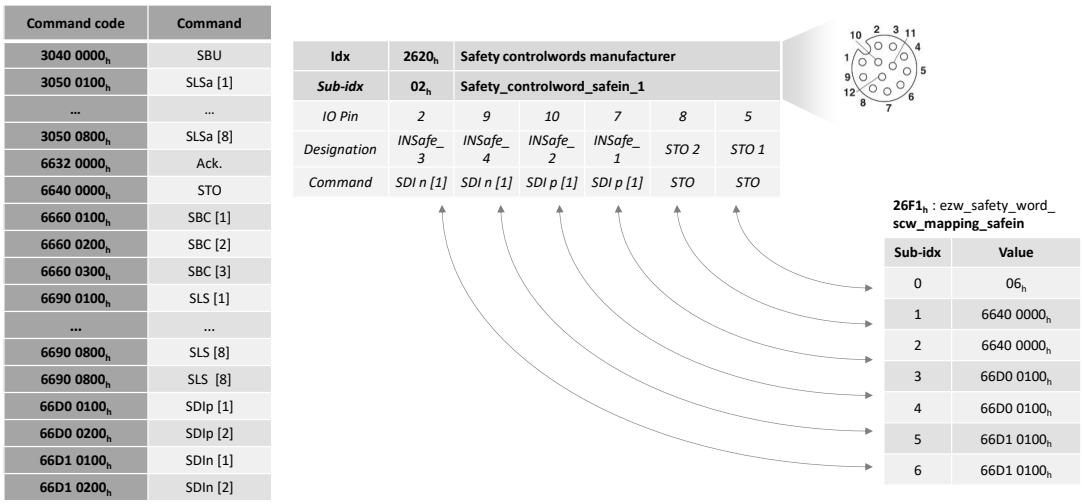


Figure 41 – Example of SafeInputs configuration of safety commands

Safety inputs can operate in pairs to achieve SIL2/PLd/Cat3 safety levels. The configuration must be carried out in such a way as to account for joint operation by associating the same safety function with the commands of both inputs.

The inputs are grouped as follows:

- INSafe_1 – INSafe_2
- INSafe_3 – INSafe_4

Permanent activation

It is possible to permanently activate a safety function. To do this, map the safety function you wish to activate permanently to one of the two sub-objects, for example 0x66400000 for STO function:

2624h ‘ezw_safety_word_scw_mapping_permanent_cw1’:

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2624h	00h	Number of entries	U8	08h	RO	NO	-	-	-	1
2624h	01h	scw_cw_permanent_can_1_bit0	U32	0000 0000h	RW	NO	-	-	-	4
2624h	02h	scw_cw_permanent_can_1_bit1	U32	0000 0000h	RW	NO	-	-	-	4
2624h	03h	scw_cw_permanent_can_1_bit2	U32	0000 0000h	RW	NO	-	-	-	4
2624h	04h	scw_cw_permanent_can_1_bit3	U32	0000 0000h	RW	NO	-	-	-	4
2624h	05h	scw_cw_permanent_can_1_bit4	U32	0000 0000h	RW	NO	-	-	-	4
2624h	06h	scw_cw_permanent_can_1_bit5	U32	0000 0000h	RW	NO	-	-	-	4
2624h	07h	scw_cw_permanent_can_1_bit6	U32	0000 0000h	RW	NO	-	-	-	4
2624h	08h	scw_cw_permanent_can_1_bit7	U32	0000 0000h	RW	NO	-	-	-	4

2625_h 'ezw_safety_word_scw_mapping_permanent_cw2':

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2625_h	00_h	Number of entries	U8	08_h	RO	NO	-	-	-	1
2625_h	01_h	scw_cw_permanent_can_2_bit0	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
2625_h	02_h	scw_cw_permanent_can_2_bit1	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
2625_h	03_h	scw_cw_permanent_can_2_bit2	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
2625_h	04_h	scw_cw_permanent_can_2_bit3	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
2625_h	05_h	scw_cw_permanent_can_2_bit4	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
2625_h	06_h	scw_cw_permanent_can_2_bit5	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
2625_h	07_h	scw_cw_permanent_can_2_bit6	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
2625_h	08_h	scw_cw_permanent_can_2_bit7	U32	$0000\ 0000_h$	RW	NO	-	-	-	4

Default 'Safety words' mappings

Safety Controlword 1

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
$66F0_h$	00_h	Number of entries	U8	08_h	RO	NO	-	-	-	1
$66F0_h$	01_h	Command 1	U32	$6640\ 0000_h$ STO command	RO	NO	-	-	-	4
$66F0_h$	02_h	Command 2	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F0_h$	03_h	Command 3	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F0_h$	04_h	Command 4	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F0_h$	05_h	Command 5	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F0_h$	06_h	Command 6	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F0_h$	07_h	Command 7	U32	$6630\ 0000_h$ RESTART_ACK	RO	NO	-	-	-	4
$66F0_h$	08_h	Command 8	U32	$6632\ 0000_h$ ERROR_ACK	RO	NO	-	-	-	4

Safety Controlword 2 to 8

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
$66F1_h$ to $66F8_h$	00_h	Number of entries	U8	08_h	RO	NO	-	-	-	1
$66F1_h$ to $66F8_h$	01_h	Command 1	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F1_h$ to $66F8_h$	02_h	Command 2	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F1_h$ to $66F8_h$	03_h	Command 3	U32	$0000\ 0000_h$	RW	NO	-	-	-	4

66F1$_h$ to 66F8$_h$	04$_h$	Command 4	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F1$_h$ to 66F8$_h$	05$_h$	Command 5	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F1$_h$ to 66F8$_h$	06$_h$	Command 6	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F1$_h$ to 66F8$_h$	07$_h$	Command 7	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F1$_h$ to 66F8$_h$	08$_h$	Command 8	U32	0000 0000$_h$	RW	NO	-	-	-	4

Safety Controlword Safein 1

In its default configuration, the safety inputs on the connector are not associated with a control. Configuration must be carried out by the machine manufacturer according to the architecture and safety functions to be assigned.

STO_1 and STO_2 inputs are read-only and cannot be modified by configuration. These connector inputs are always associated with the STO function.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2620$_h$	00$_h$	Number of entries	U8	06$_h$	RO	NO	-	-	-	1
2620$_h$	01$_h$	Command STO_1	U32	6640 0000$_h$ STO command	RO	NO	-	-	-	4
2620$_h$	02$_h$	Command STO_2	U32	6640 0000$_h$ STO command h	RO	NO	-	-	-	4
2620$_h$	03$_h$	Command INSafe_1	U32	0000 0000$_h$	RW	NO	-	-	-	4
2620$_h$	04$_h$	Command INSafe_2	U32	0000 0000$_h$	RW	NO	-	-	-	4
2620$_h$	05$_h$	Command INSafe_3	U32	0000 0000$_h$	RW	NO	-	-	-	4
2620$_h$	06$_h$	Command INSafe_4	U32	0000 0000$_h$	RW	NO	-	-	-	4

Safety Statusword 1

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
66F8$_h$	00$_h$	Number of entries	U8	08$_h$	RO	NO	-	-	-	1
66F8$_h$	01$_h$	Status 1	U32	6644 0000$_h$ STO status	RO	NO	-	-	-	4
66F8$_h$	02$_h$	Status 2	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F8$_h$	03$_h$	Status 3	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F8$_h$	04$_h$	Status 4	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F8$_h$	05$_h$	Status 5	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F8$_h$	06$_h$	Status 6	U32	0000 0000$_h$	RW	NO	-	-	-	4
66F8$_h$	07$_h$	Status 7	U32	6631 0000$_h$ RESTART_ACK	RO	NO	-	-	-	4
66F8$_h$	08$_h$	Status 8	U32	6633 0000$_h$ ERROR_ACK	RO	NO	-	-	-	4

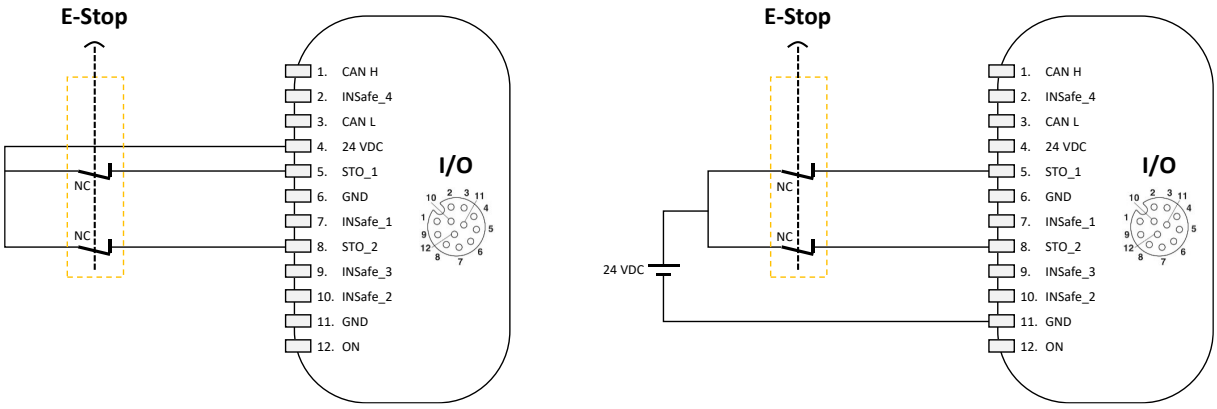
Safety Statusword 2 to 8

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
$66F9_h$ to $66FF_h$	00_h	Number of entries	U8	08_h	RO	NO	-	-	-	1
$66F9_h$ to $66FF_h$	01_h	Status 1	U32	$6644\ 0000_h$ STO status	RW	NO	-	-	-	4
$66F9_h$ to $66FF_h$	02_h	Status 2	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F9_h$ to $66FF_h$	03_h	Status 3	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F9_h$ to $66FF_h$	04_h	Status 4	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F9_h$ to $66FF_h$	05_h	Status 5	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F9_h$ to $66FF_h$	06_h	Status 6	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F9_h$ to $66FF_h$	07_h	Status 7	U32	$0000\ 0000_h$	RW	NO	-	-	-	4
$66F9_h$ to $66FF_h$	08_h	Status 8	U32	$6633\ 0000_h$ Error status	RW	NO	-	-	-	4

11.5. Recommendations for implementation

STO activation by emergency stop

STO activation by emergency stop via I/O connector interface, connected to input pair STO_1 and STO_2; SIL3 / PLe / Cat4 compliant.

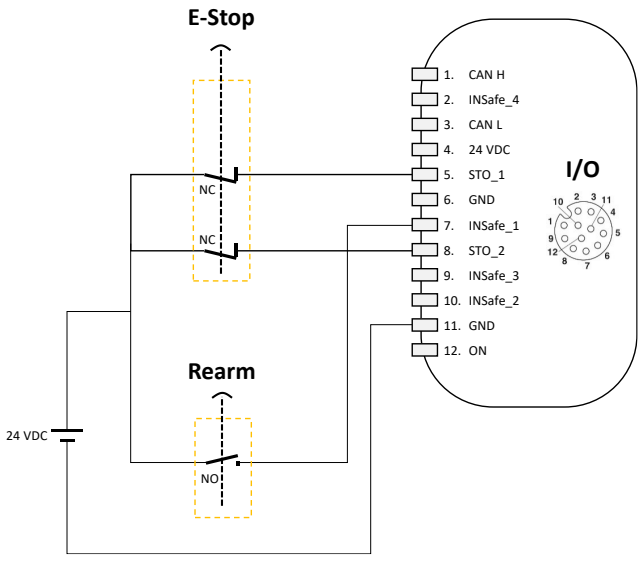


A switch conforming to ISO 13850, with positive opening as in accordance with IEC 60947-5-1, or a certified safety controller must be used as the actuating element.

The input safety component must be chosen to satisfy the overall level of the safety function.

STO rearmament

STO reset by pressing a normally open (NO) button via the I/O connector interface, connected to the INSafe_1 input; holds for up to 100 hours.



The single input INSafe_1 is configured as a reset input after STO engagement. A rising edge from 0 to 24 VDC on the INSafe_1 input resets the STO State to 0 and restores torque to the motor. Object 6641h 'STO restart acknowledge behavior' must be activated (value set to 1), and safety input 'INSafe 1' configured to STO restart acknowledge command 6630_h.

The input safety component must be chosen to satisfy the overall level of the safety function.

The safety function is guaranteed to be maintained for a maximum of 100 hours.

STO activation and reset with a safety reset relay

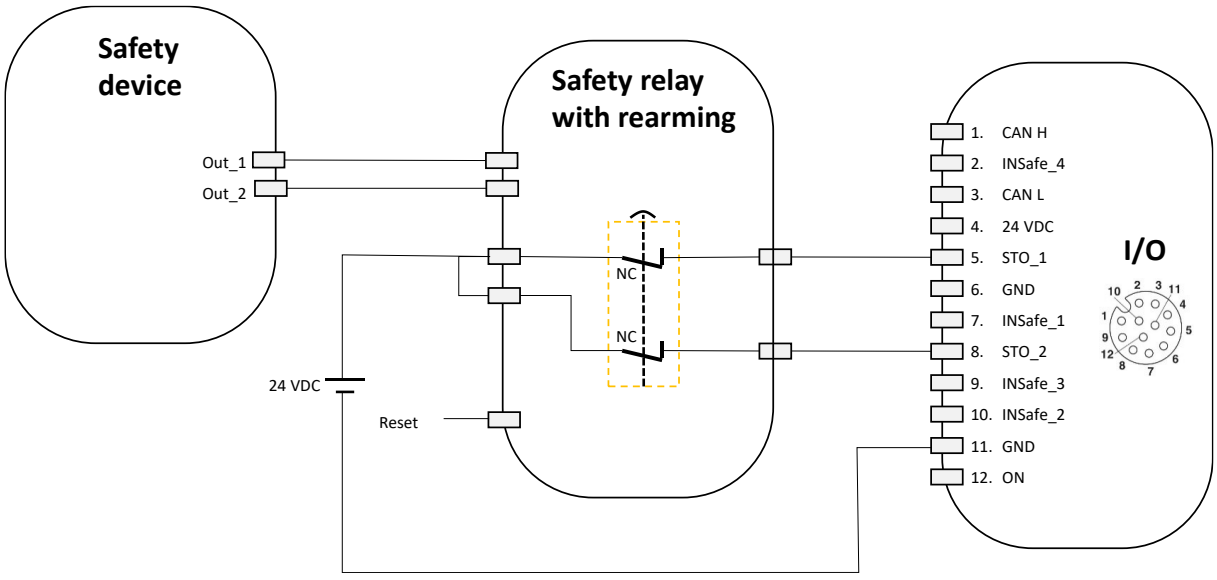
STO activation with compliance up to SIL3 / Ple, with reset, maintained beyond 100h using a safety relay.

For Machinery Directive regulations in Europe, the additional requirements of EN ISO 14118:2018 for non-restart must be considered.

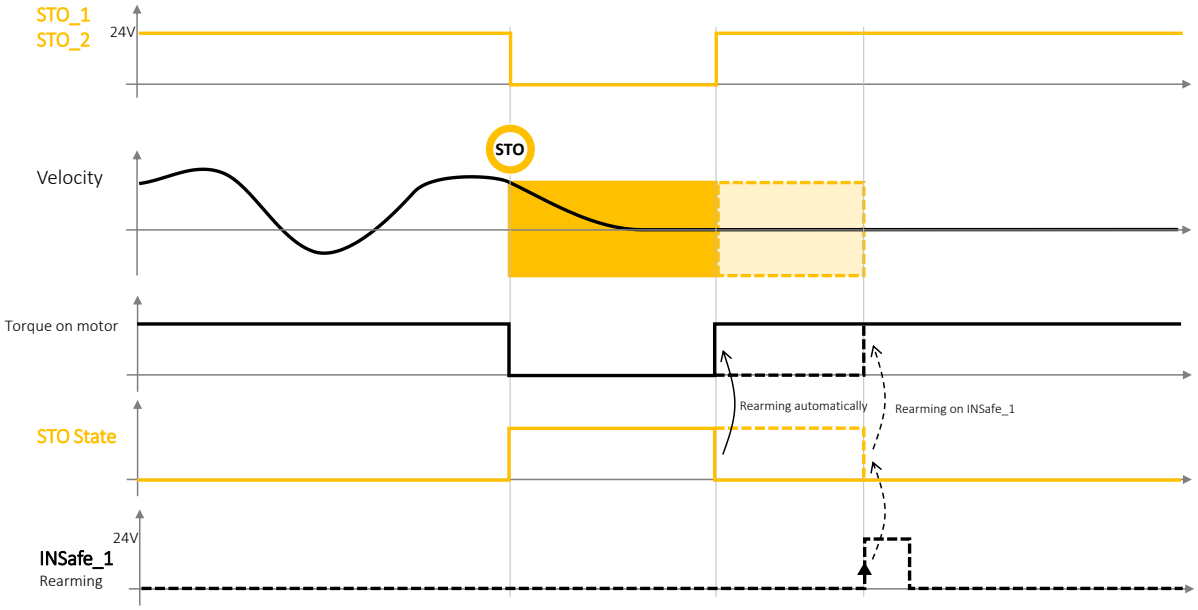
The performance levels of the safety functions and the STO position guarantee are guaranteed for a period of 100 hours.

In the event of a power cut, this time interval is guaranteed, because in normal operation, the drive must be restarted after a physical action by the user.

After this period of 100 hours, the drive can accumulate failures and, to avoid a dangerous situation, the addition of a safety relay is necessary to comply with EN ISO 14118:2018 for non-restart requirements.

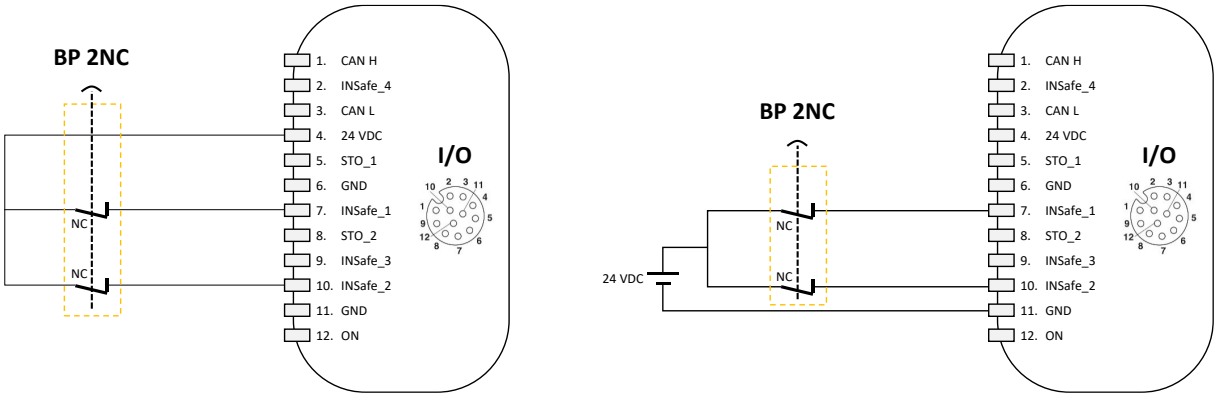


The input safety component must be chosen to satisfy the overall level of the safety function.



Activation of a safety function via a safety input pair

Activation of a software safety function with a 2-pole normally closed (2NC) switch; compliance up to SIL2 / PLd / Cat3.



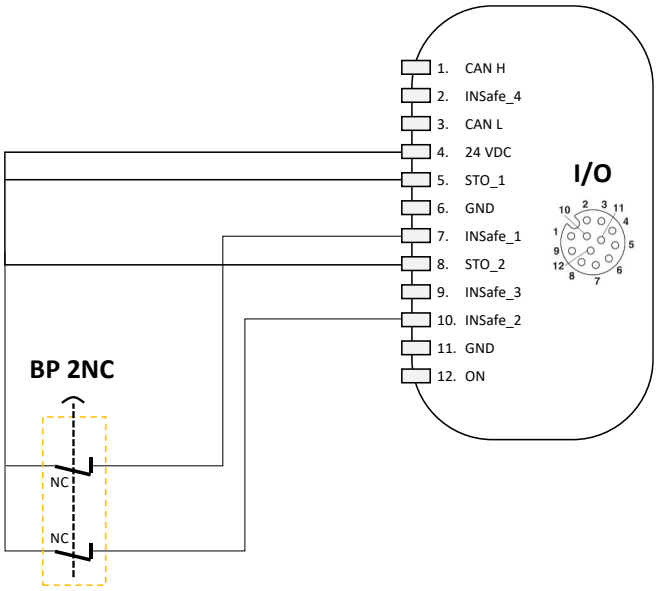
A switch according to ISO 13850, positive opening according to IEC 60947-5-1, or a certified safety controller must be used as the actuating element.

The input safety component must be chosen to satisfy the overall level of the safety function.

If a software safety function is activated by the I/O connector interface, the associated safety function must be configured for each safety inputs.

Permanent deactivation of STO and activation of a safety function

Permanent deactivation of STO and activation of another safety function by a 2-pole normally closed (2NC) switch, compliance up to SIL2 / PLd.

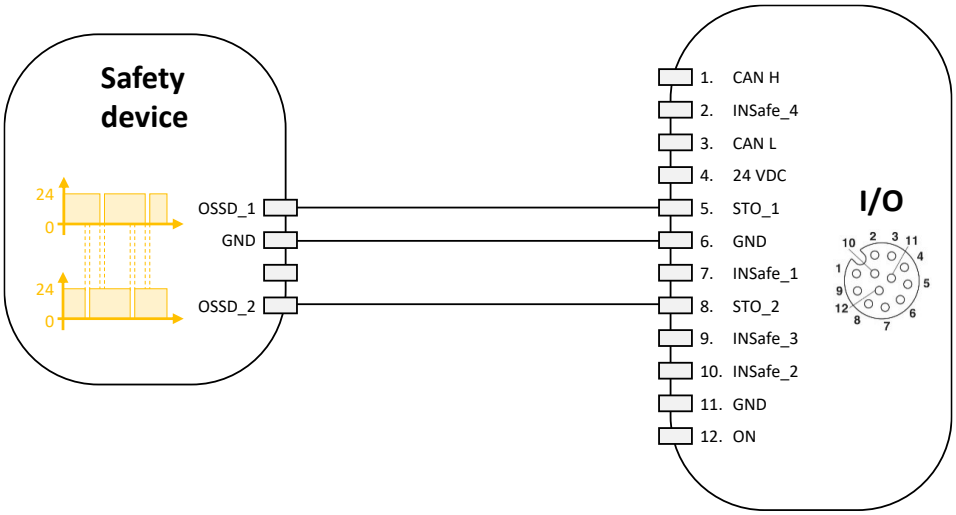


A switch according to ISO 13850, positive opening according to IEC 60947-5-1, or a certified safety controller must be used as the actuating element.

The input safety component must be chosen to satisfy the overall level of the safety function.

Activation of STO by OSSD outputs

Activation of STO by OSSD outputs, compliance up to SIL3 / PLe.

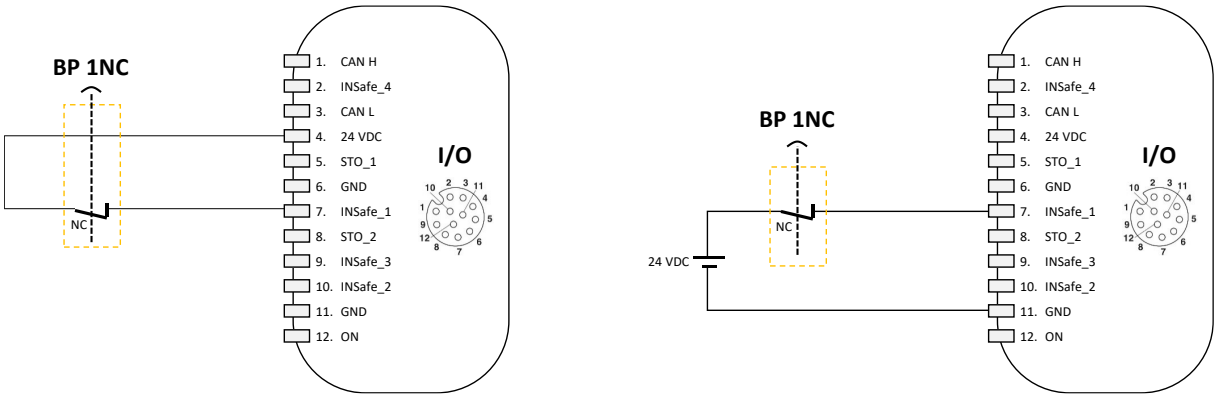


A certified sensor with a pair of OSSDs, a sensor, a safety controller.

The input safety component must be chosen to satisfy the overall level of the safety function.

Use of a single security entrance

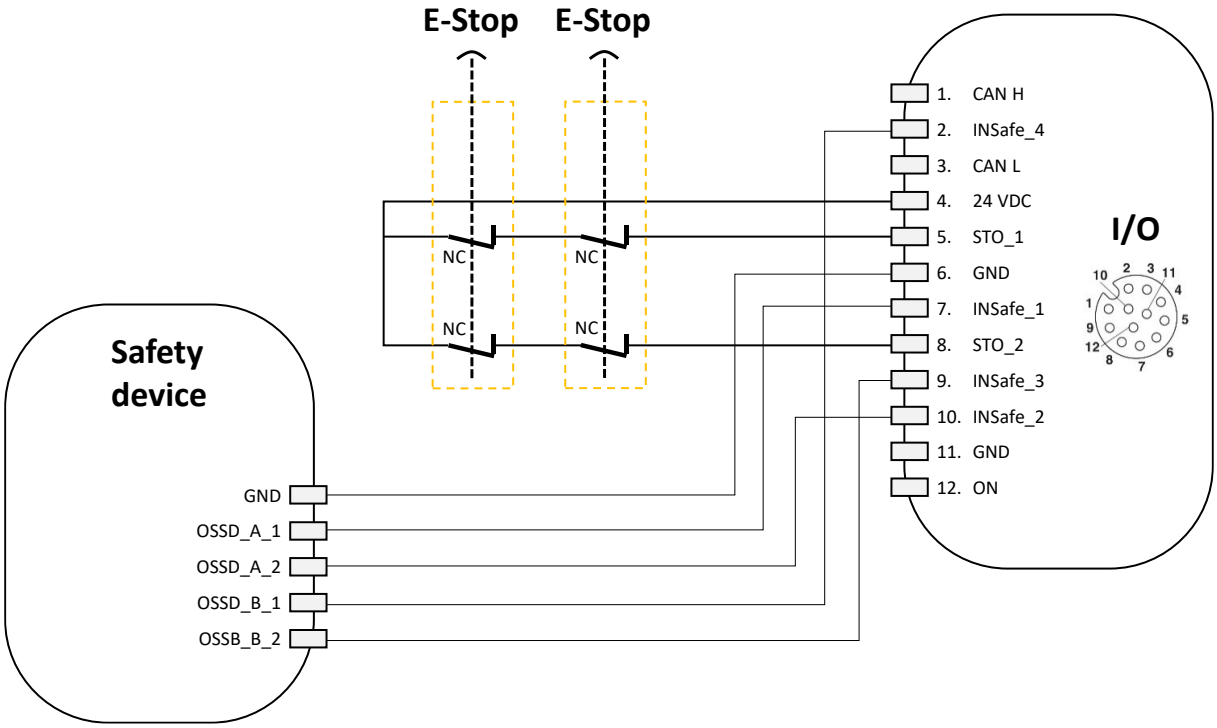
Use of a simple safety input; compliance up to SIL2 / PLd / Cat2.



The input safety component must be chosen to satisfy the overall level of the safety function.

Activation of STO by two emergency stops

Activation of STO by two emergency stops; compliance up to SIL3 / Ple, and activation of two other functions by OSSD signals, compliance up to SIL2 / PLd



The input safety components must be chosen to satisfy the overall level of the safety function.

Exchanges of safety input status on the CANopen Safety bus

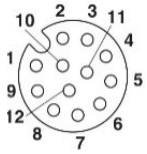
Information on the state of the safety inputs is exchanged on the CANopen Safety bus. So that any device connected to the CANopen bus can know the state of the safety inputs. This information is exchanged through a safety message on the CANopen Safety bus, called SRDO. This SRDO message contains a security object called 'Safe Control Word'. The ***SWD®*** on which the safety inputs are wired is, in this case of use, the transmitter of the SRDO containing the state of these safety inputs. The other ***SWD®*** consume the Safe Control Word contained in this SRDO. Thus, they can react to the state of a safety input of another peripheral, with a safety level SIL2 / PLd.

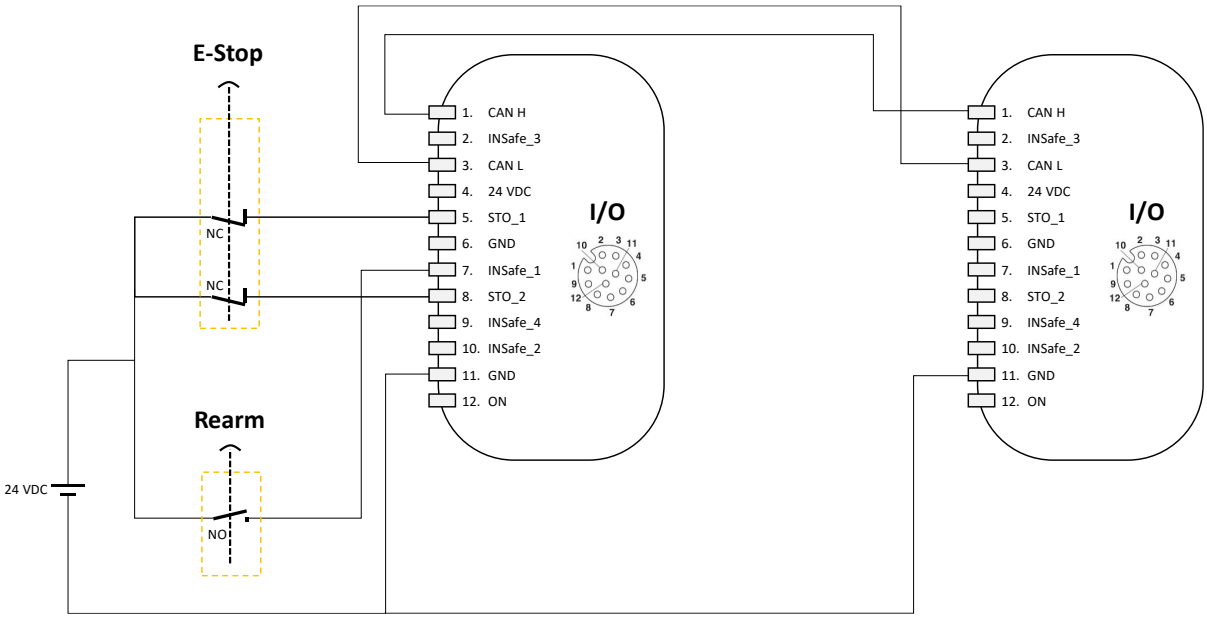
The configuration of the safety function commands must be carried out on each of the ***SWD®***.

It is carried out by configuring the **safeIn control word** on which the safety inputs are wired and by configuring the **safe control word 'n'** received by CANopen safety on the other ***SWD®***.

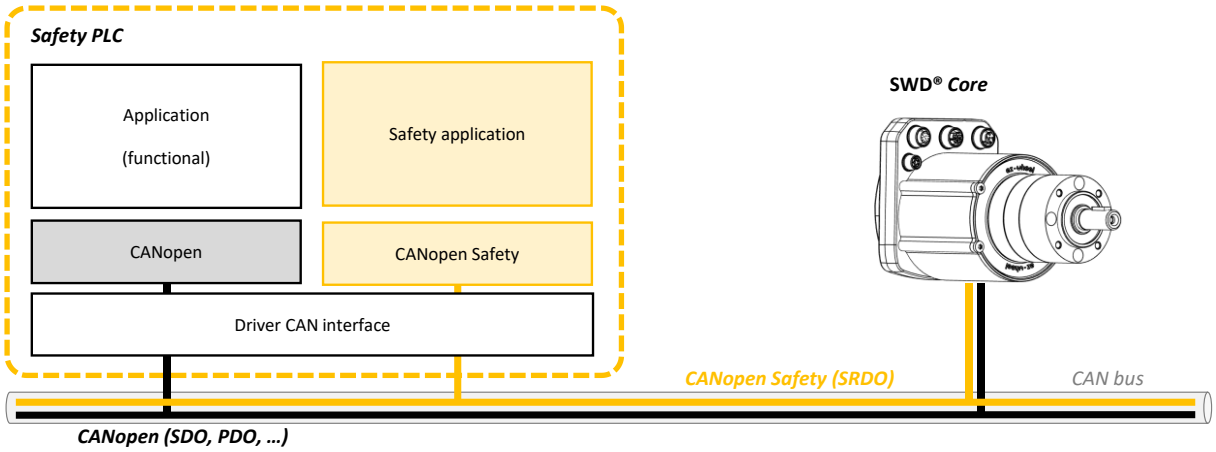
2620 00_h : safe_in controlword

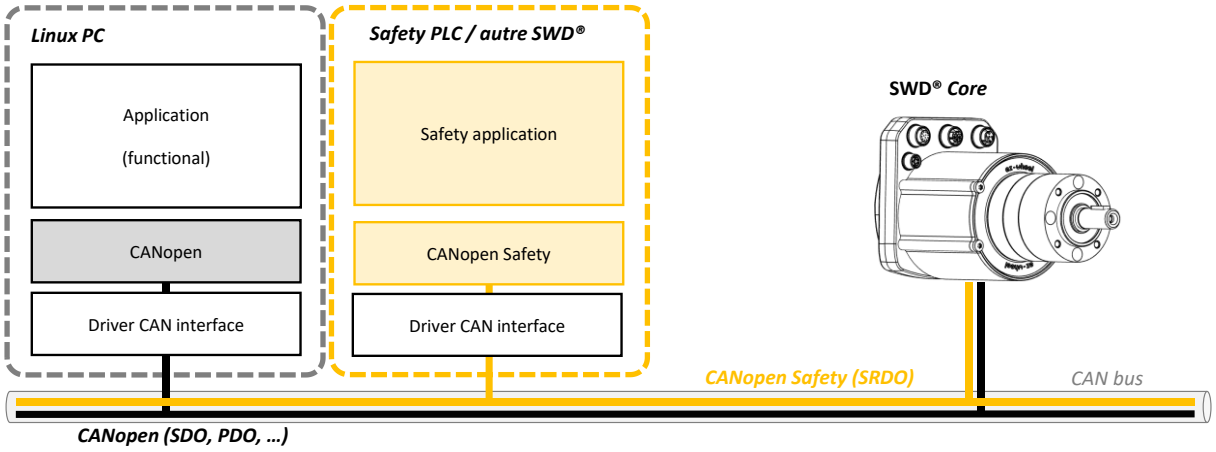
PIN#			*2*	*9*	*10*	*7*	*8*	*5*
Sig.			Safe IN 4	Safe IN 3	Safe IN 2	Safe IN 1	STO 2	STO 1
	-	-	**SDI n [1]**	**SDI n [1]**	**SDI p [1]**	**SDI p [1]**	**STO**	**STO**





Connection to a CANopen safety controller





11.6. States of safety functions

The status of the activated safety function commands are available in the 'Safety Function Output' object (2630_h). This read-only object is configurable in TxPDO. It contains the following two sub-indexes:

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
Safety Function Output										
2630_h	00_h	Number of entries	U8	02_h	RO	NO	-	02_h	-	1
	01_h	safety_function_output_1	U16	-	RO	TPDO	-	0_h	$FFFF_h$	2
	02_h	safety_function_output_2	U16	-	RO	TPDO	-	0_h	$FFFF_h$	2

safety_function_output_1															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SDI_N_2	SDI_N_1	SDI_P_2	SDI_P_1	SLS_8	SLS_7	SLS_6	SLS_5	SLS_4	SLS_3	SLS_2	SLS_1	SBC_3	SBC_2	SBC_1	STO
MSB															LSB

safety_function_output_2															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	0
Not used				**SLSa_8**	**SLSa_7**	**SLSa_6**	**SLSa_5**	**SLSa_4**	**SLSa_3**	**SLSa_2**	**SLSa_1**	**SMS**	**SBU**	**RESTART_ ACK**	**ERROR_ ACK**
MSB															LSB

Safety function	Deactivated	Activated
Bit value	**0**	**1**

11.7. Safety function

STO Activation

STO is activated by command 0x6640, active low.

After engaging an STO, the motor is electrically disconnected and no longer exerts torque. The speed slows down, but in the case of activation of STO alone, the motor is not braked, and the speed deceleration depends on the inertia of the drive.

The STO state is available in the '0x6644 STO State' object. A high state means that it is active and remains active until reset.

Activating STO

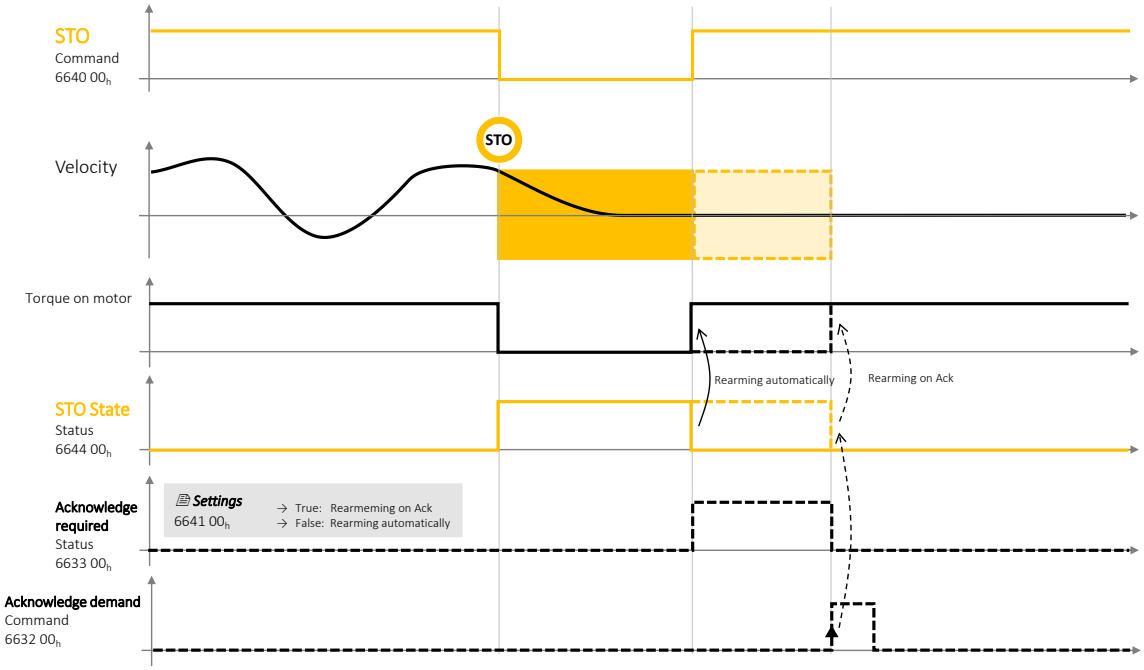


Figure 42 – STO activation and acknowledgment diagram

The restart mode is set by the '0x6641 STO restart acknowledge behavior' register. A 'False' value initiates automatic resetting on disappearance of the STO signal. A 'True' value requires an acknowledgment to be reset.

In the case of a reset by Acknowledge, the status is active in the high state as long as an acknowledgment is expected. The Acknowledge request must be made by switching the command to the high state, the rising edge triggering the acknowledgement.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
6630h	00	Restart acknowledge command	BOOLEAN	-	WO	NO	-	0	1	1
6631h	00	Restart acknowledge status	BOOLEAN	-	RO	NO	-	0	1	1
6640h	00	STO Command	BOOLEAN	-	RO	NO	-	0	1	1
6641h	00	STO restart acknowledge behavior	BOOLEAN	0	RW 🖹	NO	-	0	1	1
6643h	00	STO active SBC	U32	0x66600300	RW 🖹	NO	-	0000 0000h 6660 0100h 6660 0200h 6660 0300h		4
6644h	00	STO status	BOOLEAN	0	RO	NO	-	0	1	1

SBC Activation

The ***SWD®*** manages two engine braking modes. Internal motor braking and external braking. Internal braking works by short-circuiting the motor phases. While the external brake works by lack of current. It is mounted on the motor shaft, at the rear of the product. The use or not of the internal brake is configurable[33] :

Braking applied according to the configuration used		
SBC commands	Internal brake	External brake
No SBC		
SBC command 1	✓	✓
SBC command 2		✓
SBC command 3	✓	

⚠ *If configured on 'SBC command 3', no external brake should be connected. If necessary, it will not be powered, and the **SWD®** will be braked by the external brake.*

When an external brake is fitted, the user must enter it in the dedicated field of the CANopen dictionary, 'Brake present' (2660_h).

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
2660_h	00_h	Brake present	BOOL	-[34]	RW	NO	-	0	1	1

2660_h: Brake present	No external brake mounted	External brake mounted
Value	**0**	**1**

The external brake is present if it has been correctly mounted on the motor and if it is connected to the dedicated connector.

ℹ If the configuration indicates that an external brake is fitted, but it is not detected, an error in the form of an EMCY message is sent, with the code 8016h BLC (Brake Lock Check).

ℹ If the configuration indicates that no external brake is fitted, but it is detected, an error in the form of an EMCY message is sent, with the code 8010h (EXT_BRAKE_PRESENCE).

By default, the "SBC command 3" braking mode is activated on an STO signal, i.e., internal motor braking.

By default, the "SBC command 3" braking mode is activated on an STO signal, i.e., internal motor braking.

An SBC command can also be activated via a safety command, i.e., a 'Safety Controlword' (6620_h).

⚠ *If an SBC command mode incompatible with the hardware has been set, an error in the form of an EMCY message is sent, with the code 6020h (CANOPEN_PARAMETER_ERROR).*

[33] Configuration possible from software version 'Firmware 2.0.x'. SBC command 3 by default otherwise.
[34] 'True' for products with external brake, 'False' for products without external brake.

The internal brake is maintained after disconnecting the power supply to the motors. Brake hold time is approximately three minutes.

Since software version 'Firmware 1.1.4', the internal brake remains disabled in the event of:

1. Driven by means external to the engine (towing)
2. Disconnection of a power source allowing energy storage, or that this one does not accept any more current in input.
3. 3. Detection of motor electronics overheating.

Towing is strongly discouraged and must remain within the nominal operating ranges of the engines.

SBU Activation

The ***SWD®*** provides a safe motor disengagement mode, to facilitate maintenance operations on mobile platforms. This SBU mode (Safe Brake Unlock) allows the brakes to be deactivated and the motor to be freewheeled.

This mode can be activated by a signal on a safety input of the ***SWD®*** or by reception of a safety message on the CANopen safety bus.

When this mode is activated, an EMCY emergency message is sent with the code 8012_h "SBU_SET".

SBU activation is limited to SWITCHED_ON_DISABLED and READY_TO_SWITCH_ON modes of the motor drive state machine. Otherwise, the activation of this mode fails, and an EMCY emergency message is sent with the code 8013_h "SBU_ACTIVATION_ERROR", causing the state machine to go into the 'Fault' state.

SDI Activation

Two SDIp and SDIn commands can be activated in the low state to prohibit the positive and negative direction of rotation of the motor respectively.

The two commands each have two indexes to enable two different nZero_SDI (speed in rpm) tolerance threshold levels in the SDI trigger mechanism. A total of four commands can be controlled:

- 66D0 01_h: SDI positive beyond threshold +*nZero_SDI*, value of sub-index 01
- 66D1 01_h: SDI negative beyond threshold -*nZero_SDI*, value of sub-index 01
- 66D0 02_h: SDI positive beyond threshold +*nZero_SDI*, value of sub-index 02
- 66D1 02_h: SDI negative beyond threshold -*nZero_SDI*, value of sub-index 02

The nZero_SDI trigger thresholds can be configured by registers 66D5 01_h and 66D5 02_h.

Exceeding the threshold by the rotational speed of the motor when an SDI command is active leads to the violation of an SDI and the triggering of the STO.

The reset mode then works in the same way as for the activation of STO only (see Activation of STO).

SDI Activation

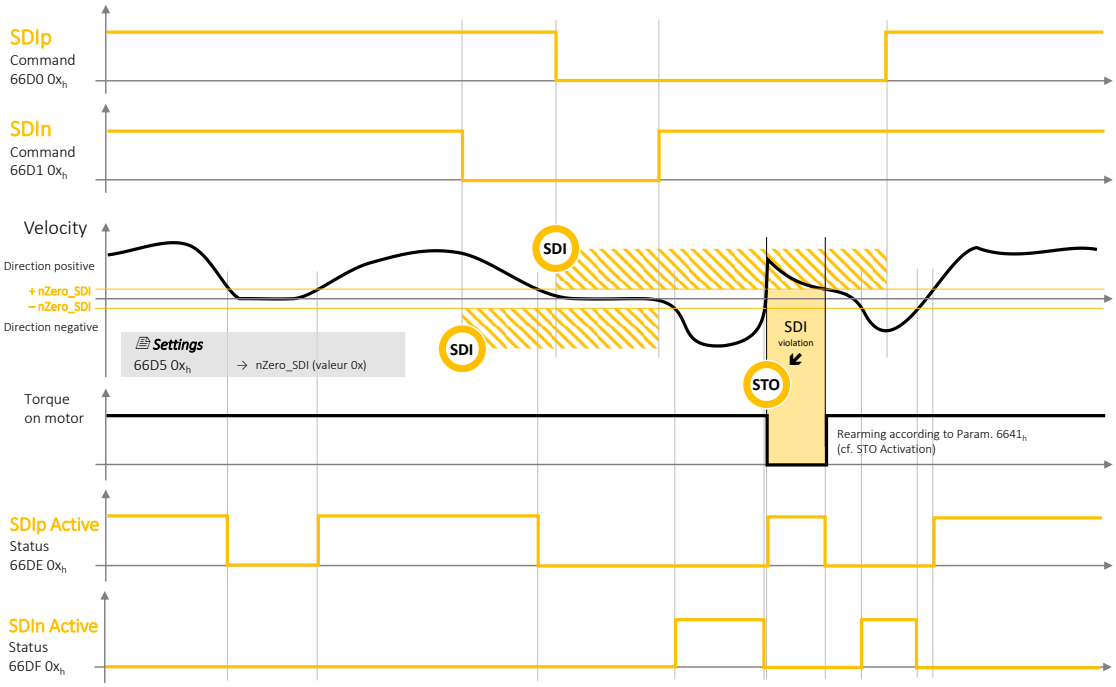


Figure 43 – SDI activation diagram

Two statuses SDIp Active and SDIn Active are available, even without activation of the SDI functions, to indicate if the rotation speed is beyond the nZero_SDI trigger threshold. The thresholds use the values of the two configured indexes, so there are a total of four rotation direction detection statuses that can be used to monitor motor activity:

- 66DE 01_h: rotation positive beyond threshold *+nZero_SDI*, value of sub-index 01
- 66DF 01_h: rotation negative beyond threshold *-nZero_SDI*, value of sub-index 01
- 66DE 02_h: rotation positive beyond threshold *+nZero_SDI*, value of sub-index 01
- 66DF 02_h: rotation negative beyond threshold *-nZero_SDI*, value of sub-index 01

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
66D0h	00h to 02h	SDIp Commands	BOOLEAN	1	RO	NO	-	0	1	1
66D1h	00h to 02h	SDIn Commands	BOOLEAN	1	RO	NO				
66D5h	00	SDI velocity zero window	U32	0	RW 🖹	NO	RPM	0	1	1
66DEh	00h to 02h	SDIp statuses	BOOLEAN	0	RO	NO	-	0	1	1
66DFh	00h to 02h	SDIn statuses	BOOLEAN	0	RO	NO	-	0	1	1

⚠ The SDI safety function is guaranteed from 50 rpm (rotations per minute) at the motor shaft. Below this value, the SIL level and the associated criteria PL, category are not guaranteed.

SLS Activation

Eight levels of SLS speed limitation can be activated by command 6690h, index 01 to 08.

The SLS 6690h is active low, and speed monitoring is activated after a trigger time defined by the first of these two events:

- The time t_SLS (in ms) elapsed since the activation of the SLS command
- The time t_L_SLS (in ms) elapsed after the speed passed through the authorized window

The authorized speed is limited by the n_SLS value (in rpm), in both directions of rotation.

For each of the eight actionable SLSs, the values can be configured using the registers:

- 6693 0xh: Speed n_SLS, value for index 0x [01..08]
- 6691 0xh: Time t_SLS, value for index 0x [01..08]
- 6694 0xh: Time t_L_SLS, value for index 0x [01..08]

Exceeding the motor rotation speed threshold when an SLS command is active leads to the violation of an SLS This can trigger an STO signal, depending on the configuration of the 'SLS error reactions' object 0x6698h. In this case, the reset mode works in the same way as for the activation of STO alone (see Activation of STO).

SLS Activation

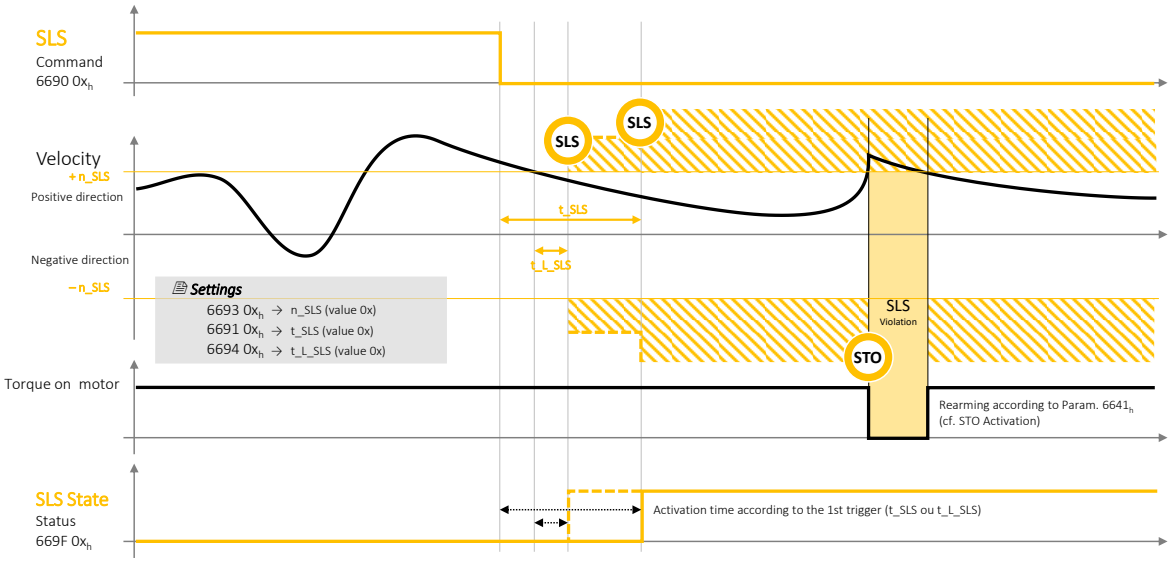


Figure 44 – SLS activation diagram

The status accessible in the SLS State 6697 0xh object, at sub-indexes [01...08], is active in the high state when the speed monitoring triggered by the SLS 6690 0xh command is activated.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
6690h	00h to 08h	SLS Commands	BOOLEAN	-	RO	NO	-	0	1	1	-
6691h	00h to 08h	Time to Monitoring start	U16	0	RW	NO	MS	0	U16	2	YES
6693h	00h to 08h	SLS speed limit	U32	0	RW	NO	rpm	100	U32	4	YES
6694h	00h to 08h	Time to SLS start “within limit”	U16	0	RW	NO	MS	0	U16	2	YES
6698h	00h to 08h	SLS reactions in the case of error	U32	6640 0000 h	RW	NO	-	0: no reaction 6640 0000 STO		4	YES
669Fh	00h to 02h	SLS statuses	BOOLEAN	-	RO	NO	-	0	1	1	-

⚠ The user can choose to disable the reaction if he wishes to use the 'status' of the SLS for another purpose. This configuration depends on the use case of the ***SWD®*** and is under the responsibility of the machine manufacturer.

⚠ The SLS safety function is guaranteed from 100 rpm (rotations per minute) at the motor shaft. Below this value, the SIL level and the associated criteria PL, category are not guaranteed.

Activation du SLSa

Eight levels of SLSa speed limit can be activated by command 3050h, sub-index 01 to 08.

SLSa 3050h is active in the low state, and speed monitoring is activated after a trigger time defined by the first of these two events:

- The time t_SLSa (in ms) elapsed since the activation of the SLSa command
- The time t_L_SLSa (in ms) elapsed after the speed passed through the authorized window

The authorized speed is limited by the n_SLSa value (in rpm), in a single direction of rotation.

For each of the eight actionable SLSa, the values can be configured with the objects:

- 3051h: 01h … 08h Positive Time t_SLSa+
- 3052h: 01h … 08h Positive speed n_SLSa+
- 3053h: 01h … 08h Positive limited time t_L_SLSa+
- 3054h: 01h … 08h Negative Time t_SLSa-
- 3055h: 01h … 08h Negative speed n_SLSa-
- 3056h: 01h … 08h Negative limited time t_L_SLSa-

Exceeding the motor rotation speed threshold when an SLSa command is active leads to the violation of an SLSa. This may trigger an STO signal, depending on the configuration of the 'SLSa error reactions' object 0x3057h. In this case, the reset mode works in the same way as for the activation of STO alone (see Activation of STO).

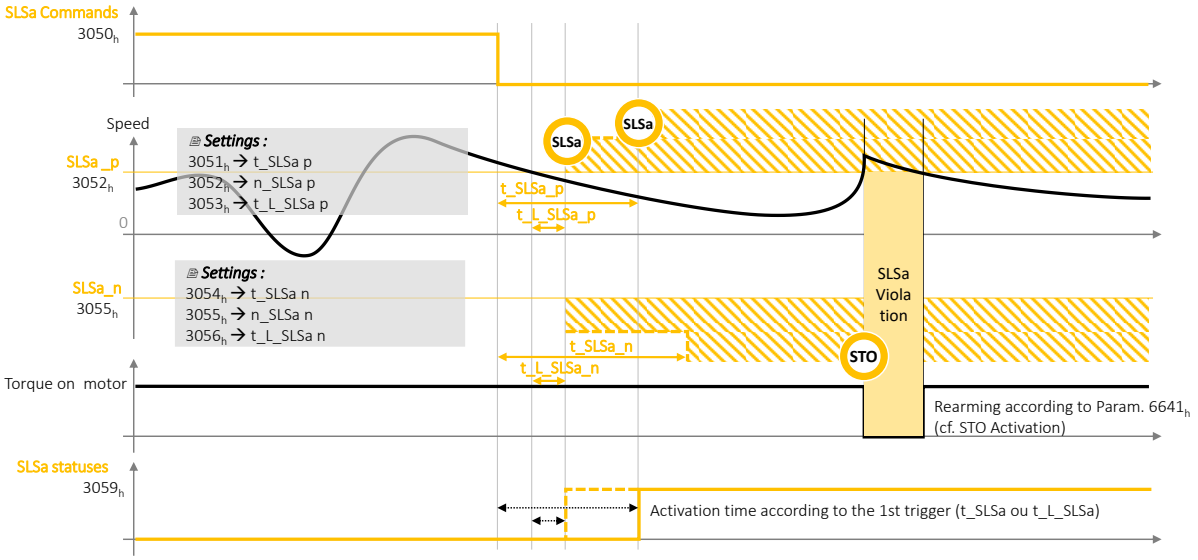


Figure 45 – SLSa activation diagram

The status accessible in the SLSa State 3059h object, at sub-index [01h … 08h], is active in the high state when the speed monitoring triggered by the SLSa 3050h command is activated.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
3050h	00h to 08h	SLSa commands	BOOLEAN	-	RO	NO	-	0	1	1	-
3051h	00h to 08h	SLSa time to positive velocity monitoring	U16	0	RW	NO	MS	0	U16	2	YES
3052h	00h to 08h	SLSa positive velocity limit u32	U32	0	RW	NO	RPM	0	U32	4	YES
3053h	00h to 08h	SLSa time for positive velocity in limits	U16	0	RW	NO	MS	0	U16	2	YES
3054h	00h to 08h	SLSa time to negative velocity monitoring	U16	0	RW	NO	MS	0	U16	2	YES
3055h	00h to 08h	SLSa negative velocity limit u32	U32	0	RW	NO	RPM		U32	4	YES
3056h	00h to 08h	SLSa time for negative velocity in limits	U16	0	RW	NO	MS		U16	2	YES
3057h	00h to 08h	SLSa error reactions	U32	6640 0000 h	RW	NO	-	0 = pas de reaction 6640 0000 = STO		4	YES
3058h	00h to 08h	SLSa safety application configuration signature	U16	-	RW	NO	-	0	1	2	YES
3059h	00h to 08h	SLSa statuses	BOOLEAN	-	RO	NO	-	0	1	1	YES

⚠ The user can choose to deactivate the reaction if he wishes to use the 'status' of the SLSa for another purpose. This configuration depends on the use case of the ***SWD®*** and is under the responsibility of the machine manufacturer.

⚠ The SLSa safety function is guaranteed from 100 rpm (rotations per minute) at the motor shaft. Below this value, the SIL level and the associated criteria PL, category are not guaranteed.

SMS Activation

Two maximum positive and negative SMS velocity commands can be activated in the low state to prevent the exceeding of a speed threshold.

The two SMS functions each have indexes for activating the speed threshold not to be exceeded (RPM), for a positive speed and for a negative speed.

- 66AA 01_h: SMS maximum positive velocity, value of sub-index 01
- 66AC 01_h: SMS maximum negative velocity, value of sub-index 01

Exceeding the motor rotation speed threshold when the SMS is activated leads to the violation of the SMS and the triggering of a configurable reaction.

- 66AD 01_h: SMS error reaction, sub-index value 01

By default, the configured reaction mode is the activation of the STO command '6640 00h' (see Activation of STO). If no reaction is set, the speed will be reduced to the given speed range.

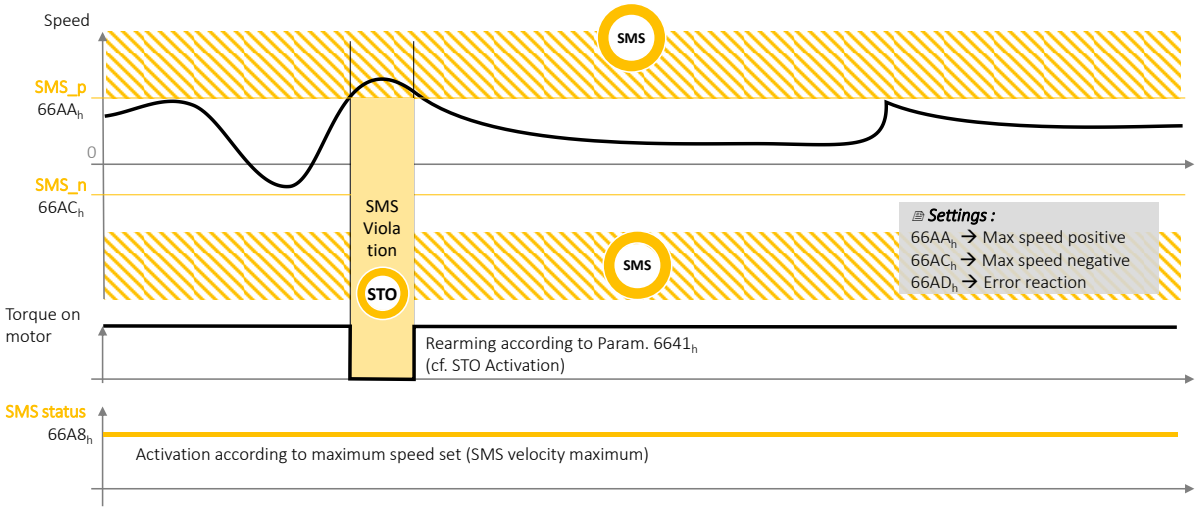


Figure 46 – SMS (Safe Maximum Speed) activation diagram

SMS status is available, indicating whether a maximum rotation speed has been set. If no speed threshold is configured (value of index 01 zero), then the SMS function is not activated.

- 66A8 01_h: SMS status, subindex value 01

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes
66A8$_h$	00_h to 01_h	SMS statuses	BOOLEAN	-	RO	NO	-	0	1	1
66AA$_h$	00_h to 01_h	SMS velocity maximum positive u32	U32	0	RW	NO	RPM	0	U32	4
66AC$_h$	00_h to 01_h	SMS velocity maximum negative u32	U32	0	RW	NO	RPM	0	U32	4
66AD$_h$	00_h to 01_h	SMS error reactions	U32	6640 00_h	RW	NO	-	0	U32	4
66AE$_h$	00_h to 01_h	SMS safety application configuration signature	U16	-	RW	NO	-	0	U16	2

11.8. Signatures

Security configurations include a CRC-type signature to ensure their integrity. A signature error prevents the transition to the NMT operational state.

In the event of an error on the signature of one of the safety functions, the object '6607 00h Safety application configuration valid' is worth 00h, otherwise it is worth A5h. Signatures must be updated in NMT Pre-Operational mode.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
6607_h	0	Safety application configuration valid	U8	0	RW	NO	-	0	255	1	Yes

Value	Definition
00_h	Safety configuration invalid
$A5_h$	Safety configuration valid

Calculation method

The calculation of the CRCs of the security data groups is performed with the CRC-16-CCITT generation polynomial: $G(x) = x^{16}+x^{12}+x^5+1$

Example of implementation:

```
UNSIGNED16   crc = 0u;
For each byte do:
crc = CrcCalc(crc, octet);
```

Example of CRC calculation function; CrcCalc():

```
UNSIGNED16 CrcCalc(
UNSIGNED16          crc,                /* start value for CRC     */
UNSIGNED8           value               /* pointer to data for CRC */
)
{
static const UNSIGNED16 crc_tabccitt[256] = {
 0x0000u, 0x1021u, 0x2042u, 0x3063u, 0x4084u, 0x50a5u, 0x60c6u, 0x70e7u,
 0x8108u, 0x9129u, 0xa14au, 0xb16bu, 0xc18cu, 0xd1adu, 0xe1ceu, 0xf1efu,
 0x1231u, 0x0210u, 0x3273u, 0x2252u, 0x52b5u, 0x4294u, 0x72f7u, 0x62d6u,
 0x9339u, 0x8318u, 0xb37bu, 0xa35au, 0xd3bdu, 0xc39cu, 0xf3ffu, 0xe3deu,
 0x2462u, 0x3443u, 0x0420u, 0x1401u, 0x64e6u, 0x74c7u, 0x44a4u, 0x5485u,
 0xa56au, 0xb54bu, 0x8528u, 0x9509u, 0xe5eeu, 0xf5cfu, 0xc5acu, 0xd58du,
 0x3653u, 0x2672u, 0x1611u, 0x0630u, 0x76d7u, 0x66f6u, 0x5695u, 0x46b4u,
 0xb75bu, 0xa77au, 0x9719u, 0x8738u, 0xf7dfu, 0xe7feu, 0xd79du, 0xc7bcu,
 0x48c4u, 0x58e5u, 0x6886u, 0x78a7u, 0x0840u, 0x1861u, 0x2802u, 0x3823u,
 0xc9ccu, 0xd9edu, 0xe98eu, 0xf9afu, 0x8948u, 0x9969u, 0xa90au, 0xb92bu,
 0x5af5u, 0x4ad4u, 0x7ab7u, 0x6a96u, 0x1a71u, 0x0a50u, 0x3a33u, 0x2a12u,
 0xdbfdu, 0xcbdcu, 0xfbbfu, 0xeb9eu, 0x9b79u, 0x8b58u, 0xbb3bu, 0xab1au,
 0x6ca6u, 0x7c87u, 0x4ce4u, 0x5cc5u, 0x2c22u, 0x3c03u, 0x0c60u, 0x1c41u,
 0xedaeu, 0xfd8fu, 0xcdecu, 0xddcdu, 0xad2au, 0xbd0bu, 0x8d68u, 0x9d49u,
 0x7e97u, 0x6eb6u, 0x5ed5u, 0x4ef4u, 0x3e13u, 0x2e32u, 0x1e51u, 0x0e70u,
 0xff9fu, 0xefbeu, 0xdfddu, 0xcffcu, 0xbf1bu, 0xaf3au, 0x9f59u, 0x8f78u,
 0x9188u, 0x81a9u, 0xb1cau, 0xa1ebu, 0xd10cu, 0xc12du, 0xf14eu, 0xe16fu,
 0x1080u, 0x00a1u, 0x30c2u, 0x20e3u, 0x5004u, 0x4025u, 0x7046u, 0x6067u,
```

```
 0x83b9u, 0x9398u, 0xa3fbu, 0xb3dau, 0xc33du, 0xd31cu, 0xe37fu, 0xf35eu,
 0x02b1u, 0x1290u, 0x22f3u, 0x32d2u, 0x4235u, 0x5214u, 0x6277u, 0x7256u,
 0xb5eau, 0xa5cbu, 0x95a8u, 0x8589u, 0xf56eu, 0xe54fu, 0xd52cu, 0xc50du,
 0x34e2u, 0x24c3u, 0x14a0u, 0x0481u, 0x7466u, 0x6447u, 0x5424u, 0x4405u,
 0xa7dbu, 0xb7fau, 0x8799u, 0x97b8u, 0xe75fu, 0xf77eu, 0xc71du, 0xd73cu,
 0x26d3u, 0x36f2u, 0x0691u, 0x16b0u, 0x6657u, 0x7676u, 0x4615u, 0x5634u,
 0xd94cu, 0xc96du, 0xf90eu, 0xe92fu, 0x99c8u, 0x89e9u, 0xb98au, 0xa9abu,
 0x5844u, 0x4865u, 0x7806u, 0x6827u, 0x18c0u, 0x08e1u, 0x3882u, 0x28a3u,
 0xcb7du, 0xdb5cu, 0xeb3fu, 0xfb1eu, 0x8bf9u, 0x9bd8u, 0xabbbu, 0xbb9au,
 0x4a75u, 0x5a54u, 0x6a37u, 0x7a16u, 0x0af1u, 0x1ad0u, 0x2ab3u, 0x3a92u,
 0xfd2eu, 0xed0fu, 0xdd6cu, 0xcd4du, 0xbdaau, 0xad8bu, 0x9de8u, 0x8dc9u,
 0x7c26u, 0x6c07u, 0x5c64u, 0x4c45u, 0x3ca2u, 0x2c83u, 0x1ce0u, 0x0cc1u,
 0xef1fu, 0xff3eu, 0xcf5du, 0xdf7cu, 0xaf9bu, 0xbfbau, 0x8fd9u, 0x9ff8u,
 0x6e17u, 0x7e36u, 0x4e55u, 0x5e74u, 0x2e93u, 0x3eb2u, 0x0ed1u, 0x1ef0u
};
UNSIGNED16 tmp, x;

x = (UNSIGNED16) value;
x &= 0xffu;

tmp = (crc >> 8) ^ x;
crc = (UNSIGNED16)((crc & 0xffu) << 8) ^ crc_tabccitt[tmp];

return(crc);
}
```

SRDO Signatures

Security message configurations include a CRC type signature to ensure their integrity. A signature error prevents the transition to the operational NMT state.

In the event of an error on the signature of one of the safety functions, the '6607 00_h Safety application configuration valid' object takes the value 00_h, otherwise it is $A5_h$. Signatures must be updated in NMT Pre-Operational mode.

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
$13FE_h$	0	Configuration valid	U8	0xA5	RW	NO	-	0	255	1	Yes

Value	Definition
00_h	Safety configuration invalid
$A5_h$	Safety configuration valid

As soon as the configuration of an SRDO has been modified, the object '13FE 00_h Configuration valid' automatically switches to the invalid state '00_h'.

There is a signature for each SRDO. The signatures are stored in the 'Safety configuration checksum' object 13FFh at Each sub-index corresponds to the signature of an SRDO:

Idx	Sub	Name	Data Type	Default Value	Access Type	PDO Mapping	Unit	Lower Limit	Upper Limit	Size in Bytes	Non-volatile storage
$13FF_h$	00_h	Number of entries	U8	16	RO	NO	-	0	255	1	Yes
	01_h	SRDO 1 signature	U16	$70AB_h$	RW	NO	-	0	65535	2	Yes
	02_h	SRDO 2 signature	U16	$8C5F_h$	RW	NO	-	0	65535	2	Yes

	09_h	SRDO 9 signature	U16	$6C9B_h$	RW	NO	-	0	65535	2	Yes
	$0A_h$	SRDO 10 signature	U16	2274_h	RW	NO	-	0	65535	2	Yes
	$0B_h$	SRDO 11 signature	U16	$78AE_h$	RW	NO	-	0	65535	2	Yes
	$0C_h$	SRDO 12 signature	U16	4052_h	RW	NO	-	0	65535	2	Yes
	$0D_h$	SRDO 13 signature	U16	9968_h	RW	NO	-	0	65535	2	Yes
	$0E_h$	SRDO 14 signature	U16	$1BF6_h$	RW	NO	-	0	65535	2	Yes
	$0F_h$	SRDO 15 signature	U16	$3A94_h$	RW	NO	-	0	65535	2	Yes
	10_h	SRDO 16 signature	U16	9941_h	RW	NO	-	0	65535	2	Yes

The configurations for each SRDO are stored in the following objects:

	Communication parameters index	Mapping parameters index
SRDO 1	1301_h	1381_h
SRDO 2	1302_h	1382_h
SRDO 9	1309_h	1389_h
SRDO 10	$130A_h$	$138A_h$
SRDO 11	$130B_h$	$138B_h$
SRDO 12	$130C_h$	$138C_h$
SRDO 13	$130D_h$	$138D_h$
SRDO 14	$130E_h$	$138E_h$
SRDO 15	$130F_h$	$138F_h$
SRDO 16	1310_h	1390_h

⚠ NB: SRDOs 3 to 8 do not exist.

The data to consider are, in order:

Index	Sub	Name	Size (Bytes)
<u>SRDO Communication Parameter:</u> 1301_h; 1302_h; 1309_h; $130A_h$; $130B_h$; $130C_h$; $130D_h$; $130E_h$; $130F_h$; 1310_h	01	Information direction	1
	02	Refresh time or SCT	2
	03	SRVT	1
	05	COB-ID 1	4
	06	COB-ID 2	4
Index of mapping parameters	00	Number of mapping parameters	1
For each object 'X' in the mapping:			
<u>SRDO Mapping Parameter:</u> 1381_h; 1382_h; 1389_h; $138A_h$; $138B_h$; $138C_h$; $138D_h$; $138E_h$; $138F_h$; 1390_h	00	Number of mapped objects	1
	0x	Mapped object entry	4

STO signature

The STO configuration signature is stored in object 6645 00_h

The data to be considered are, in order:

Index	Sub-Index	Name	Size (Bytes)
6641h	00	STO acknowledge behavior	1
66F0h	01 to 08	Safety Controlword 1	4 (for each sub-index)
66F1h	01 to 08	Safety Controlword 2	4 (for each sub-index)
66F2h	01 to 08	Safety Controlword 3	4 (for each sub-index)
66F3h	01 to 08	Safety Controlword 4	4 (for each sub-index)
66F4h	01 to 08	Safety Controlword 5	4 (for each sub-index)
66F5h	01 to 08	Safety Controlword 6	4 (for each sub-index)
66F6h	01 to 08	Safety Controlword 7	4 (for each sub-index)
66F7h	01 to 08	Safety Controlword 8	4 (for each sub-index)
66F8h	01 to 08	Safety Statusword 1	4 (for each sub-index)
66F9h	01 to 08	Safety Statusword 2	4 (for each sub-index)
66FAh	01 to 08	Safety Statusword 3	4 (for each sub-index)
66FBh	01 to 08	Safety Statusword 4	4 (for each sub-index)
66FCh	01 to 08	Safety Statusword 5	4 (for each sub-index)
66FDh	01 to 08	Safety Statusword 6	4 (for each sub-index)
66FEh	01 to 08	Safety Statusword 7	4 (for each sub-index)
66FFh	01 to 08	Safety Statusword 8	4 (for each sub-index)
26F1h	01 to 06	Safety Controlword Safe_in	4 (for each sub-index)
26F8h	01 to 04	Safety Statusword Safe_out	4 (for each sub-index)
2624h	01 to 08	Permanent Safety Controlword 1	4 (for each sub-index)
2625h	01 to 08	Permanent Safety Controlword 2	4 (for each sub-index)

SBC / SBU signatures

There is a signature for each of the 3 SBC functions.

Signatures are stored in the object **6662 0xh**, with sub-index x corresponding to the SBC configuration number.

The data to be considered are, in order:

Index	Sub-Index	Name	Size (Bytes)
6600h	00	Time unit	4
6661h	01 to 03 *(01 for instance 1, to 03 for instance 3)*	SBC brake time delay	2
2660h	00	Brake present	1
66F0h	01 to 08	Safety Controlword 1	4 (for each sub-index)
66F1h	01 to 08	Safety Controlword 2	4 (for each sub-index)
66F2h	01 to 08	Safety Controlword 3	4 (for each sub-index)
66F3h	01 to 08	Safety Controlword 4	4 (for each sub-index)
66F4h	01 to 08	Safety Controlword 5	4 (for each sub-index)
66F5h	01 to 08	Safety Controlword 6	4 (for each sub-index)
66F6h	01 to 08	Safety Controlword 7	4 (for each sub-index)
66F7h	01 to 08	Safety Controlword 8	4 (for each sub-index)
66F8h	01 to 08	Safety Statusword 1	4 (for each sub-index)

66F9h	01 to 08	Safety Statusword 2	4 (for each sub-index)
66FAh	01 to 08	Safety Statusword 3	4 (for each sub-index)
66FBh	01 to 08	Safety Statusword 4	4 (for each sub-index)
66FCh	01 to 08	Safety Statusword 5	4 (for each sub-index)
66FDh	01 to 08	Safety Statusword 6	4 (for each sub-index)
66FEh	01 to 08	Safety Statusword 7	4 (for each sub-index)
66FFh	01 to 08	Safety Statusword 8	4 (for each sub-index)
26F1h	01 to 08	Safety Controlword Safe_in	4 (for each sub-index)
26F8h	01 to 08	Safety Statusword Safe_out	4 (for each sub-index)
2624h	01 to 08	Permanent Safety Controlword 1	4 (for each sub-index)
2625h	01 to 08	Permanent Safety Controlword 2	4 (for each sub-index)

SLS Signatures

There is a signature for each of the 8 SLS functions.

Signatures are stored in object **6699 0xh**, with sub-index x corresponding to the SLS configuration number.

The data to be considered are, in order:

Index	Sub-Index	Name	Size (Bytes)
6600h	00	Time unit	4
6602h	00	Velocity unit	4
6603h	00	Acceleration unit	4
6691h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS time to velocity monitoring	2
6692h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS velocity limit u16 *(object does not exist, a null value 0000h must be used in the calculation)*	2
6693h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS velocity limit u32	4
6694h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS time for velocity in limits	2
6695h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS time delay deceleration monitoring *(object does not exist, a null value 0000h must be used in the calculation)*	2
6696h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS deceleration limit u16 *(object does not exist, a null value 0000h must be used in the calculation)*	2
6697h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS deceleration limit u32 *(object does not exist, a null value 0000 0000h must be used in the calculation)*	4
6698h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLS error reactions	4
66F0h	01 to 08	Safety Controlword 1	4 (for each sub-index)
66F1h	01 to 08	Safety Controlword 2	4 (for each sub-index)
66F2h	01 to 08	Safety Controlword 3	4 (for each sub-index)

66F3h	01 to 08	Safety Controlword 4	4 (for each sub-index)
66F4h	01 to 08	Safety Controlword 5	4 (for each sub-index)
66F5h	01 to 08	Safety Controlword 6	4 (for each sub-index)
66F6h	01 to 08	Safety Controlword 7	4 (for each sub-index)
66F7h	01 to 08	Safety Controlword 8	4 (for each sub-index)
66F8h	01 to 08	Safety Statusword 1	4 (for each sub-index)
66F9h	01 to 08	Safety Statusword 2	4 (for each sub-index)
66FAh	01 to 08	Safety Statusword 3	4 (for each sub-index)
66FBh	01 to 08	Safety Statusword 4	4 (for each sub-index)
66FCh	01 to 08	Safety Statusword 5	4 (for each sub-index)
66FDh	01 to 08	Safety Statusword 6	4 (for each sub-index)
66FEh	01 to 08	Safety Statusword 7	4 (for each sub-index)
66FFh	01 to 08	Safety Statusword 8	4 (for each sub-index)
26F1h	01 to 08	Safety Controlword Safe_in	4 (for each sub-index)
26F8h	01 to 08	Safety Statusword Safe_out	4 (for each sub-index)
2624h	01 to 08	Safety Controlword permanent 1	4 (for each sub-index)
2625h	01 to 08	Safety Controlword permanent 2	4 (for each sub-index)

SLSa Signatures

There is a signature for each of the 8 SLSa functions.

Signatures are stored in object **3058 0xh**, with sub-index x corresponding to the SLSa configuration number.

The data to be considered are, in order:

Index	Sub-Index	Name	Size (Bytes)
6600h	00	Time unit	4
6602h	00	Velocity unit	4
3051h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLSa time to velocity monitoring	2
3052h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLSa positive velocity limit u32	4
3053h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLSa time for positive velocity in limits	2
3054h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLSa time to negative velocity monitoring	2
3055h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLSa negative velocity limit u32	4
3056h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLSa time for negative velocity in limits	2
3057h	01 to 08 *(01 for instance 1, to 08 for instance 8)*	SLSa error reactions	4
66F0h	01 to 08	Safety Controlword 1	4 (for each sub-index)

66F1h	01 to 08	Safety Controlword 2	4 (for each sub-index)
66F2h	01 to 08	Safety Controlword 3	4 (for each sub-index)
66F3h	01 to 08	Safety Controlword 4	4 (for each sub-index)
66F4h	01 to 08	Safety Controlword 5	4 (for each sub-index)
66F5h	01 to 08	Safety Controlword 6	4 (for each sub-index)
66F6h	01 to 08	Safety Controlword 7	4 (for each sub-index)
66F7h	01 to 08	Safety Controlword 8	4 (for each sub-index)
66F8h	01 to 08	Safety Statusword 1	4 (for each sub-index)
66F9h	01 to 08	Safety Statusword 2	4 (for each sub-index)
66FAh	01 to 08	Safety Statusword 3	4 (for each sub-index)
66FBh	01 to 08	Safety Statusword 4	4 (for each sub-index)
66FCh	01 to 08	Safety Statusword 5	4 (for each sub-index)
66FDh	01 to 08	Safety Statusword 6	4 (for each sub-index)
66FEh	01 to 08	Safety Statusword 7	4 (for each sub-index)
66FFh	01 to 08	Safety Statusword 8	4 (for each sub-index)
26F1h	01 to 08	Safety Controlword Safe_in	4 (for each sub-index)
26F8h	01 to 08	Safety Statusword Safe_out	4 (for each sub-index)
2624h	01 to 08	Safety Controlword permanent 1	4 (for each sub-index)
2625h	01 to 08	Safety Controlword permanent 2	4 (for each sub-index)

SDI Signatures

There is a signature for each of the 2 SDI functions.

Signatures are stored in object **66D6 0xh**, the sub-index x corresponding to the SDI configuration number.

The data to be considered are, in order:

Index	Sub-Index	Name	Size (Bytes)
6601h	00	Position unit	4
6602h	00	Velocity unit	4
66D2h	01 to 02 *(01 for instance 1, to 02 for instance 2)*	SDI position zero window u16 *(object does not exist, a null value 0000h must be used in the calculation)*	2
66D3h	01 to 02 *(01 for instance 1, to 02 for instance 2)*	SDI position zero window u32	4
66D4h	01 to 02 *(01 for instance 1, to 02 for instance 2)*	SDI velocity zero window u16 *(object does not exist, a null value 0000h must be used in the calculation)*	2
66D5h	01 to 02 *(01 for instance 1, to 02 for instance 2)*	SDI velocity zero window u32	4
66F0h	01 to 08	Safety Controlword 1	4 (for each sub-index)
66F1h	01 to 08	Safety Controlword 2	4 (for each sub-index)
66F2h	01 to 08	Safety Controlword 3	4 (for each sub-index)
66F3h	01 to 08	Safety Controlword 4	4 (for each sub-index)
66F4h	01 to 08	Safety Controlword 5	4 (for each sub-index)
66F5h	01 to 08	Safety Controlword 6	4 (for each sub-index)
66F6h	01 to 08	Safety Controlword 7	4 (for each sub-index)
66F7h	01 to 08	Safety Controlword 8	4 (for each sub-index)

66F8h	01 to 08	Safety Statusword 1	4 (for each sub-index)
66F9h	01 to 08	Safety Statusword 2	4 (for each sub-index)
66FAh	01 to 08	Safety Statusword 3	4 (for each sub-index)
66FBh	01 to 08	Safety Statusword 4	4 (for each sub-index)
66FCh	01 to 08	Safety Statusword 5	4 (for each sub-index)
66FDh	01 to 08	Safety Statusword 6	4 (for each sub-index)
66FEh	01 to 08	Safety Statusword 7	4 (for each sub-index)
66FFh	01 to 08	Safety Statusword 8	4 (for each sub-index)
26F1h	01 to 08	Safety Controlword Safe_in	4 (for each sub-index)
26F8h	01 to 08	Safety Statusword Safe_out	4 (for each sub-index)
2624h	01 to 08	Safety Controlword permanent 1	4 (for each sub-index)
2625h	01 to 08	Safety Controlword permanent 2	4 (for each sub-index)

SMS Signature

There is a signature for the SMS function. The signature is stored in object **66AE 01h**.

The data to be considered are, in order:

Index	Sub-Index	Name	Size (Bytes)
6602h	00	Velocity unit	4
66A9h	01	SMS velocity maximum positive u16 *(object does not exist, a null value 0000h must be used in the calculation*	2
66AAh	01	SMS velocity maximum positive u32	4
66ABh	01	SMS velocity maximum negative u16 *(object does not exist, a null value 0000h must be used in the calculation*	2
66ACh	01	SMS velocity maximum negative u32	4
66ADh	01	SMS error reactions	4
66F0h	01 to 08	Safety Controlword 1	4 (for each sub-index)
66F1h	01 to 08	Safety Controlword 2	4 (for each sub-index)
66F2h	01 to 08	Safety Controlword 3	4 (for each sub-index)
66F3h	01 to 08	Safety Controlword 4	4 (for each sub-index)
66F4h	01 to 08	Safety Controlword 5	4 (for each sub-index)
66F5h	01 to 08	Safety Controlword 6	4 (for each sub-index)
66F6h	01 to 08	Safety Controlword 7	4 (for each sub-index)
66F7h	01 to 08	Safety Controlword 8	4 (for each sub-index)
66F8h	01 to 08	Safety Statusword 1	4 (for each sub-index)
66F9h	01 to 08	Safety Statusword 2	4 (for each sub-index)
66FAh	01 to 08	Safety Statusword 3	4 (for each sub-index)
66FBh	01 to 08	Safety Statusword 4	4 (for each sub-index)
66FCh	01 to 08	Safety Statusword 5	4 (for each sub-index)
66FDh	01 to 08	Safety Statusword 6	4 (for each sub-index)
66FEh	01 to 08	Safety Statusword 7	4 (for each sub-index)
66FFh	01 to 08	Safety Statusword 8	4 (for each sub-index)
26F1h	01 to 08	Safety Controlword Safe_in	4 (for each sub-index)
26F8h	01 to 08	Safety Statusword Safe_out	4 (for each sub-index)
2624h	01 to 08	Safety Controlword permanent 1	4 (for each sub-index)
2625h	01 to 08	Safety Controlword permanent 2	4 (for each sub-index)

11.9. Periodic checks

The correct operation of the safety functions must be periodically checked on the machine to guarantee operation with the correct safety level.

The need for and frequency of this check must be indicated in the documentation for the machine or equipment incorporating a safety drive from the ***SWD***® range.

The check verifies the ability of the safety functions to perform the associated safety function.

The frequency of testing depends on the safety level of the function. The test should be carried out once a year for SIL2 functions and once a month for the STO SIL3 function.

11.10. CANopen safety - SRDO

Engineers implementing this Part shall provide a safety manual containing at least the following information:

a) The safety manual must inform users of the constraints involved in calculating the system's characteristics (see 9.4);

b) The safety manual must inform users of their responsibilities regarding the correct configuration of the devices (6.4);

c) The safety manual should include advice on calculating the maximum expected response time of the network.

Constraints linked to the calculation of system characteristics

SRDO Number

The number of SRDO producers is limited to 64 in an SR system. The number of SRDO consumers is not limited.

i The number of SRDO producers is limited due to compatibility with EN 50325-4 which specifies only 128 reserved high priority CAN Identifiers, and the limited available bandwidth. A greater number of SRDO generators increases the likelihood of excessive traffic on the CAN giving rise to SR reactions due to simple overload.

Calculations used to determine the residual probability of error of the SRDO.

This paragraph describes the calculations used to determine the residual probability of error of the SRDO.

The worst-case residual error probability for ADC according to [17], [18] and [19] is given in (4). This worst-case residual error probability applies because the data link layer is used as part of the white channel method, which differs from the black channel method defined by the FSCPs specified in EN 61784-3-X.

$R(P_{CAN}) = 7 \times 10^{-9} \approx 1 \times 10^{-8}$

The worst-case residual error probability is squared in accordance with GS-ET-26 for the use of Model III (see A.4) as shown in (5). Other models may be used but it must be demonstrated that the following formula is still valid.

$R_{SL}(P) = R(P_{CAN})^2 = 4{,}9 \times 10^{-17}$

ℹ The definition of the white channel (EN 61784-3) requires an evaluation of the complete solution with all possible errors and failures of the transmission channel in accordance with the EN 61508 series.

ℹ The probability of residual error calculated in this paragraph and the formula used assume that the implementation of this SRCP uses redundant mechanisms or diversified methods to maintain security.

User responsibilities

SRDOs are configured under the responsibility of the user, who is responsible for configuring the product to meet the needs of the application.

Reaction time

The Safety Function Response Time (SFRT) represents the worst-case time from an SR event, considered as an input to the system or as a fault within the system, until the system is in the safe state. To determine the worst-case SFRT of any SR control loop, the user must sum all the worst-case safety reaction times of each subsystem of the SR control loop (see definitions in EN 61784-3). An example of the range of reaction times is shown below:

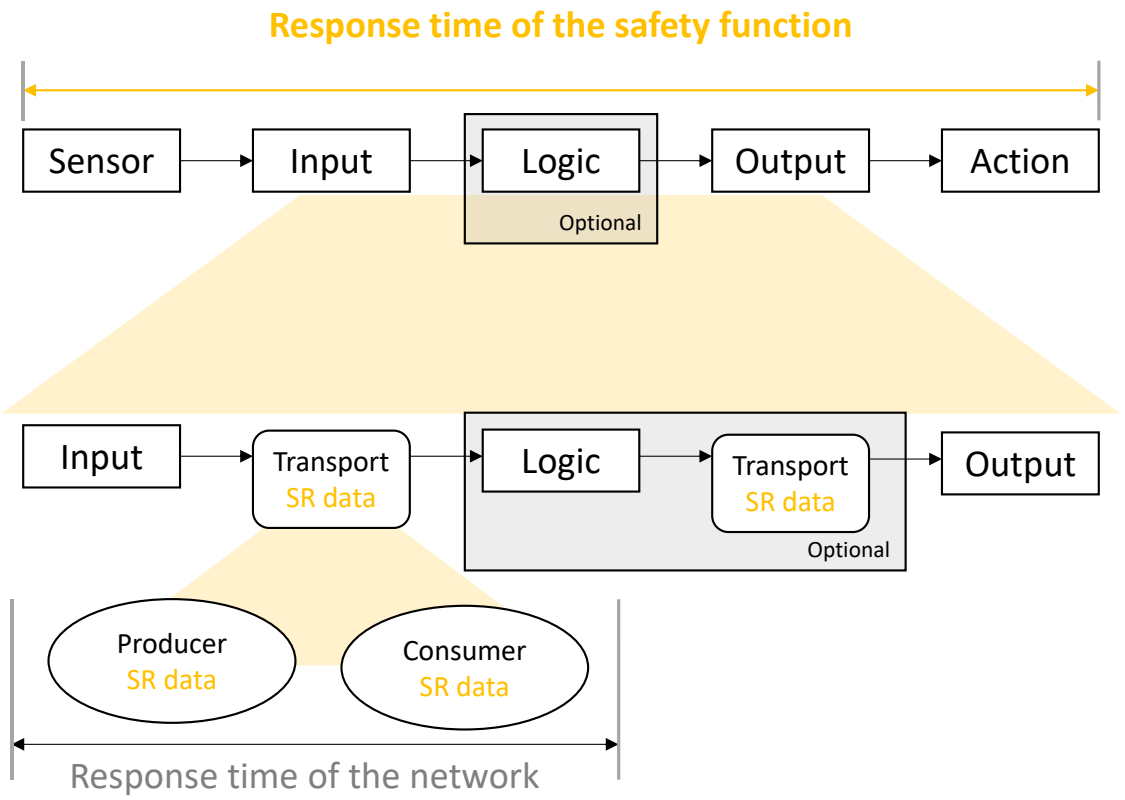


Figure 47 – example of the range of reaction times

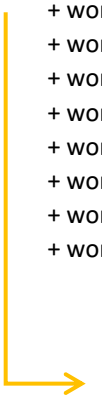
Example:

The SFRT shown in the previous figure comprises the following elements:

- Detector response time
- Input reaction time; network reaction time

- Manager reaction time, if a manager is present; network reaction time, if a manager is present; output reaction time; and
- Actuator reaction time

The SFRT is then the sum of the worst-case reaction times mentioned above:



+ worst-case detector reaction time
+ worst-case input reaction time
+ worst-case network's reaction time
+ worst-case manager's reaction time
+ worst-case network response time
+ worst-case output reaction time
+ worst-case actuator response time
+ worst-case time difference of a failed subsystem when the safety function is triggered

= safety function response time

For ***SWD***®, the maximum SFRT is 100 milliseconds.

Information & contact

For more information on *SWD®* technology:

	https://github.com/IDEC-ezWheel
	https://ez-wheel.com/en

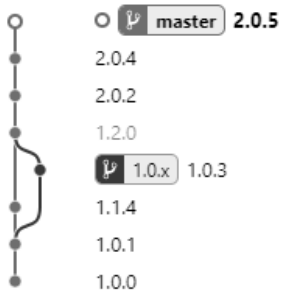
APEM SAS
IDEC Mobility Solutions division
Moulin de l'Abbaye, 135 route de Bordeaux
16400 La Couronne – France

Phone: +33 (0)5 31 61 55 80

Appendices

- **Versions overview**
- **Release notes**
- **SWD® - Quick start**

'Firmware' versions overview



Version 'Firmware'	Release date	Content
2.0.5	April 2025	- LSS Store unavailable in NMT OPER - Exact values for gear ratios - Fix startup at low voltage - Update Device type as CiA-402 compliant - Fix PDO mappings - Fix timeout on auto-test at low temperature
2.0.4	September 2024	- Odometry and speed measurement improved - Add default values for CANopen objects dictionary: o *0x6046 – vl_velocity_min_max_amount* o *0x6048 – vl_velocity_acceleration* o *0x6049 – vl_velocity_deceleration* o *0x604a – vl_velocity_quick_stop* o *0x604b – vl_set_point_factor* o *0x607e – polarity* o *0x6607 - Safety application configuration valid* o *0x6608 - Safety application configuration failed* - Removed default value for CANopen object dictionary : o *0x6061 –modes_of_operation_display* - Removed CANopen object dictionnay : o *0x6092 – feed_constant*
2.0.2	September 2023	- Add safety fuctions SMS, SLSa, SBC et SBU
1.2.0	March 2023	- Improve motor control - Support LSS protocol (NodeID, Baudrate) - Add TPDO revealing Safety commands states (STO, SDI, SLS, …) - Speed measurement from 1 RPM
1.1.4	October 2022	- Electronic protection in case of towing - Firmware update availbable with GUI - Implementation of SRDO 10 (Sending safety position and safety speed)
1.0.3	Decembre 2022	- Improve speed information - Improve motor calibration
1.0.1	March 2022	- 1st release of ***SWD®***
1.0.0	-	

Release notes

Release note 'Firmware 1.0.3'

Implementation of improved motor calibration and, consequently, improved symmetry of motor behaviour. The motor control considers geometric variations in the position of the magnets on the rotor.

A new calculation method provides motor speed data with less noise. This data is available in the 'od_velocity_actual_value' CANopen object. It gives a smoother motor speed, making it more stable over a given time.

Release note 'Firmware 1.1.4'

Incorporation of an overvoltage protection mechanism. These are likely to occur when the motor is driven by an external means. And when a power source used to store energy is no longer connected or no longer accepts input current. Or when an overheated motor is detected.

This protection deactivates internal braking by short-circuiting the motor phases when the system is no longer powered, or when a fault is detected on the power supply. As a result, the motor will not apply internal braking in the event of a loss of power, detection of an internal overtemperature or detection of an external drive.

⚠ Warning: this modification removes braking on battery disconnection. An impact analysis is required in the machine risk analysis before using this update.
⚠ Drive from the outside is not recommended, for example when towing the system.

A mechanism for checking software and hardware compatibility has been added. This means that when a software version is updated, the version of electronic board used is checked. This also ensures that the update runs smoothly.

⚠ A compatibility or integrity error makes it impossible to use the ***SWD®***. A new update will then be required.
⚠ For software updates, ***SWD®* Bootloader GUI** application must be used.

Safe speed and position information is now available in a CANopen Safety message. This information is sent in the SRDO 10 issued by the ***SWD®***.

Release note 'Firmware 1.2.0'

Version 1.2.0 includes improved engine control:

- The integral action of the PID controller is given an "anti-windup" action to limit saturation of the integral action. The time constant can be set via the "motctrl_speed_pid_tw" CANopen object.
- - The derivative action of the PID controller incorporates a low-pass filter with a time constant that can be set from the "od_motctrl_speed_pid_tn" CANopen object. The gain of the derivative action is now considered and can be set from the "od_motctrl_speed_pid_d" CANopen object.
- The minimum speed setpoint is lowered to 5 rpm-1 motor (before gear reduction)

An improvement in speed measurements, with a speed that can be measured at up to 1 rpm (before gear reduction).

Implementation of the Layer Settings Service (LSS) protocol, in accordance with the CiA-305 standard. This service enables the Baudrate and NodeID of a CANopen node to be configured. Each node can be individually addressed using the parameters in its "Identity Object". This object contains the node's serial number. This method allows

a node to be configured without knowing its NodeID, or if several nodes connected to the CANopen bus have the same NodeID.

The addition of objects that can be used to find out the status of safety function commands. The "safety_function_output_1" and "safety_function_output_2" objects indicate whether a request to activate a safety function is in progress. If this is the case, the bit corresponding to the safety function is set to 1b. These objects are read-only and can be transmitted in a TPDO.

Safety_function_output_1																
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Safety Function	STO	SBC_1	SBC_2	SBC_3	SLS_1	SLS_2	SLS_3	SLS_4	SLS_5	SLS_6	SLS_7	SLS_8	SDIP_1	SDIP_2	SDIN_1	SDIN_2

Bit = 0 → Safety Function command deactivated

Bit = 1 → Safety Function command activated

Safety_function_output_2																
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Safety Function	ERROR_ ACK	RESTART _ ACK	Not used													

Bit = 0 → Safety Function command deactivated

Bit = 1 → Safety Function command activated

The 'Hardware' and 'Firmware' versions and the commercial designation of the product are available in the CANopen dictionary objects, 'Hardware version', 'Software version' and 'Product ID'.

A fix for version 'Firmware 1.1.4' which caused the motor to enter a 'Fault' state when an external brake was connected.

Release note 'Firmware 2.0.2'

SMS, SLSa, SBC and SBU safety functions added.

SMS

The SMS safety function allows an overall speed limit to be imposed on the engine. The maximum forward and reverse speed can be set independently via two separate thresholds.

Depending on the configuration, the SMS can limit the motor speed to the thresholds or trigger an STO if a threshold is exceeded.

SLSa

The SLSa uses the same operating principle as the SLS but allows the thresholds for forward and reverse travel to be set independently.

SBC

It is now possible to configure motor braking via the SBC function.

4 modes are available:

- SBC NONE: freewheel
- SBC 1: application of the internal brake and the external brake
- SBC 2: application of the external brake only
- SBC 3: internal brake only

The availability of these modes depends on the object 2660_h 'brake_present':

- If an external brake is present: SBC NONE, SBC 1, or SBC 2
- If no external brake is present: SBC NONE or SBC 3.

Braking can be activated during an STO by configuring object 6643_h 'STO activate SBC'.

It is also possible to activate an SBC function on demand, via a controlword.

SBU

The SBU function enables the motor brakes to be disengaged for maintenance operations. This function can be activated via a controlword. Its activation is limited to the PDS states SWITCHED_ON_DISABLED and READY_TO_SWITCH_ON.

Release note 'Firmware 2.0.4'

Improved odometry and speed measurement, with better detection of events linked to digital Hall sensors.

Corrections to default values and read/write access rights for dictionary objects.

- Addition of default values for:
  - 0x6046 – vl_velocity_min_max_amount
  - 0x6048 – vl_velocity_acceleration
  - 0x6049 – vl_velocity_deceleration
  - 0x604a – vl_velocity_quick_stop
  - 0x604b – vl_set_point_factor
  - 0x607e – polarity
  - 0x6607 - Safety application configuration valid
  - 0x6608 - Safety application configuration failed
- Deletion of object default value:
  - 0x6061 – modes_of_operation_display
- Object deleted, as not implemented:
  - 0x6092 – feed_constant

Release note 'Firmware '2.0.5'

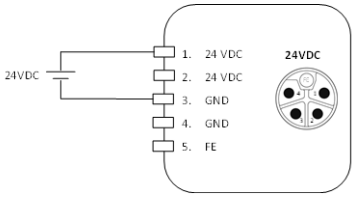
- Saving to EEPROM via LSS is now prohibited in 'NMT OPERATIONAL' state.
- Use of exact values for the reduction ratio in the '6091h gear_ratio' object, instead of rounded values in the numerator and denominator.
- Improvement when starting the product at low voltage (<24VDC). The product could be blocked in PDS 'FAULT' after the self-tests had failed.
- The device is now recognised as a CiA-402 compatible device. The value of the '1000h Device Type' object has been updateded.
- Corrections to the 'PDO mappable' values in the dictionary
    - '2064h accurate_position_value': changed from 'BOTH' to 'TPDO only'.
    - '3041h SBU status': changed from 'TPDO only' to 'NONE'.
    - '6042h vl_target_velocity': changed from 'BOTH' to 'RPDO only'.
    - '6043h vl_velocity_demand': changed from 'BOTH' to 'TPDO only'.
    - '6044h vl_velocity_actual_value': changed from 'BOTH' to 'TPDO only'.
- Improvement when starting the product at low temperature. The timeout for testing the motor driver has been increased from 200µs to 10ms.

SWD® - Quick start

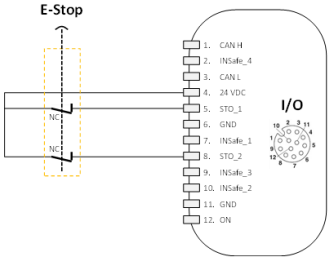
SWD® standalone without load

Hardware set-up:

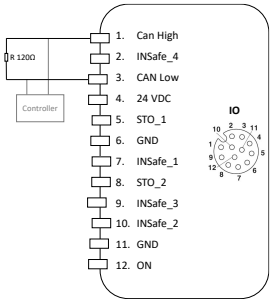
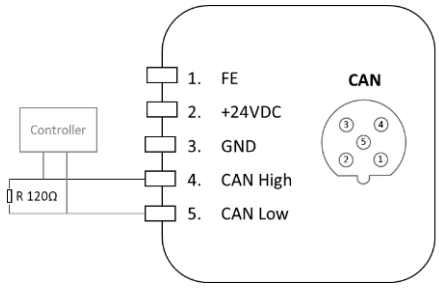
1. Connect the power to the ***SWD®*** *(~5A max. required):*



2. Connect ***SWD®***'s STO inputs to 24V:



3. Connect you controller to the CAN bus

Either on ***SWD®*** IO connector:	Or on dedicated ***SWD®*** CAN connector:
	

You can add a resistor to avoid a passive bus in you don't have an internal resistor on your controller.

A configurable internal resistor is integrated within ***SWD®*** between CAN High and CAN Low.

Default CANopen bus parameters

- *Node-ID = 0x10*
- *Baudrate = 1.000 kbits/s*

If you need to change those parameters, report to the section *CAN bus and CANopen protocol* of the manual.

Once the hardware has been set-up, there are two different ways to continue:

- **Method A: Using your own controller (e.g. PLC)**
- **Method B: Using an IPC with Linux can-utils and swd-services API delivered as a Debian package**

Method A: Using a PLC controller

4. Disable SRDO1 (RX), or send the safety message, i.e. SRDO1:

Disable reception of SRDO1 on ***SWD®***

Set SRDO1 field direction 0x1301:01 to:
Value = 0x00

CAN-ID	Length	Data
610_h	8_h	2F 01 13 01 00 00 00 00_h

Update SRDO1 safety signature 0x13FF:01 to:
Value = 0xBABD

CAN-ID	Length	Data
610_h	8_h	2B FF 13 01 BD BA 00 00_h

Set SRDO configuration to valid by setting object 0x13FE:00 to:
Value = 0xA5

CAN-ID	Length	Data
610_h	8_h	2F FE 13 00 A5 00 00 00_h

Or send the SRDO1 to the ***SWD®***

Send the safety message on bus can with the following parameters:

CAN-ID	Length	Data
$11F_h$	1_h	FF_h
120_h	1_h	00_h

5. Save configuration *(optional)*

Set the object 'Store Parameters', sub-object 'Save all Parameters' $1010:01_h$ to value = 0x65 76 61 73. So that, the configuration will be saved after On-Off[35].

CAN-ID	Length	Data
610_h	8_h	23 10 10 01 73 61 76 65_h

[35] Cf. *Store and restore configurations*

6. Start NMT state machine:

Send NMT start to SWD node (0x10):

CAN-ID	Length	Data
000_h	2_h	01 10_h

Or Broadcast NMT start to all nodes:

CAN-ID	Length	Data
000_h	2_h	01 00_h

*NB: **SWD®** CAN led should be solid green. **SWD®** heartbeat (CAN-ID $0x710_h$) should display value 05_h*

7. Start PDS state machine:

The 'Controlword' object 0x6040:00, is used for controlling the drive.
The 'Statusword' object 0x6041:00, is used to observe its state.

State	Statusword value	Command	Controlword value:
Fault	00 28_h 00 08_h	'Reset Fault' *(If necessary)*:	00 80_h (↑Bit7 rising edge)
Switch on disabled	00 40_h 00 60_h	Go to 'Ready to switch on'	00 06_h
Ready to switch on	00 21_h	Go to 'Switched on'	00 07_h
Switched on	00 23_h	Go to 'Operation enabled'	00 $0F_h$
Operation enabled	**00 27_h**		
Quick stop	00 07_h	Go to 'Operation enabled'	00 $0F_h$

*NB: **SWD®** Status led should be solid green. If it is red, a STO is activated. Release the emergency stop or check your STO wiring.*

SDO request to read 'Statusword' value and get PDS state:

	CAN-ID	Length	Data
Request	610_h	8_h	40 41 60 00 00 00 00 00_h
Response	590_h	8_h	4B 41 60 00 ***XX*** 00 00 00_h

Where '***XX***' represents the first 8-bits value of the Statusword.

SDO request to write 'Controlword' value and control the PDS:

Command	CAN-ID	Length	Data
'Reset Fault'	610_h	8_h	2B 40 60 00 80 00 00 00_h
Go to 'Ready to switch on'	610_h	8_h	2B 40 60 00 06 00 00 00_h
Go to 'Switched on'	610_h	8_h	2B 40 60 00 07 00 00 00_h
Go to 'Operation enabled'	610_h	8_h	2B 40 60 00 0F 00 00 00_h
Halt	610_h	8_h	2B 40 60 00 40 00 00 00_h
Quick stop	610_h	8_h	2B 40 60 00 02 00 00 00_h

8. Send velocity commands:

Write the desired target velocity in 0x6042:00 'vl_target_velocity':

Speed (RPM on motor shaft)	CAN-ID	Length	Data
$012A_h$ = $+298_d$	610_h	8_h	2B 42 60 00 2A 01 00 00_h
$FED6_h$ = -298_d	610_h	8_h	2B 42 60 00 D6 FE FF FF_h
0578_h = $+1400_d$	610_h	8_h	2B 42 60 00 78 05 00 00_h
$FA88_h$ = -1400_d	610_h	8_h	2B 42 60 00 88 FA FF FF_h

Method B: Use an IPC with swd-services Debian package.

- Install the package, following the installation guide on our GitHub page :

https://github.com/IDEC-ezWheel/swd-services/blob/main/docs/md/installation.md

- Start using it, following the usage guide on our GitHub page:

https://github.com/IDEC-ezWheel/swd-services/blob/main/docs/md/usage.md

Once your CAN is configured, you can create a bash script to execute the same steps as explained in **Method A**.

```
#!/usr/bin/env bash

# 4.Disable SRDO1 (RX)
printf "Set SRDO1 field direction to 0 \n"
cansend can0 610#2F01130100000000

printf "Update SRDO1 safety signature to: 0xBABD \n"
cansend can0 610#2BFF1301BDBA0000

printf "Set SRDO configuration to valid (Value = 0xA5) \n"
cansend can0 610#2FFE1300A5000000

# 5. Save configuration
printf "Store Parameters \n"
cansend can0 610#2310100173617665
sleep 1

# 6. Start NMT state machine:
printf "Broadcast NMT start to all nodes \n"
cansend can0 000#0100 & sleep 1

# 7. Start PDS state machine:
printf "Reset Fault \n"
cansend can0 610#2B40600080000000 & sleep 1

printf "Go to ‘Ready to switch on’ \n"
cansend can0 610#2B40600006000000 & sleep 1

printf "Go to ‘Switched on’ \n"
cansend can0 610#2B40600007000000 & sleep 1

printf "Go to ‘Operation enabled’ \n"
cansend can0 610#2B4060000F000000 & sleep 1

# 8. Send velocity commands:
printf "Speed +298d RPM \n"
cansend can0 610#2B4260002A010000 & sleep 5

printf "Speed -298d RPM \n"
cansend can0 610#2B426000D6FEFFFF & sleep 5

printf "Speed -1400d RPM \n"
cansend can0 610#2B42600088FAFFFF & sleep 5

printf "Quick Stop \n"
cansend can0 610#2B40600002000000 & sleep 3

printf "Speed +1400d RPM \n"
cansend can0 610#2B42600078050000

printf "Go to ‘Operation enabled’ \n"
cansend can0 610#2B4060000F000000 & sleep 5
```

```
printf "Speed 0 RPM \n"
cansend can0 610#2B42600000000000 & sleep 5

printf "Halt \n"
cansend can0 610#2B40600040000000 & sleep 5

# # 4. Example for sending SRDO1
# while(true)
# do
#     cansend can0 11F#FF
#     cansend can0 120#00
# done
```

NB: You can create a symbolic link of the remote script in your home directory "~" *(Optional):*

```
ln -s /opt/ezw/usr/sbin/remote.py ~
```

- Start the remote in a new terminal:

```
~/remote.py smc_drive
```

Keyboard commands are listed at the end.

NB: You can also use remote.py for steps 6, 7, and 8.

NB: For further information, see our GitHub repositories*:* *https://github.com/IDEC-ezWheel/*.