LAFERT
CANOPEN MANUAL

Release 1.4
Date 05/02/2021

This page has been intentionally left blank

INDEX

Tables and Figures:

REFERENCE DOCUMENTS:

- *Lafert User Guide*
- *CiA 301 (310_1v01010005_cor.pdf)*
- *CiA 402 (CiA® 402 Draft Standard Proposal.pdf)*

TERMS AND ABBREVIATIONS

CAN	Controller Area Network.
CiA	CAN in Automation.
COB	Communication OBject, transport unit in a CAN network.
COB-ID	Communication OBject Identifier.
DS301	Profile 301 standardizes the CANopen communication profile.
DSP402	Profile 402 standardizes the CANopen device profile for drives.
EDS	Electronic Data Sheet.
EMCY	Emergency Object.
EMC	Electromagnetic compatibility.
HMI	Human Machine Interface.
I/O	Input/output.
LSB	Least significant bit/byte.
LSD	Lafert Servo Drives.
MASTER	It is a device that controls and communicates with drive.
MSB	Most significant bit/byte.
MSM	Macro State Machine of Lafert Servo Drives.
NMT	Network Management.
IdNode	Node address assigned to a device on the network.
OD	Object dictionary.
PDO	Process Data Object.
PDS	Power Drive System.
REG	Register.
RO	Denotes read-only access.
RPDO	Receive (incoming) PDO
RW	Denotes read/write access.
RX	Messages sent by Main Control Board and received by Drive.
SDO	Service Data Object.
STO	Safe Torque Off
TX	Messages sent by Drive and received by Main Control Box
TPDO	Transmit (outgoing) PDO

FIRMWARE AND MANUAL RELEASED

This table shows the correlation between firmware and CANopen Manual.

Lafert Servo Drive	Firmware Released	CANopen Manual
SMARTRIS	2.0.1	1.4

1. | PURPOSE OF MANUAL

PURPOSE OF THE MANUAL

This operating guide provides information for safe installation and commissioning of the Drive.

Read and follow the instructions to use the Lafert Drive safely and professionally and pay attention to the safety instructions and general warnings.
Always keep this operating guide available with the Drive.

This operating guide provides information for safe installation and commissioning of the Lafert Drive:read carefully the entire guide before installing and using the equipment.

Caution

The operating guide is intended for use by qualified personnel.
THIS MANUAL IS ONLY FOR THE CANOPEN ON LAFERT DRIVE

This guide is delivered subject to the following conditions and restrictions:

- This guide contains proprietary information belonging to Lafert Spa.
- Such information is supplied solely for the purpose of assisting users of Lafert servo drives in implementing CANopen networking.
- The text and graphics included in this manual are for the purpose of illustration and reference only. The specifications on which they are based are subject to change without notice.
- Information in this document is subject to change without notice. Corporate and individual names and data used in examples herein are fictitious unless otherwise noted.

This manual is regularly reviewed and updated. All suggestions for improvement are welcome.

WARNING SAFETY INFORMATION

In order to achieve the optimum, safe operation of the Drive, it is imperative that you implement the safety procedures included in this installation guide. This information is provided to protect you and to keep your work area safe when operating the Drive and accompanying equipment.

Safety Instructions: for the electrical installation, the ESD instructions must be observed.

Caution

- The Systems that are electrically connected must be properly secured so they cannot be

switched back on and warnings signs must be put up.

- Before start-up, it must be checked that the wiring is correct and is free of mechanical damages. Only drive with wiring in perfect condition may be enabled to operation.
- Incorrect voltage, reverse polarity and defective wiring can damage the drive.
- Do not connect or disconnect electric cables while the equipment is powered or running.
- The operator is responsible for keeping the safety installations in perfect working order, conforming to prevailing laws and standards.

Please read these chapters carefully before you begin the installation process.

The Lafert Drive contains electrostatic-sensitive components that can be damaged if handled incorrectly. To prevent any electrostatic damage, avoid contact with highly insulating materials, such as plastic film and synthetic fabrics. Place the product on a conductive surface and ground yourself in order to discharge any possible static electricity build-up.
To avoid any potential hazards that may cause severe personal injury or damage to the product during operation, keep all covers and cabinet doors shut.

The following safety symbols are used in this manual:

Warning

This information is needed to avoid a safety hazard, which might cause bodily injury or death as a result of incorrect operation:

- To avoid electric arcing and hazards to personnel and electrical contacts, never connect/disconnect the servo drive while the power source is on.
- Power cables can carry a high voltage, even when the motor is not in motion. Disconnect the Lafert Drive from all voltage sources before servicing.
- After shutting off the power and removing the power source from your equipment, wait at least 1 minute before touching or disconnecting parts of the equipment that are normally loaded with electrical charges (such as capacitors or contacts). Measuring the electrical contact points with a meter, before touching the equipment, is recommended.

Caution

This information is necessary to prevent bodily injury, damage to the product or to other equipment:

- The maximum DC power supply connected to the instrument must comply with the parameters outlined in this guide.
- When connecting the Lafert Drive to an approved control supply, connect it through a line that is separated from hazardous live voltages using reinforced or double insulation in accordance with approved safety standards.
- Before switching on the Drive, verify that all safety precautions have been observed and that the installation procedures in this manual have been followed.
- Make sure that the Safe Torque Off is operational.
- If a fire breaks out, do not direct the water extinguishers near the equipment to put out the flames.

Important

Identifies information that is critical for successful application and understanding of the product.

Safety measures must be taken both for people and machines, in compliance with Standards and local conditions.

APPROVALS

CE Conformity

The Lafert Drive was tested in authorized testing laboratories in accordance with the requirements of this documentation.

The Lafert Drive is in conformity with the following **EC Directives:**

- Low Voltage Directive (*2014/35/EC*)
- Electromagnetic Compatibility (EMC) (*2014/30/EU*)
- RoHS Directive (*2011/65/EU*)
- WEEE Directive (*2012/19/UE*)

Safety

EN 61800-5-1 Adjustable speed electrical power drive systems - Part 5-1: Safety requirements - Electrical, thermal and energy.

EMC Requirements

In terms of emission and immunity, the Lafert Drive fulfills the requirement for the category"second environment" (industrial environment).
EN 61800-3 - Adjustable speed electrical power drive systems - Part 3: EMC requirements and specific test methods.

Safety Conformity (STO) - Where Available

The Lafert Drive provides a two-channel, functionally safe STO function (Safe Torque Off).
The function disables the PWM and the drive can be switched safely to torque OFF.

The circuit design has been tested and subsequently assessed by TÜV Süd. According to thatassessment, the circuit design used for the "Safe Torque Off" safety function in the Lafert Drive is suitable for meeting the requirements for .... in accordance with

- ***EN61508*** - Functional safety of electrical/electronic/programmable safety-related systems
- ***EN61800-5-2*** and category ... – Adjustable speed electrical power drive systems – Part 5-2: Safety requirements – Functional
- ***EN ISO 13849-1:2015*** - Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design.

The subsystems (Lafert Drive) are fully described in terms of safety by the following characteristics:

EN 13849-1	EN 61508	PFHD [1/h]
PLe	SIL3	

STARTUP

Startup is prohibited within the scope of the EC directives until it has been determined that the machine/system in which this Lafert Drive is installed corresponds to the regulations within these directives.

Correct Use

The Lafert Drive is intended for operation of permanent magnet synchronous servomotors with compatible feedback systems in stationary machines and systems.
Installation of the Lafert Drive is only approved in industrial environments. For use in residential areas, additional EMC measures are necessary. The user must prepare a hazard analysis of the final product.

Caution

Other uses must first be approved by the manufacturer.

Improper Use

The Lafert Drive is not suitable for operation of motors other than synchronous servo motors or motors with non-compatible feedback systems.

In addition, the following applications are expected from intended use.

The installation of drives in areas at risk, where inflammable substances or combustible vapors or powders are present, could trigger fire outbreaks or explosions. As such, install the drives far away from said areas at risk, even if they are used with motors fit for use under these conditions.

2. | CANOPEN OPERATION

CANopen is a communication protocol and device profile specification for embedded systems used in automation.

The CANopen standard uses an addressing scheme, several small communication protocols and an application layer defined by a device profile.

CANOPEN NETWORK TOPOLOGY OVERVIEW

CANOpen SIGNAL	
SIGNAL	**DESCRIPTION**
GND_CAN	GND reference for CAN
CAN_T	120 Ω Termination resistance CAN (connect to CAN H)
CAN_L	CAN_L Connection
CAN_H	CAN_H Connection

Table 1 - CANOpen Signal

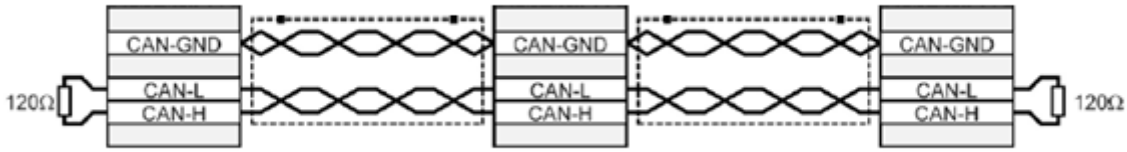


Figure 1- CANopen Network

All nodes of a network are principally connected in series, so that the CAN cable is looped through all controllers. The two ends of the CAN cables have to be terminated by a resistor of 120Ω +/-5%.

For further information refer to the Controller Area Network protocol specification, Ver. 2.0, Robert Bosch GmbH, 1991.

CANOpen Baud Rate and ID Node

Compliance with the directives CiA DS301 v4.02 and DSP402 v2.0

- Baudrate set by a CANOpen object (default: 1000Kb)
- Id-Node set by software (default value: ID 1)

Caution

When there are more than one drives in the same bus CAN it is mandatory to have different Id-nodes.

information

Referring to "APPENDIX" chapter to know the "How to change Id-Node" and "How to change BaudRate"

Client - Server

A CAN master (or client) is a controller that makes requests to nodes to respond to its commands. A CAN slave (or server) responds to the commands issued by the CAN master. The CAN protocol permits both single-master and multiple-master networks.

Information

The Lafert Servo Drive is the SLAVE and the machine controller or PLC is the MASTER.

Every servo drive has a unique ID in the range [1…127]. The network master does not require an ID.
As a slave, the servo drive never sends an unrequested message, other than emergencies. The drive responds only to messages addressed to its ID or to broadcast messages, which have an ID of 0. All messages sent by a servo drive are marked with its own ID.

Caution

If two servo drives have been assigned the same ID, the CAN network may crash.

Electronic Data Sheet (EDS)

The EDS file is the standardized format for the description of devices.

It contains information about:

- File properties (name, version, release date, …)
- General device information (manufacturer name and code)
- Device name and type, version
- Supported baud rates and boot-up options
- Description of supported objects and attributes

Object Dictionary (O.D.)

The most important part of a device profile is the Object Dictionary description. The Object Dictionary is essentially a grouping of objects accessible via the network in an ordered pre-defined fashion. Each object within the dictionary is addressed using a 16-bit index.

The general structure of the object dictionary is as follows:

Index, Sub-Index (HEX)	OBJECT (Symbolic Name)	Name	Type	Attribute	M/O

- **Index, Sub-Index:** The Index column denotes the objects position within the Object Dictionary. This acts as a kind of address to reference the desired data field. The sub-index is not specified

here. The sub-index is used to reference data fields within a complex object such as an array or record.

- **Object:** The Object column contains the Object Name and is used to denote what kind of object is at that particular index within the Object Dictionary.
- **Name:** The name column provides a simple textual description of the function of that particular object.
- **Type:** The type column gives information as to the type of the object. Eg: Boolean, Floating number, Unsigned Integer, Signed Integer etc.
- **Attribute:** The Attribute column defines the access rights for a particular object. Eg: rw (read and write access), wo (write only), ro (read only), Const (read only and value is constant).
- **M/O:** The M/O column defines whether the object is **M**andatory or **O**ptional

The standard object dictionary is as shown below:

Index (HEX)	Object
0000	Not used
0001-001F	Static data types
0020-003F	Complex data types
0040-005F	Manufacturer specific Complex data types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved for further us
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardized Device Profile Area
A000-FFFF	Reserved for further use

System Description

information

Compliance with the directives CiA DS301 v4.02 and DSP402 v2.0

- **Identity objects:** Identity including vendor ID, product code, revision number and serial number. BaudRate set by a CANOpen (default: 1000Kb), Id-Node set by CANOpen object (default: Id node is 1)
- **Service Data Object (SDO):** SDO messages are used for reading and writing access to all entries of the object dictionary. SDOs are used for device configuration in the first place.
- **Process Data Object:** The real-time data transfer of target position, target velocity and definitions input and output is performed by PDO messages. Data is transmitted within four TPDO's (transmit PDO) and each with a maximum 8 byte wide data block. There are a static map with 4 TPDO and 4 RPDO.
- **Network Management (NMT):** The NMT state machine defines the communication behaviour of the CANopen device.

- **Emergency object:** Emergency messages are triggered by the occurrence of a device internal fatal error situation and are transmitted from the application device concerned to the other devices with highest priority. This makes them suitable for interrupting type error alerts.
- **Sync Message:** The SYNC protocol enables synchronous network behaviour.
- **Node-Guard Protocol:** Cyclic querying of the node state by the NMT Master Controller. The NMT Master Controller sends messages to the CANopen slaves that then respond within a defined time.
- **Heartbeat Function Protocol:** Automatic transmission of a heartbeat message by the network nodes. A heartbeat message is sent to the bus in millisecond intervals. Heartbeats are useful for detecting the presence or absence of a node on the network.
- **Event timer:** (not implemented)
- **Store and Restore Parameters:** Parameters save on non volatile memory (communication, manufacturer and device profile).
- **Input/Output:** the digital input and output are defined by object Enable input (a low level put the Drive in StandBy mode, Switch on disabled)
- **State machine:** The device control is performed by a state machine according to DSP402**.**
- **Modes of operation:** Different operation modes are available with the CiA 402 profile. Also, the drive supports the manufacture operation mode where the drive is to control with hardware interface

Communication CANopen Object (COB)

The communication objects are standardized with the DS301 CANopen communication profile.
The objects can be classified into 4 groups according to their tasks.

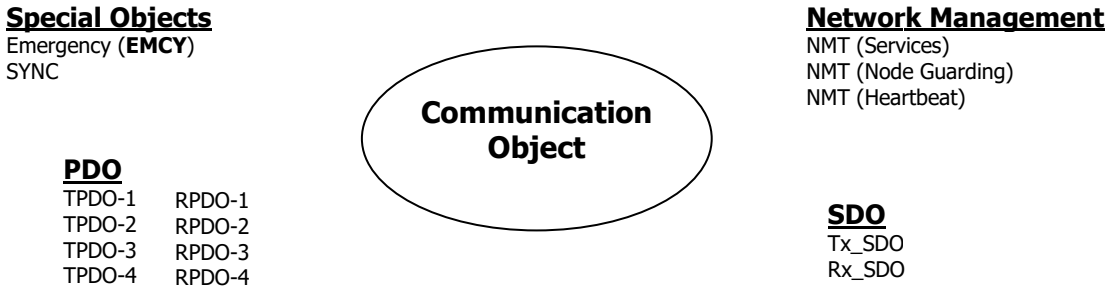


Figure 2 - Communication CANopen Object (COB)

- PDOs (process data objects) for real-time transmission of process data
- SDOs (service data object) for read and write access to the object Dictionary
- Objects for controlling CAN messages:
  - SYNC object (synchronization object) for synchronization of network devices (Not implemented)
  - EMCY object (emergency object), for signalling errors of a device or its peripherals.
- Network management services:
  - NMT services for initialization and network control (NMT: network management)
  - NMT Node Guarding for monitoring the network devices
  - NMT Heartbeat for monitoring the network devices

For communication between Master Controller and Lafert Servo Drive the following communication objects (COB) are available.

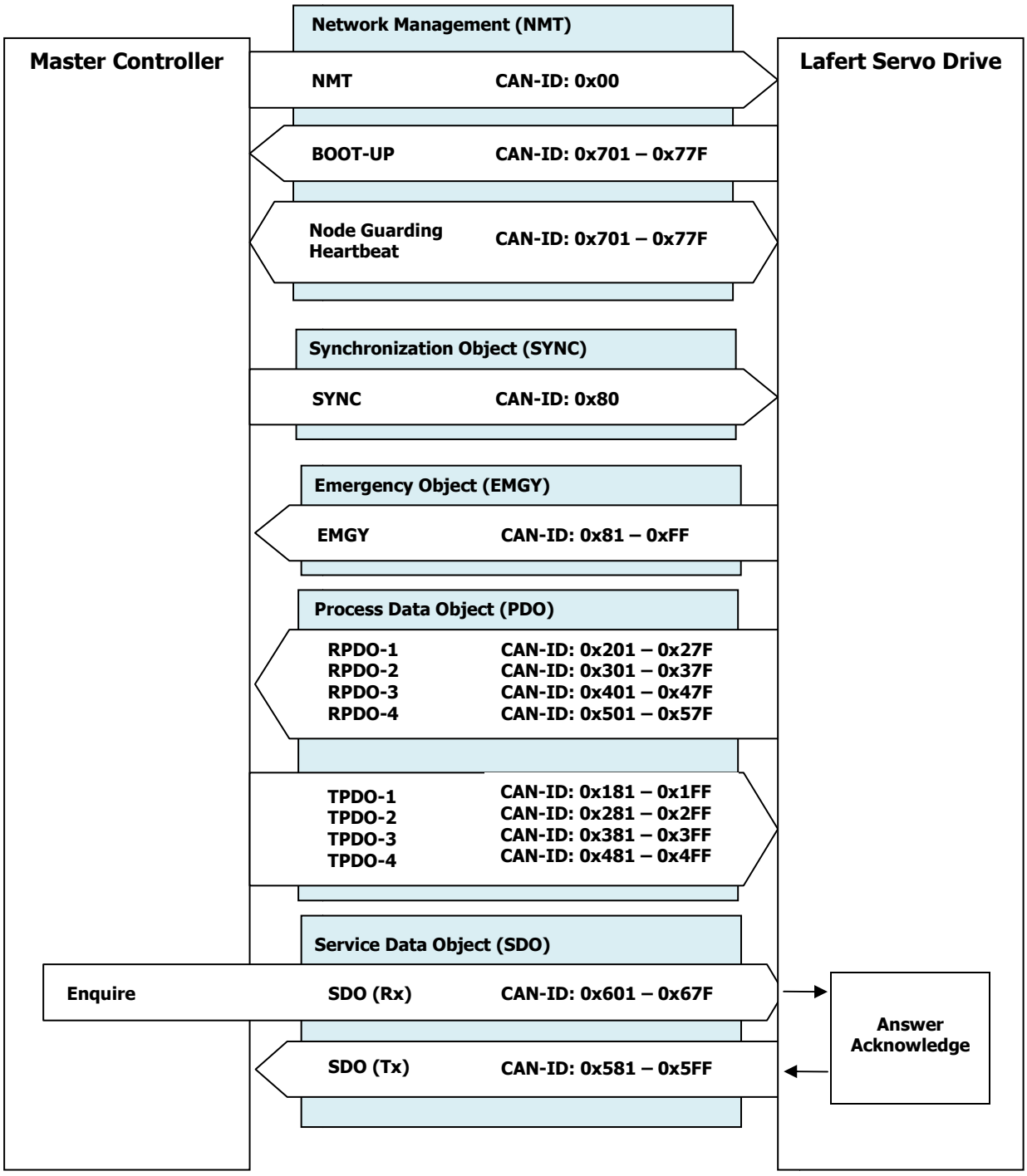


Figure 3 - communication between Master Controller and Drive

information

For Additional Information please refer to CiA DS301 standard.

CANopen makes available a simple and standardised possibility for accessing the parameters of the Lafert Drive (i.e. Target Speed or profile Acceleration). A unique number (index and sub-index) is assigned to each parameter (CANopen object). The totality of all adjustable parameters is contained in the object directory (OD).

There are 2 methods for accessing CANopen objects via the CAN bus:

- **Access via Service data object (SDO):** confirmed type of access where the Lafert Drive acknowledges every parameter access

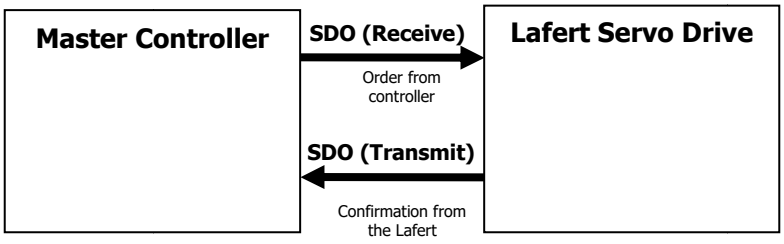


Figure 5 - SDO Communication

- **Access via Process data object (PDO):** unconfirmed type of access for which no acknowledgement takes place

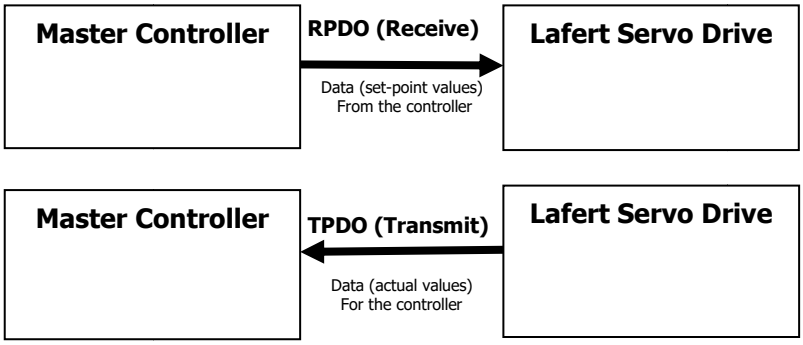


Figure 6 – PDO Communication

- **Abort Code via Service data object (SDO)**: case of an error when reading or writing (for example, because the written value is too large), the Lafert Servo Drive answers with an error message instead of the acknowledgement.

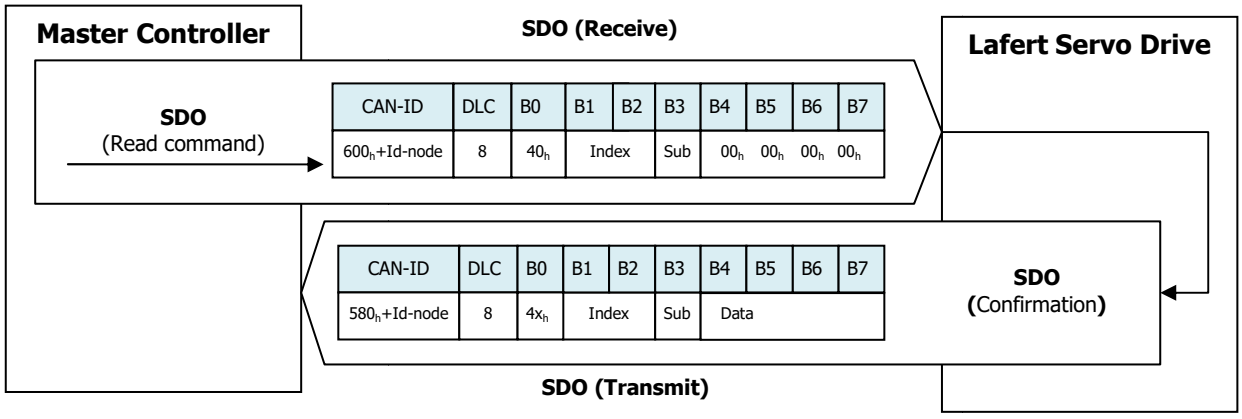


Figure 7 - Abort SDO Communication

SDO PROTOCOL

The SDO protocol is used for setting and for reading parameters device. The SDOs are used to implement access to the object dictionary. The communication is always initiated by the SDO client.

At the request of the client (Master Controller, PC Application, PLC - programmable logic controller) the drive makes data available.

The following communication protocols are supported:

- SDO Download Protocol
- SDO Upload Protocol
- SDO Abort Protocol

SDO Download Protocol (WRITE)

The SDO download service is used to configure the communication, device and manufacturer specific parameters.

SDO Download Message structure:

COB-ID	Request/ Respond	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
0x600+IdNode	Rx	8	0x2x	Index		Sub Index	Data LSB	Data	Data	Data MSB
0x580+IdNode	Tx	8	0x60	Index		Sub Index	0x00	0x00	0x00	0x00

Table 2- SDO Download Message Structure

SDO Download Message - Data Field:

D0	Description	Number of data bytes
0x22	Write Request (Initiate Domain Download)	-
0x23	Write Request (Initiate Domain Download)	4 bytes
0x27	Write Request (Initiate Domain Download)	3 bytes
0x2B	Write Request (Initiate Domain Download)	2 bytes
0x2F	Write Request (Initiate Domain Download)	1 byte
0x60	Write Response (Initiate Domain Download)	-

Table 3 - SDO Download Message Data Field

SDO Upload Protocol (READ)

The SDO upload service is used to read the communication, device and manufacturer specific parameters SDO.

SDO Upload Message structure:

COB-ID	Request/ Respond	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
0x600+IdNode	Rx	8	0x40	Index		Sub Index	0x00	0x00	0x00	0x00
0x580+IdNode	Tx	8	0x4x	Index		Sub Index	Data LSB	Data	Data	Data MSB

Table 4 - SDO Upload Message Structure

SDO Upload Message - Data Field:

D0	Description	Number of data bytes
0x40	Read Request (Initiate Domain Upload)	-
0x43	Read Response (Initiate Domain Upload)	4 bytes
0x47	Read Response (Initiate Domain Upload)	3 bytes
0x4B	Read Response (Initiate Domain Upload)	2 bytes
0x4F	Read Response (Initiate Domain Upload)	1 byte

Table 5 - SDO Upload Message Data Field

SDO Abort Code

The SDO Abort service is used to communicate fault by download or upload service.
If the SDO fails then the CANOpen does not respond with the corresponding SDO message, but it uses the SDO abort protocol.
In the Abort message there is the data abort code that recognizes the kind of fault.

SDO Abort Message structure:

COB-ID	Request/ Respond	DLC	Data							
			D0	D1	D2	D3	D4	D5	D6	D7
0x580+IdNode	**Tx**	**8**	0x80	Index		Sub Index	Abort Code			

Table 6 - SDO Abort Message Structure

The Abort Code as defined in follow table, It is encoded as UNSIGNED32 value.

Abort Code	Description
$0504\ 0000_h$	SDO protocol timed out.
$0504\ 0001_h$	Client/server command specifier not valid or unknown.
$0504\ 0002_h$	Invalid block size (block mode only).
$0504\ 0003_h$	Invalid sequence number (block mode only).
$0504\ 0004_h$	CRC error (block mode only).
$0504\ 0005_h$	Out of memory.
$0601\ 0000_h$	Unsupported access to an object.
$0601\ 0001_h$	Attempt to read a write only object.
$0601\ 0002_h$	Attempt to write a read only object.
$0602\ 0000_h$	Object does not exist in the object dictionary.
$0604\ 0041_h$	Object cannot be mapped to the PDO.
$0604\ 0042_h$	The number and length of the objects to be mapped would exceed PDO length.
$0604\ 0043_h$	General parameter incompatibility reason.
$0604\ 0047_h$	General internal incompatibility in the device.
$0606\ 0000_h$	Access failed due to an hardware error.
$0607\ 0010_h$	Data type does not match, length of service parameter does not match
$0607\ 0012_h$	Data type does not match, length of service parameter too high
$0607\ 0013_h$	Data type does not match, length of service parameter too low

0609 0011$_h$	Sub-index does not exist.
0609 0030$_h$	Invalid value for parameter (download only).
0609 0031$_h$	Value of parameter written too high (download only)
0609 0032$_h$	Value of parameter written too low (download only).
0609 0036$_h$	Maximum value is less than minimum value.
060A 0000$_h$	Operation not allowed in this state
060A 0023$_h$	Resource not available: SDO connection
0800 0000$_h$	General error
0800 0020$_h$	Data cannot be transferred or stored to the application.
0800 0021$_h$	Data cannot be transferred or stored to the application because of local control.
0800 0022$_h$	Data cannot be transferred or stored to the application because of the present device state.
0800 0023$_h$	Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error).
0800 0024$_h$	No data available
0800 0030$_h$	Data cannot be written because it need STORE command and a reset or power cycle

Table 7 - SDO Abort Code Description

PDO PROTOCOL

The PDO protocol is used to process real time data among various nodes. This communication Object uses the unconfirmed communication service. Data transferring will be limited to 1 to 8 bytes and there is no hand-shake restriction in PDO communication.

2 Objects in object dictionary are used for each PDO:

- PDO communication parameter: it contains the PDO configuration COB-ID, transferring-type, restriction time and cycle of timer used by PDO (objects 1400$_h$, 1401$_h$, 1402$_h$, 1403$_h$, 1800$_h$, 1801$_h$, 1802$_h$, 1803$_h$)
- PDO mapping parameter: it contains a list of objects dictionary. These object are mapped into PDO, include their data length in bits (objects 1600$_h$, 1601$_h$, 1602$_h$, 1603$_h$, 1A00$_h$, 1A01$_h$, 1A02$_h$, 1A03$_h$). Producers and clients must know this mapping.

The Lafert Servo Drive has a default mapping that it can be changed when the network initializes. The drive supports the dynamic PDO mapping and changing of mapping can be done in the state PRE-OPERATIONAL.

Caution

It is possible to change the mapping in OPERATIONAL state but is not suggested. If the client change the mapping during OPERATIONAL state, then it is responsible for the data consistency.

information

The PDO protocol are available in OPERATIONAL state

The PDO mapping has 2 rules:

- Each PDO could be mapped into 4 objects
- The length of each PDO will be no more than 64 bits

There are multiple ways to transmit PDO:

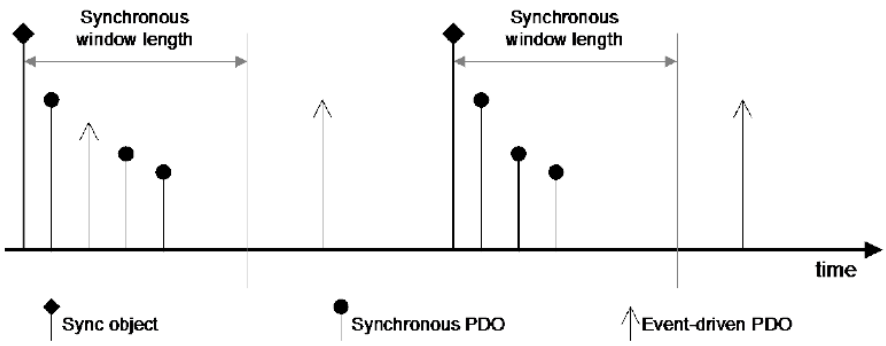
- Synchronous (synchronous by receiving SYNC object)
- Asynchronous (transmit triggered by special object event regulated in sub-object protocol)

The following PDO transmission modes are distinguished:

- Synchronous transmission
- Event-driven transmission

In order to synchronize CANopen devices a synchronization object (SYNC object) is transmitted periodically by a synchronization application. The SYNC object is represented by a pre-defined communication object.
In following picture shows the principle of synchronous and event driven transmission is shown.

Synchronous PDOs are transmitted within a pre-defined time-window immediately after the SYNC object.



Three message-triggering modes are distinguished:

- Event- and timer-driven: message transmission is either triggered by the occurrence of an application-specific event specified in the device profile, application profile or manufacturer-specific, or if a specified time (event-time) has elapsed without occurrence of an event.
- Remotely requested: The transmission of an event-driven PDO is initiated on receipt of a RTR initiated by a PDO consumer.
- Synchronously triggered: Message transmission is triggered by the occurrence of the SYNC object. The trigger condition is the number of Sync and optionally an internal event.

Receive PDO (RPDO)

1400_h – 1403_h: RPDO Communication Objects

These objects contain the communication parameters for the PDOs the device is able to receive.

- Sub-index 00_h contains the number of valid entries within the communication record
- Sub-index 01_h is the COB-ID of the PDO, this entry has been defined ad UNSIGNED32 in order to define for 11-bit CAN identifiers (CAN 2.A) as well as for 29 bit identifiers (CAN 2.0B)

31 (MSB)	30	29	28 ... 11	10 ... 0 (LSB)
Valid	Reserved	frame	0000_h	11-bit CAN-ID
			29-bit CAN-ID	

Name	BIT	VALUE	Meaning
Valid	31	0_b 1_b	PDO exist / is valid PDO does not exist /is not valid
Reserved	30	-	-
frame	29	0_b 1_b	Message standard 11 bit (CAN 2.0 A) Message extended 29 bit (CAN 2.0 B)

29-bit CAN-ID	11 - 28	x	If Bit 29 = 1 : bits 28-11 of COB-ID message extended
11-bit CAN-ID	0 – 10	x	bits 10-0 of COB-ID

information

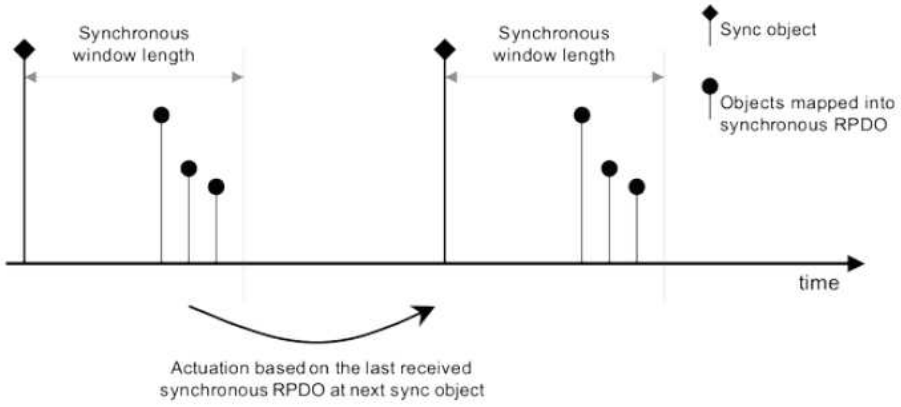
The PDO valid or no allows to select which PDOs are used in the OPERATIONAL state.

- Sub-index 02_h is the transmission type, it defines the reception character of the PDO. The following table describes the usage entry.

Transmission Type		Meaning
0 - 240	00_h – $F0_h$	synchronous
241 - 253	00_h – FD_h	reserved
254	0xFE	event-driven (manufacturer-specific)
255	0xFF	event-driven (device profile and application profile specific)

Synchronous means that the CANopen device shall actuate the received data with the reception of the next SYNC

Event-driven means that the PDO may be received at any time. The CANopen device will actualize the data immediately.



- Sub-index 03_h contains the inhibit time. The value is defined as multiple of 100 µs. The value of 0 shall disable the inhibit time. It is not allowed to change the value while the PDO exists (bit 31 of sub-index 01h is set to 0b). The RPDO may use the time implementation specific.
- Sub-index 04_h is reserved. It shall not be implemented; in this case read or write access leads to the SDO abort transfer service in this case read or write access leads to the SDO abort transfer service (abort code: 0609 0011h).
- Sub-index 05_h contains the event-timer. The value is defined as multiple of 1 ms. The value of 0 shall disable the event-timer. The RPDO may use the time for deadline monitoring. The deadline monitoring is activated within the next reception of an RPDO after configuring the event-timer. A timeout results in an indication to the local application.
- Sub-index 06_h contains the SYNC start value. This is not used by RPDOs. It shall not be implemented; in this case read or write access shall lead to the SDO abort transfer service (abort-code: 0609 0011h). (it is not available)

1600h – 1603h: RPDO Mapping Parameters

These objects contain the mapping for the PDOs device is able to receive.

Sub-index 00_h contains the number of valid object entries within the mapping record or a specific value. The number of valid object entries shall be the number of the application objects that shall be received with the corresponding RPDO.

Value	Description

00_h	Mapping disabled
01_h	Sub-index 01h valid
02_h	Sub-index 01h and 02h valid
...	...

Sub-index from 01h to 40h contains the information of the mapped application objects. The object describes the content of the PDO by their index, sub-index and length. The length contains the length of the application object in bit. This may be used to verify the mapping.

Byte: MSB		LSB
Index (16 bit)	Sub-Index (8bit)	Object Length (8 bit)

If the change of the PDO mapping cannot be executes (e.g. the PDO length is exceeded or the SDO client attempts to map an object that cannot be mapped) the drive responds with an Abort Code SDO transfer service.

RPDO Default

RPDOs are CAN frames identified by their 11-bit header.

- RPDO1: 0x200 + Node ID
- RPDO2: 0x300 + Node ID
- RPDO3: 0x400 + Node ID
- RPDO4: 0x500 + Node ID

The following tables describe the default mapping for RPDO:

Index	SubIndex	Description	Type	Attr.	Dafault Value	Description
Receive Process Data Object (RPDO1)						
1400h	0	Receives 1st PDO	UNSIGNED8	ro	5	Number of Entries
	1	COB ID used by PDO1	UNSIGNED32	rw	0x200+NodeID	PDO enabled
	2	Transmission Type	UNSIGNED8	rw	0xFE	Asynchronous Man. Spec.
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1600h	0	N receive PDO mapping	UNSIGNED8	rw	3	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x6040 0010	Control word
	2	2 - application object	UNSIGNED32	rw	0x6060 0008	Mode of operation
	3	3 - application object	UNSIGNED32	rw	0x60FE 0120	Digital Outputs
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-
	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-
Receive Process Data Object (RPDO2)						
1401h	0	Receives 2nd PDO	UNSIGNED8	ro	5	Number of Entries
	1	COB ID used by PDO2	UNSIGNED32	rw	0x300+NodeID	PDO enabled
	2	Transmission Type	UNSIGNED8	rw	0xFE	Asynchronous Man. Spec.
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1601h	0	N receive PDO mapping	UNSIGNED8	rw	2	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x6040 0010	Control word
	2	2 - application object	UNSIGNED32	rw	0x607A 0020	Target Position
	3	3 - application object	UNSIGNED32	rw	0	-
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-

	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-
Receive Process Data Object (RPDO3)						
1402h	0	Receives 3rd PDO	UNSIGNED8	ro	5	Number of Entries
	1	COB ID used by PDO3	UNSIGNED32	rw	0x400+NodeID	PDO enabled
	2	Transmission Type	UNSIGNED8	rw	0xFE	Asynchronous Man. Spec.
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1602h	0	N receive PDO mapping	UNSIGNED8	rw	2	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x6040 0010	Control word
	2	2 - application object	UNSIGNED32	rw	0x60FF 0020	Target Velocity
	3	3 - application object	UNSIGNED32	rw	0	-
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-
	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-
Receive Process Data Object (RPDO4)						
1403h	0	Receives 4th PDO	UNSIGNED8	ro	5	Number of Entries
	1	COB ID used by PDO4	UNSIGNED32	rw	0x500+NodeID	PDO enabled
	2	Transmission Type	UNSIGNED8	rw	0xFE	Asynchronous Man. Spec.
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1603h	0	N receive PDO mapping	UNSIGNED8	rw	2	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x6040 0010	Control word
	2	2 - application object	UNSIGNED32	rw	0x6071 0010	Target Torque
	3	3 - application object	UNSIGNED32	rw	0	-
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-
	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-

Table 8 - RPDO Description

Mapping default RPDO 1: Controls PDS FSA – mandatory

Index	Sub-Index	Name	Default Value
1600h		**Receive PDO 1**	**COB-ID**
	0	Number of mapped objects	3
	1	Control word	6040 0010h
	2	Mode of operation	6060 0008h
	3	Digital Output	60FE 0120h

Table 9 - RPDO1 Mapping

Mapping default RPDO 2: Controls PDS FSA and target position (pp) - optional

Index	Sub-Index	Name	Default Value
1601h		**Receive PDO 2**	**COB-ID**
	0	Number of mapped objects	2
	1	Control word	6040 0010h
	2	Target Position	607A 0020h

Table 10 - RPDO2 Mapping

Mapping default RPDO 3: Controls PDS FSA and target velocity (pv) - optional

Index	Sub-Index	Name	Default Value
1602h		**Receive PDO 3**	**COB-ID**
	0	Number of mapped objects	2

	1	Control word	6040 0010h
	2	Target Velocity	60FF 0020h

Table 11 - RPDO3 Mapping

Mapping default RPDO 4: Controls PDS FSA and target torque (tq)- optional

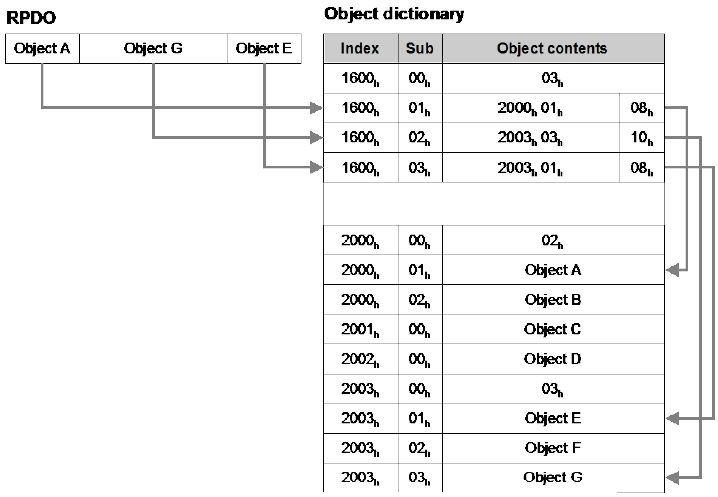
Index	Sub-Index	Name	Default Value
1603h		**Receive PDO 4**	**COB-ID**
	0	Number of mapped objects	2
	1	Control word	6040 0010h
	2	Target Torque	6071 0010h

Table 12 - RPDO4 Mapping

Re-Mapping Procedure:

The following procedure shall be used for re-mapping, which may take place during the NMT state Pre-operational:

1) For changing the PDO mapping first the PDO has to invalidate the PDO. Destroy the bit *valid* into sub-index 01_h of "RPDO Communication" objects (1400_h, 1401_h, 1402_h e 1403_h). The 31 bit must be set to 0.
2) Disable mapping PDO setting 0 into sub-index 00_h of "Mapping Parameters" object (1600_h, 1601_h, 1602_h e 1603_h). This will disable PDO.
3) Modify mapping by changing the values of the corresponding sub-indices. Write in sub-index correspondent the description of the object (Index, Sub-Index and Length)
4) Set the sub-index 00_h of PDO coordinated mapping parameter (objects 1600_h, 1601_h, 1602_h e1603_h) as legal number (number of PDO's mapping objects). This will enable new mapping.
5) Create RPDO by setting bit valid to 0 of sub-Index 01_h (COB-ID) of "communication object" (objects 1400_h, 1401_h, 1402_h e 1403_h) the according RPDO communication parameter.
6) PDO mapping completing



If during step 3 the drive detects that the Index and sub-Index of the mapped objects does not exist or the object cannot be mapped, the device responds with the SDO abort transfer service (abort code: 0602 0000h or 0604 0041h).

If during step 4 the drive detects that the RPDO mapping is not valid or not possible the CANopen device shall respond with the SDO abort transfer service (abort code: 0602 0000h or 0604 0042h).

If the drive receives a PDO that has more data bytes than the number of mapped If the CANopen device receives a PDO that is having more data bytes than the number of mapped data bytes is (length), then the CANopen device shall use the first data bytes up to the length and may be initiate the EMCY write service, if supported.

Transmit PDO (TPDO)

1800_h – 1803_h: TPDO Communication Objects

These objects contain the communication parameters for the PDOs the device is able to transmit.

- Sub-index 00_h contains the number of valid entries within the communication record
- Sub-index 01_h is the COB-ID of the PDO, this entry has been defined ad UNSIGNED32 in order to define for 11-bit CAN identifiers (CAN 2.A) as well as for 29 bit identifiers (CAN 2.0B)

31 (MSB)	30	29	28 – 11	10 – 0 (LSB)
Valid	RTR	frame	0000_h	11-bit CAN-ID
			29-bit CAN-ID	

Name	BIT	VALUE	Meaning
Valid	31	0_b 1_b	PDO exist / is valid PDO does not exist /is not valid
RTR	30	0_b 1_b	RTR allowed RTR not allowed
frame	29	0_b 1_b	Message standard 11 bit (CAN 2.0 A) Message extended 29 bit (CAN 2.0 B)
29-bit CAN-ID	11 - 28	x	If Bit 29 = 1 : bits 28-11 of COB-ID message extended
11-bit CAN-ID	0 – 10	x	bits 10-0 of COB-ID

- Sub-index 02_h is the transmission type, it defines the transmitting character of the PDO. The following table describes the usage entry.

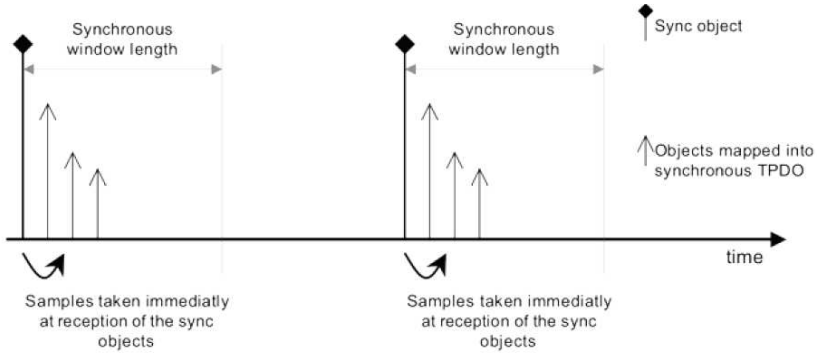
Transmission Type		Meaning
0	00_h	synchronous (acyclic)
1 - 241	01_h – $F0_h$	synchronous (cyclic every N sync)
241 - 251	$F0_h$ – FB_h	reserved
252	FC_h	RTR-only (synchronous)
253	FD_h	RTR-only (event-driven)
254	FE_h	event-driven (manufacturer-specific)
255	FF_h	event-driven (device profile and application profile specific)

Synchronous means that the PDO is transmitted after the SYNC. The CANopen device will start sampling of the data with the reception of the SYNC. In case it is acyclic the CANopen device internal event is given and with the next SYNC he sampling is started and the PDO is transmitted afterwards. In case it is cyclic the sampling is started with the reception of every SYNC, every 2nd SYNC, every 3rd SYNC, and s.o. depending on the given value and the PDO is transmitted afterwards.

RTR-only means that the PDO is not transmitted normally it shall be requested via RTR. In case it is synchronous the CANopen device will start sampling with the reception of every SYNC and then will buffer the PDO In case it is event-driven the CANopen device will start sampling with the reception of the RTR and will transmit the PDO immediately.

- Event-driven means that the PDO may be transmitted at any time based on the occurrence of a CANopen device internal event. The definition of the event does not fall into the scope of this specification and may be specified in device profiles and application profiles.

- Sub-index 03h contains the inhibit time. The time is the minimum interval for PDO transmission if the transmission type is set to FE_h and FF_h. The value is defined as multiple of 100 µs. The value of 0 shall disable the inhibit time. The value shall not be changed while the PDO exists (bit 31 of sub-index 01h is set to 0_b)
- Sub-index 04_h is reserved. It does shall not be implemented; in this case read or write access leads to the SDO abort transfer service (abort code: 0609 0011_h).



- Sub-index 05_h contains the event-timer. The time is the maximum interval for PDO transmission if the transmission type is set to FE_h and FF_h. The value is defined as multiple of 1 ms. The value of 0 shall disable the event-timer.
- Sub-index 06_h contains the SYNC start value. The SYNC start value of 0 shall indicate that the counter of the SYNC message shall not be processed for this PDO. The SYNC start value 1 to 240 shall indicate that the counter of the SYNC message shall be processed for this PDO. In case the counter of the SYNC message is not enabled (see 7.5.2.22) sub-index 06_h shall be ignored. The SYNC message of which the counter value equals the SYNC start value shall be regarded as the first received SYNC message. The value shall not be changed while the PDO exists (bit 31 of sub-index 01_h is set to 0_b). (it is not available)

$1A00_h$ – $1A03_h$: TPDO Mapping Parameters

These objects contain the mapping for the PDOs device is able to transmit.

Sub-index 00_h contains the number of valid object entries within the mapping record or a specific value. The number of valid object entries shall be the number of the application objects that shall be received with the corresponding RPDO.

Value	Description
00_h	Mapping disabled
01_h	Sub-index 01h valid
02_h	Sub-index 01h and 02h valid
...	...

Sub-index from 01h to 40h contains the information of the mapped application objects. The object describes the content of the PDO by their index, sub-index and length. The length contains the length of the application object in bit. This may be used to verify the mapping.

Byte: MSB		LSB
Index (16 bit)	Sub-Index (8bit)	Object Length (8 bit)

If the change of the PDO mapping cannot be executes (e.g. the PDO length is exceeded or the SDO client attempts to map an object that cannot be mapped) the drive responds with an Abort Code SDO transfer service.

TPDO Mapping Default

- TPDO1: 0x180 + Node ID
- TPDO2: 0x280 + Node ID
- TPDO3: 0x380 + Node ID
- TPDO4: 0x480 + Node ID

The following tables describe the default mapping for TPDO:

Index	SubIndex	Description	Type	Attr.	Dafault Value	Description
Transmit Process Data Object (TPDO1)						
1800h	0	Transmit 1st PDO	UNSIGNED8	ro	5	Number of Entries
	1	COB ID used by PDO1	UNSIGNED32	rw	0x180+NodeID	PDO enabled
	2	Transmission Type	UNSIGNED8	rw	0xFD	Asynchronous RTR
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1A00h	0	N transmit PDO mapping	UNSIGNED8	rw	3	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x6041 0010	Status word
	2	2 - application object	UNSIGNED32	rw	0x60610008	Mode Of Operation Display
	3	3 - application object	UNSIGNED32	rw	0x60FD0020	Digitals Inputs
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-
	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-
Transmit Process Data Object (TPDO2)						
1801h	0	Transmit 2nd PDO	UNSIGNED8	ro	5	Number of Entries
	1	COB ID used by PDO2	UNSIGNED32	rw	0x280+NodeID	PDO enabled
	2	Transmission Type	UNSIGNED8	rw	0xFD	Asynchronous RTR
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1A01h	0	N transmit PDO mapping	UNSIGNED8	rw	2	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x60410010	Status word
	2	2 - application object	UNSIGNED32	rw	0x60640020	Position Actual Value
	3	3 - application object	UNSIGNED32	rw	0	-
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-
	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-
Transmit Process Data Object (TPDO3)						
1802h	0	Transmit 3rd PDO	UNSIGNED8	ro	5	Number of Entries
	1	COB ID used by PDO3	UNSIGNED32	rw	0x380+NodeID	PDO enabled
	2	Transmission Type	UNSIGNED8	rw	0xFD	Asynchronous RTR
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1A02h	0	N transmit PDO mapping	UNSIGNED8	rw	2	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x60410010	Status word
	2	2 - application object	UNSIGNED32	rw	0x606C0020	Velocity Actual Value
	3	3 - application object	UNSIGNED32	rw	0	-
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-
	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-
Transmit Process Data Object (TPDO4)						
1803h	0	Transmit 4th PDO	UNSIGNED8	ro	3	Number of Entries
	1	COB ID used by PDO4	UNSIGNED32	rw	0x480+NodeID	PDO enabled

	2	Transmission Type	UNSIGNED8	rw	0xFD	Asynchronous RTR
	3	Inhibit Time	UNSIGNED16	rw	0x5	units (100us)
	4	Compatibility Entry	UNSIGNED8	rw	0	disabled
	5	Event Timer	UNSIGNED16	rw	0	disabled
1A03h	0	N transmit PDO mapping	UNSIGNED8	rw	2	Number of Entries
	1	1 - application object	UNSIGNED32	rw	0x60410010	Status word
	2	2 - application object	UNSIGNED32	rw	0x60770010	Torque Actual Value
	3	3 - application object	UNSIGNED32	rw	0	-
	4	4 - application object	UNSIGNED32	rw	0	-
	5	5 - application object	UNSIGNED32	rw	0	-
	6	6 - application object	UNSIGNED32	rw	0	-
	7	7 - application object	UNSIGNED32	rw	0	-
	8	8 - application object	UNSIGNED32	rw	0	-

Table 13 - TPDO Description

Mapping default TPDO 1: Specifies PDS FSA status – mandatory

Index	Sub-Index	Name	Default Value
1A00h		**Transmit TDO 1**	**COB-ID**
	0	Number of mapped objects	3
	1	Status word	6041 0010h
	2	Mode Of Operation Display	6061 0008h
	3	Digital Input	60FD 0020h

Table 14 - TPDO1 Mapping

Mapping default TPDO 2: Specifies PDS FSA status and current position (pp) - optional

Index	Sub-Index	Name	Default Value
1A01h		**Transmit TDO 2**	**COB-ID**
	0	Number of mapped objects	2
	1	Status word	6041 0010h
	2	Position Actual Value	6064 0020h

Table 15 - TPDO2 Mapping

Mapping default TPDO 3: Specifies PDS FSA status and current current velocity (pv) - optional

Index	Sub-Index	Name	Default Value
1A02h		**Transmit TDO 3**	**COB-ID**
	0	Number of mapped objects	2
	1	Status word	6041 0010h
	2	Velocity Actual Value	606C 0020h

Table 16 - TPDO3 Mapping

Mapping default TPDO 4: Specifies PDS FSA status and current torque (tq) - optional

Index	Sub-Index	Name	Default Value
1A03h		**Transmit TDO 4**	**COB-ID**
	0	Number of mapped objects	2
	1	Status word	6041 0010h
	2	Torque Actual Value	6077 0010h

Table 17–TPDO4 Mapping

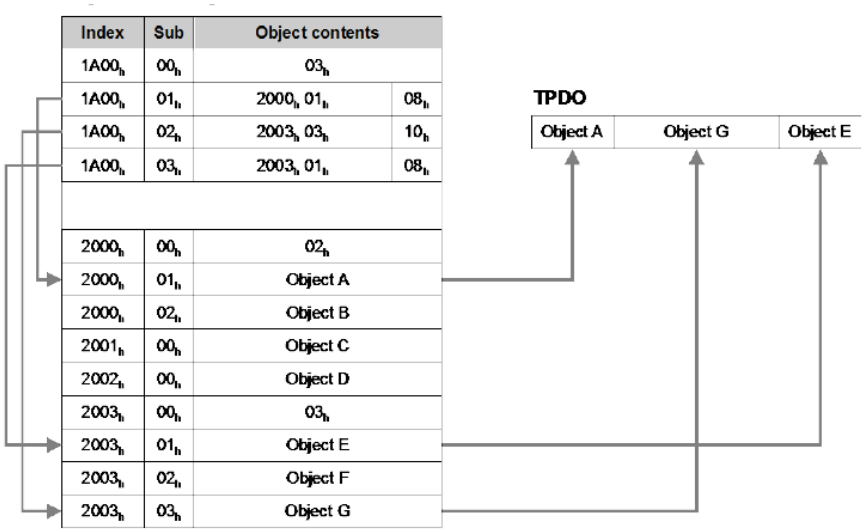
Re-Mapping Procedure:

The following procedure shall be used for re-mapping, which may take place during the NMT state Pre-operational:

1) For changing the PDO mapping first the PDO has to invalidate the PDO. Destroy the bit valid into sub-index 01h of "TPDO Communication" objects (1800_h, 1801_h, 1802_h and 1803_h). The 31 bit must be set to 0.

2) Disable mapping PDO setting 0 into sub-index 00h of "Mapping Parameters" object (1A00h, 1A01h, $1A02_h$ and $1A03_h$). This will disable PDO.

3) Modify mapping by changing the values of the corresponding sub-indices. Write in sub-index correspondent the description of the object (Index, Sub-Index and Length)
4) Set the sub-index 00h of PDO coordinated mapping parameter (1A00h, $1A01_h$, $1A02_h$ and $1A03_h$) as legal number (number of PDO's mapping objects). This will enable new mapping.
5) Create RPDO by setting bit valid to 0 of sub-Index 01h (COB-ID) of "communication object" (1800_h, 1801_h, 1802_h and 1803_h) the according TPDO communication parameter.
6) PDO mapping completing

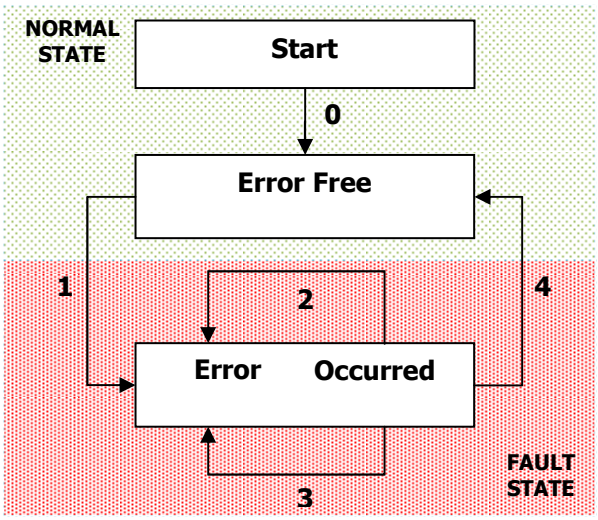


If during step 3 the CANopen device detects that index and sub-index of the mapped object does not exist or the object cannot be mapped the CANopen device shall respond with the SDO abort transfer service (abort code: 0602 0000h or 0604 0041h).

If during step 4 the CANopen device detects that the RPDO mapping is not valid or not possible the CANopen device shall respond with the SDO abort transfer service (abort code: 0602 0000h or 0604 0042h).

Emergency Message (EMCY)

The Lafert Servo Drive monitors the function of internal modules and of the firmware.
Whenever an error occurs, the parameterised error response is initiated and the corresponding EMCY message is transmitted.
The latest error message is always stored here in Error Code object ($603F_h$:0_h).
Also, it is in the highest error memory slot (1003_h: 01_h), the error memory always saves the 15 most recent error messages that can also be read out.



The following status transitions are possible:

Transition	Cause	Description
0	Initialisation completed	There is no error. The drive sends the error code 0000_h (Error reset/No error) and the state of CAN (8170_h)
1	Error occurs	No error was present and a new error occurs. The drive goes to Fault State. Verify the diagnostic state and the Emergency message.
2	Error acknowledgment not successful	Not all causes of error have been remedied and an error acknowledgement was performed.
3	New error occurs	There is an error and a new error occurs. An EMCY message with the error code for the new error (1003_h: 01_h, standard error field 1) is written.
4	Error acknowledgment successful	All causes of error have been remedied and an error acknowledgement was performed. The EMCY message was transmitted with error code 0000_h (Error reset/No error).

Emergency objects are triggered by the incident of a CANopen device internal error situation and are transmitted onto the network.

Emergency objects are suitable for error alerts.

Emergency message structure by CanOpen DSP402:

COB-ID	Rx/Tx	DLC	Byte							
			0	1	2	3	4	5	6	7
0x80+IdNode	Tx	8	Error Code		Reg	Manufacturer specific error field				
			E0	E1	R0	M0	M1	M2	M3	M4

Table 18 - Emergency Message Structure

Error Code field standard by DS301:

Error Code	Name	Description
0x0000	NO ERROR	error Reset or No Error
0x1000	GENERIC ERROR	Generic Error
0x2000	CURRENT	Current
0x2100	CURRENT INPUT	Current , device input side
0x2200	CURRENT INSIDE	Current inside the device
0x2300	CURRENT OUTPUT	Current , device output side
0x3000	VOLTAGE	Voltage
0x3100	VOLTAGE MAINS	Mains Voltage

0x3200	VOLTAGE INSIDE	Voltage inside the device
0x3300	VOLTAGE OUTPUT	Output Voltage
0x4000	TEMPERATURE	Temperature
0x4100	TEMP AMBIENT	Ambient Temperature
0x4200	TEMP DEVICE	Device Temperature
0x5000	HARDWARE	Device Hardware
0x6000	SOFTWARE DEVICE	Device Software
0x6100	SOFTWARE INTERNAL	Internal Software
0x6200	SOFTWARE USER	User Software
0x6300	DATA SET	Data Set
0x7000	ADDITIONAL MODULE	Additional Modules
0x8000	MONITORING	Monitoring
0x8100	COMMUNICATION	Communication
0x8200	PROTOCOL ERROR	Protocol Error
0x9000	EXTERNAL ERROR	External Error
0xF000	ADDITIONAL FUNC	Additional Functions
0xFF00	DEVICE SPECIFIC	Device specific

Table 19 - Emergency Error Code

Register field standard by DS301: CANopen device maps internal errors into this object. The bit 0 is the generic error and it is mandatory when error fault is occurred, the others bits specific different type error.

Reg	BIT	NAME	Description
0x00		NO ERROR	none error
0x01	1	REGISTER GENERIC ERROR	generic error
0x02	2	REGISTER CURRENT	current
0x04	3	REGISTER VOLTAGE	voltage
0x08	4	REGISTER TEMPERATURE	temperature
0x10	5	REGISTER COMMUNICATION ERROR	communication error (overrun, error state)
0x20	6	REGISTER DEVICE PROFILE	device profile specific
0x40	7	REGISTER RESERVED	reserved (always 0)
0x80	8	REGISTER MANUFACTURER	manufacturer specific

Table 20- Emergency Register Field

The following table defines the alarms group (Fault / Warning) implemented in Lafert with CANopen code.

The "Led Code" column describes the number of blinking of leds.
For example [x, y] = 6,2 means the green Led blinks 6 times, after that, the yellow led blinks 2 times.

"BLINK" code [x]

"BLINK" code [y]

The "Error Code" describes the univocal value of alarm. The last alarm occurred can be read with $603F_h$ object "Error Code".

Some alarms have the sub-codes defined by manufacturer. The column meaning describes the Manufacturer specific error field.

The alarm can be Fault (F) or warning (W), if it is a fault the drive will stop.

Error	Error Code	Description	Meaning	F - W	Led Code
NO ERROR	**0x0000**	**No Error**	The Fault Reset command has been executed or there was a reset with power cycle	-	-

GENERIC ERROR	**0X1000**	**Generic Error**	Generic Error	**-**	**-**
		ALARM CURRENT			
SHORT CIRCUIT MOTOR	**0x2340**	**Short circuit (motor-side)**	Alarm Over Current has been occurred	**F**	**3,1**
LOAD LEVEL FAULT	**0x2350**	**Load level fault (I2t, thermal state)**	Alarm Over Current with integral i^2t (Over Load)	**F**	**5,2**
	0x2351	Warning (I2t, thermal state)	Warning Limitation i^2t (Over Load)	**W**	**-**
	0x2352	Load Level (i2t) not rearmed	Alarm Over Current with integral i^2t (Over Load) not Rearmed	**F**	**5,2**
		ALARM VOLTAGE			
OVER VOLTAGE	**0x3210**	**DC link over-voltage**	Over Voltage alarm has been occurred	**F**	**4,2**
DC LINK UNDER VOLTAGE	**0x3220**	**DC link under-voltage**	Under Voltage alarm has been occurred	**F**	**4,1**
		ALARM TEMPERATURE			
TEMPERATURE DRIVE	**0x4300**	**Temperature Drive**	Over Temperature Heat Sink (value depends by Manufacturer)	**F**	**1,1**
	0x4301	Warning temperature drive	Warning temperature drive (value depends by Manufacturer)	**W**	**-**
	0x4310	Excess temperature drive	Heat Sink Temperature too high of maximum Range	**F**	**1,3**
	0x4320	Too low temperature drive	Heat Sink Temperature too low of minimum Range	**F**	**1,3**
TEMPERATURE INTERNAL 1 – BOARD	**0x4500**	**Temperature Logic Board**	Over Board Temperature (> 68°C)	**F**	**1,4**
	0x4501	Warning Logic Board temperature	Warning Logic Board temperature (> 63°C)	**W**	**-**
	0x4510	Excess Logic Board temperature	Board Temperature too high of maximum Range	**F**	**1,5**
	0x4520	Too low Logic Board temperature	Board Temperature too low of minimum Range	**F**	**1,5**
TEMPERATURE EXTERNAL 1 - MOTOR	**0x4A00**	**Temperature Motor**	Over Motor Temperature (> 140°C)	**F**	**1,10**
	0x4A01	Warning temperature Motor	Warning Motor Temperature (> 130°C)	**W**	**-**
	0x4A10	Excess temperature Motor	Motor Temperature too high of maximum Range	**F**	**1,6**
	0x4A20	Too low temperature Motor	Motor Temperature too low of minimum Range	**F**	**1,6**
		ALARM HARDWARE			
INPUT STAGES	**0x5430**	**Input stages**	Generic Input Stages	**-**	**-**
	0x5431	Offset Sensor	Offset Sensor	**F**	**3,10**
HARDWARE MEMORY	**0x5500**	**Hardware Memory**	Generic Hardware Memory	**-**	**-**
	0x5501	HardwareError Write EEprom: Vbus too Low	Write is not possible because the Bus Voltage is too low to guarantee the writing complete	**F**	**5,3**
HARDWARE MEMORY E^2PROM - USER	**0x5530**	**E^2PROM**	Generic Error E^2prom	**-**	**-**
	0x5531	E^2prom General Error	Generic Error E^2prom Writing	**F**	**6,1**
	0x5532	E^2prom Error Parameter 1	Error Writing E^2prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5533	E^2prom Error Parameter 2	Error Writing E^2prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5534	E^2prom Error Parameter 3	Error Writing E^2prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5535	E^2prom Error Parameter 4	Error Writing E^2prom Parameters (contact Manufacturer)	**F**	**6,1**

	0x5536	E²prom Error Parameter 5	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5537	E²prom Error Parameter 6	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5538	E²prom Error Parameter 7	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5539	E²prom Error Parameter 8	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x553A	E²prom Error Parameter 9	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x553B	E²prom Error Parameter 10	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x553C	E²prom Error Parameter 11	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x553D	E²prom Error Parameter 12	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x553E	E²prom Error Parameter 13	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x553F	E²prom Error Parameter 14	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5540	E²prom Error Parameter 15	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5541	E²prom Error Parameter 16	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5542	E²prom Error Parameter 17	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5543	E²prom Error Parameter 18	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5544	E²prom Error Parameter 19	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5545	E²prom Error Parameter 20	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5546	E²prom Error Parameter 21	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5547	E²prom Error Parameter 22	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5548	E²prom Error Parameter 23	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5549	E²prom Error Parameter 24	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x554A	E²prom Error Parameter 25	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x554B	E²prom Error Parameter 26	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x554C	E²prom Error Parameter 27	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x554D	E²prom Error Parameter 28	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x554E	E²prom Error Parameter 29	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x554F	E²prom Error Parameter 30	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5550	E²prom Error Parameter 31	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5551	E²prom Error Parameter 32	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5552	E²prom Error Parameter 33	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5553	E²prom Error Parameter 34	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5554	E²prom Error Parameter 35	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5555	E²prom Error Parameter 36	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1
	0x5556	E²prom Error Parameter 37	Error Writing E²prom Parameters (contact Manufacturer)	F	6,1

	0x5557	E²prom Error Parameter 38	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5558	E²prom Error Parameter 39	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5559	E²prom Error Parameter 40	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x555A	E²prom Error Parameter 41	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x555B	E²prom Error Parameter 42	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x555C	E²prom Error Parameter 43	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x555D	E²prom Error Parameter 44	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x555E	E²prom Error Parameter 45	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x555F	E²prom Error Parameter 46	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5560	E²prom Error Parameter 47	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5561	E²prom Error Parameter 48	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5563	E²prom Error Parameter 49	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5564	E²prom Error Parameter 51	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5565	E²prom Error Parameter 52	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5566	E²prom Error Parameter 53	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5567	E²prom Error Parameter 54	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5568	E²prom Error Parameter 55	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x5569	E²prom Error Parameter 56	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x556A	E²prom Error Parameter 57	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x556B	E²prom Error Parameter 58	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x556C	E²prom Error Parameter 59	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x556D	E²prom Error Parameter 60	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
	0x556F	E²prom Error Parameter 61	Error Writing E²prom Parameters (contact Manufacturer)	**F**	**6,1**
HARDWARE MEMORY E²PROM - FACTORY PARAMETERS	**0x5A00**	**E²PROM Data Area Golden Image**	Generic Error E²prom Data Area Golden Image	**-**	**-**
	0x5A01	Warning Data Golden Image	Warning Data Golden Image is free	**W**	**-**
	0x5A02	Error Data Golden Image	Data Golden Image is not written	**F**	**8,1**
		ALARM SOFTWARE			
SOFTWARE DEVICE	**0x6000**	**Software Device**	Generic Error Software Device	**-**	**-**
	0x6001	Update Parameters	Warning Request update by canopen is not permission (ONLY RS232)	**W**	**-**
		ALARM PARAMETERS			
DATA SET	**0x6300**	**Data Set Parameters Table**	Data Set Programming Error	**-**	**-**
	0x6301	Data record no. 1	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6302	Data record no. 2	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**

	0x6303	Data record no. 3	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6304	Data record no. 4	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6305	Data record no. 5	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6306	Data record no. 6	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6307	Data record no. 7	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6308	Data record no. 8	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0X6309	Data record no. 9	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x630A	Data record no. 10	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x630B	Data record no. 11	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x630C	Data record no. 12	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x630D	Data record no. 13	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x630E	Data record no. 14	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x630F	Data record no. 15	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6401	Data record no. 16	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6402	Data record no. 17	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
	0x6403	Data record no. 17	Programming Error Data Set (contact Manufacturer)	**F**	**7,1**
PARAMETER ERROR	**0x6320**	**Parameter Error**	Generic Parameter Error	**-**	**-**
	0x6321	Incongruity Data Configuration 1	Configuration Error (contact Manufacturer)	**F**	**6,4**
		ALARM ADDITIONAL MODULE			
ENCODER SINCOS	**0X7350**	**Encoder SinCos**	Error Generic Encoder SinCos	**F**	**2,6**
	0x7351	Rx Error	Error Message Received	**F**	**2,6**
	0x7352	Tx Error	Error Message Transmitted	**F**	**2,6**
	0x7353	Comand Read Position Error	Error Read Position	**F**	**2,6**
	0x7354	Comand Status Error	Error Status Encoder SinCos	**F**	**2,6**
	0x7355	Comand Type Error	Error Type Encoder SinCos	**F**	**2,6**
	0x7356	Comand Init Timeout	Error Timeout during Initialization SinCos	**F**	**2,6**
CONVERTER SINCOS	**0X7360**	**Converter Sin/Cos**	Error Generic Converter Sin/Cos	**F**	**6,3**
	0x7361	E²prom Ext	First programming E²prom external, we must reset the driver	**F**	**6,3**
	0x7362	Nerr Signal Amp	Fault has been occurred: Amplitude Error	**F**	**6,3**
	0x7363	Nerr Signal Freq	Fault has been occurred: Frequency Error	**F**	**6,3**
	0x7364	Nerr Signal Other	Fault has been occurred: configuration or Under voltage or System Error	**F**	**6,3**
	0x7365	Error Gen	General Error	**F**	**6,3**
RESOLVER	**0x7370**	**Resolver**	General Error	**-**	**-**

	0x7373	Resolver not in phasing	Alignment Fault Initial of resolver during read		-
	0x7374	Resolver Initialization	Resolver Fault Initialization has been occurred		2,4
	0x7375	Resolver Hardware Fault LOS (Loss of Signal)	Manufacturer specific value describes the cause of the triggering of the fault detection output pins (value of fault register chip resolver): - 0x01 (Bit 0): Configuration parity error - 0x02 (Bit 1):Phase error exceeds phase lock range - 0x04 (Bit 2): Velocity exceeds maximum tracking rate - 0x08 (Bit 3): Tracking error exceeds LOT threshold - 0x10 (Bit 4): Sine/cosine inputs exceed DOS mismatch threshold - 0x20 (Bit 5): Sine/cosine inputs exceed DOS over-range threshold - 0x40 (Bit 6): Sine/cosine inputs below LOS threshold - 0x80 (Bit 7): Sine/cosine inputs clipped	F	2,10
	0x7376	Resolver Hardware Fault DOS (Degradation of Signal)	Manufacturer specific value describes the cause of the triggering of the fault detection output pins (value of fault register chip resolver): - 0x01 (Bit 0): Configuration parity error - 0x02 (Bit 1):Phase error exceeds phase lock range - 0x04 (Bit 2): Velocity exceeds maximum tracking rate - 0x08 (Bit 3): Tracking error exceeds LOT threshold - 0x10 (Bit 4): Sine/cosine inputs exceed DOS mismatch threshold - 0x20 (Bit 5): Sine/cosine inputs exceed DOS over-range threshold - 0x40 (Bit 6): Sine/cosine inputs below LOS threshold - 0x80 (Bit 7): Sine/cosine inputs clipped	F	2,10
	0x7377	Resolver Hardware Fault LOT (Loss of Tracking)	Manufacturer specific value describes the cause of the triggering of the fault detection output pins (value of fault register chip resolver): - 0x01 (Bit 0): Configuration parity error - 0x02 (Bit 1):Phase error exceeds phase lock range - 0x04 (Bit 2): Velocity exceeds maximum tracking rate - 0x08 (Bit 3): Tracking error exceeds LOT threshold - 0x10 (Bit 4): Sine/cosine inputs exceed DOS mismatch threshold - 0x20 (Bit 5): Sine/cosine inputs exceed DOS over-range threshold - 0x40 (Bit 6): Sine/cosine inputs below LOS threshold - 0x80 (Bit 7): Sine/cosine inputs clipped	F	2,10
	0x7378	Resolver Hardware Fault LOS, DOS, LOT during phasing initialisation	Manufacturer specific value describes the cause of the triggering of the fault detection output pins (value of fault register chip resolver): - 0x01 (Bit 0): Configuration parity error - 0x02 (Bit 1):Phase error exceeds phase lock range - 0x04 (Bit 2): Velocity exceeds maximum tracking rate - 0x08 (Bit 3): Tracking error exceeds LOT threshold - 0x10 (Bit 4): Sine/cosine inputs exceed DOS mismatch threshold - 0x20 (Bit 5): Sine/cosine inputs exceed DOS over-range threshold - 0x40 (Bit 6): Sine/cosine inputs below LOS threshold - 0x80 (Bit 7): Sine/cosine inputs clipped	F	2,10
INCREMENTAL ENCODER	**0x7390**	**Incremental Encoder**	Error Generic Incremental Encoder	F	2, 5
	0x7391	Encoder error init	Encoder has initialization error due to sequence Hall or value null	F	2, 1
	0x7392	Encoder error congruence	Encoder has congruence error between Hall	F	2, 2
	0x7393	Encoder error phasing	Encoder has phasing error	F	2, 3
	0x7394	Encoder error Distance	Encoder Error Distance Hall	F	2, 4
COMMUNICATION	**0X7500**	**Communication**			
	0x7530	CANopen Protocol	CANopen Error Generic	-	-
	0x7531	CANopen Protocol – Init Error	Initialization Error	W	-
	0x7532	CANopen Protocol – Hardware Error	hardware Error	F	5,4
ALARM MONITORING					

COMUNICATION CANOPEN	**0x8100**	**Comunication Canopen**	**communication error**	**F**	**6,2**
	0x8110	Can Overrun	CAN Controller RX buffer hardware overrun (Overflow)	**F**	**6,2**
	0x8111	Tx Buffer Overflow	TX software buffer overflow	**F**	**6,2**
	0x8112	Rx Buffer Overflow	RX software buffer overflow	**F**	**6,2**
	0x8120	Can Passive	CAN in error passive	**F**	**6,2**
	0x8130	Heartbeat/Node Guarding	Heartbeat or Life Node Guarding	**F**	**6,2**
	0x8131	Error Node Guarding slave misses msg	Error Node Guarding: slave misses guarding message	**F**	**6,2**
	0x8132	Error Node Guarding lost connection	Error Node Guarding: lost connection life time elapsed for node	**F**	**6,2**
	0x8133	Error Node Guarding lost at least one msg	Error Node Guarding: slave misses guarding at least one msg	**W**	**-**
	0x8140	Bus Off Recovered	CAN recovered from bus-off	**W**	**-**
	0x8150	Can Id Collision	CAN-ID collision	**W**	**-**
	0x8160	State CAN Init	Drive communicates State Message :CANopen is in INIT state	**W**	**-**
	0x8170	State CAN Active	Drive communicates State Message :CANopen is in ACTIVE state	**W**	**-**
	0x8180	State CAN Busoff	Drive communicates State Message :CANopen is in BUSOFF state	**W**	**-**
	0x8190	State CAN Error Passive	Drive communicates State Message :CANopen is in PASSIVE state	**W**	**-**
		ALARM PROTOCOL			
TORQUE PROFILE CONTROL	**0x8300**	**Torque control**	**General Error for Profile Torque Controller**	**F**	**6,6**
	0x8341	Torque Type	Error type selected is not managed	**F**	**6,6**
	0x8351	Torque Dynamic Brake	Error Dynamic Brake is not implemented	**F**	**6,6**
VELOCITY SPEED CONTROLLER	**0x8400**	**Velocity speed controller**	**General Error for Profile Velocity Controller**	**F**	**6,7**
	0x8411	Following error	The difference between the velocity command and the actual velocity is greater than the value that is set in maximum velocity error	**F**	**6,7**
	0x8412	Over Speed	Actual speed exceeds the velocity over speed value	**F**	**6,7**
POSITION CONTROLLER	**0x8500**	**Position controller**	**General Error for Profile Positioner Controller**	**F**	**-**
POSITIONING CONTROLLER	**0x8600**	**Positioning controller**	**General Error for Profile Positioning Controller**	**F**	**-**
	0x8611	Following error	The difference between the position command and the actual position is greater than the value that is set in maximum position error (object 6065h)	**F**	**-**
CANOPEN EEPROM	**0x8B00**	**Store and Restore Process**	**General Error for Store and Restore Process**	**F**	**8,2**
	0x8B01	Warning Store/Restore/ Load Parameters	Warning: command store/restore/load are disabled because the drive isn't in "ready to switchon" or "disabled"	**W**	-
	0x8B02	Store Parameters Error	Error management object Store Parameters 1010h	**F**	**8,2**
	0x8B03	Memory Store Eeprom Full	Error Memory Store full for CAN object parameters	**F**	**8,2**
	0x8B04	Restore Par Eeprom	Error management object ReStore Parameters 1011h	**F**	**8,2**
	0x8B05	Memory Restore Eeprom Full	Error Memory Restore full for CAN object parameters	**F**	**8,2**
	0x8B10	**Init Object CANopen from Eeprom**	**Error Initialization Canopen Object from Eeprom.**	**F**	**8,2**
	0x8B11	Init Object CANopen 0x6081	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**

	0x8B12	Init Object CANopen 0x6082	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B13	Init Object CANopen 0x6083	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B14	Init Object CANopen 0x6084	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B15	Init Object CANopen 0x60C5	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B16	Init Object CANopen 0x60C6	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B17	Init Object CANopen 0x607F	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B18	Init Object CANopen 0x6088	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B19	Init Object CANopen 0x6096	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B1A	Init Object CANopen 0x6097	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B1B	Init Object CANopen 0x606D	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B1C	Init Object CANopen 0x606E	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B1D	Init Object CANopen 0x606F	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B1E	Init Object CANopen 0x6070	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0x8B1F	Init Object CANopen 0x6075	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0X8B20	Init Object CANopen 0x6076	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0X8B21	Init Object CANopen 0x6072	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0X8B22	Init Object CANopen 0x6073	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0X8B23	Init Object CANopen 0x60E0	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0X8B24	Init Object CANopen 0x60E1	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0X8B25	Init Object CANopen 0x6087	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
	0X8B26	Init Object CANopen 0x6086	Error Initialization Canopen Object from Eeprom.	**F**	**8,2**
DSP402 FSM	**0x8C00**	**Profile 402 Finite State Machine**	**General Error Profile 402 Finite State Machine**	**F**	**6,5**
	0x8C01	Mode Of Operation Error	Mode Of Operation (6060h) has been written when the drive is in "operation enabled" state	**F**	**6,5**
	0x8C02	Type Profile	Type Profile not defined	**F**	**6,5**
	0x8C03	Profile Error	Profile Selected not managed	**F**	**6,5**
	0x8C04	None Profile	Run State and No Profile selected	**F**	**6,5**
	0x8D00	Analog Profile	Analog Profile Condition Error	**F**	**6,5**

Table 21 - Emergency Description

SYNC Protocol

Several devices can be synchronised with each other. To that purpose one of the devices (in general the master controller) periodically sends a SYNC message.

The SYNC object is a network wide system clock and it is the trigger for synchronous message. The SYNC has a very high priority and contains no data in order to guarantee a minimum of jitter.

The SYNC object is used to trigger synchronous PDOs; all connected devices receive these messages and use them for the treatment of the PDO.

The identifier of the SYNC objects is set in the object dictionary under the index 0x1005 (by default 0x80).

COB-ID	Rx/Tx	DLC	Byte							
			0	1	2	3	4	5	6	7
0x80	Tx	8	No user data							

Table 22 - SYNC message Structure

Triggering synchronous PDOs:

- Synchronous RPDO: The command transmitted with the PDO is not executed until a SYNC object is received.
- Synchronous TPDO: The PDO with the current data is not sent until a SYNC object is received.

Error Control Protocols

There are 2 error control protocols enable the monitoring of a CANopen network.

1. Heart-Beat
2. Node/Life-Guarding

The Heartbeat protocol is used to verify that all network participants are still available in a CANopen network and that they are still in their intended NMT state. In old-fashioned CANopen systems, the CAN remote frame-based Node/Life-guarding protocol is used for this purpose, instead of the Heartbeat protocol.

All error control protocols are based on the same CAN message with the CAN-ID 0x700 + Node-ID of the CANopen device that are to be monitored.

Caution

It is important and mandatory to have at least one of monitor control for CAN communication safety.

Caution

CAN in Automation (CiA) recommends using heartbeat in the new implementations due to the sensible nature of Guarding using the RTR frames. CiA no longer recommends using CAN remote frame-based services.

Node Guarding Protocol

This service is based that the Master Controller sends an RTR message with the identifier 700h+IdNode to the respective slave. The slave must send a message as response: this message is structured as follows.

Bit 7 alternates here on each transfer, this bit determines if a message was lost.

Bit 6 to 0 define the current NMT status of the slave.

COB-ID	Rx/Tx	DLC	Byte 0	1	2	3	4	5	6	7
0x700 + IdNode	**Tx**	**1**	7Bit toggle +NMT State	-						

Table 23 - Node Guarding Message Structure

To configure the node guarding use three time intervals

- Guard time: the time between two RTR messages. This can be different for each CAN node and is stored in the slave in object $100C_h$ "Guard Time" (unit ms).
- Live time factor: a multiplier for the guard time, this is stored in the CAN slave in object $100D_h$ "Life Time Factor" and it can be different for each slave on the CAN bus.
- Possible live time: the time produced by multiplying guard time and live time factor

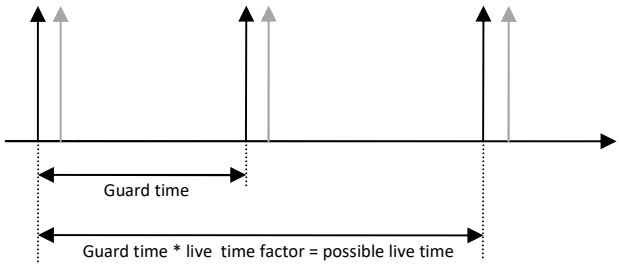


Figure 8 - Node Guarding time message

The following conditions are checked during node guarding:

- The NMT Master Controller must send the RTR requests within "possible live time"
- The slave must send the response to the RTR request within the "possible live time"
- The slave must respond with its NMT state. In addition the "toggle bit" must be set correctly

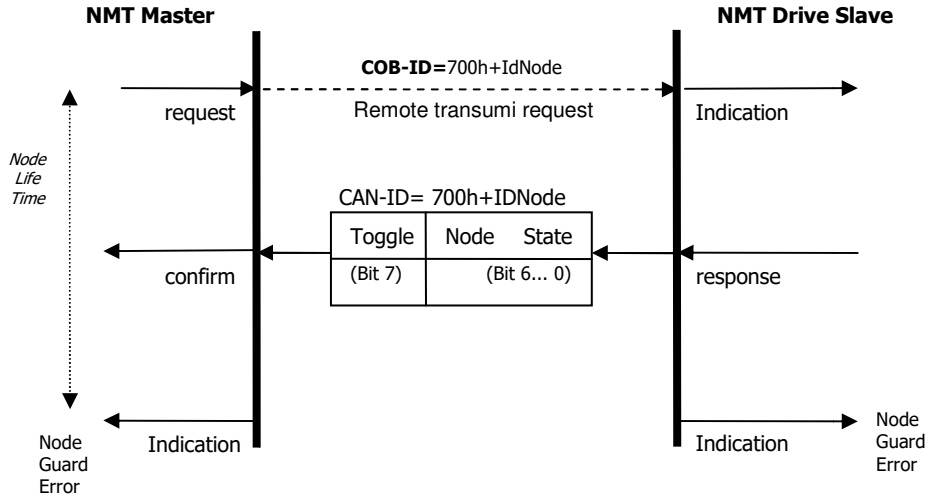


Figure 9 - Node Guarding timeframe message

Heartbeat Protocol

Heartbeat is the message to monitor the communication between drive and Master Controller.

The drive cyclically sends a message to the master controller. The master controller can check if it cyclically receives the heartbeat and initiate appropriate reactions if not.

The heartbeat message will be sent with the identifier 700h +Id-Node. It is only composed of 1 Byte, containing the NMT state of the servo.

COB-ID	Rx/Tx	DLC	Byte							
			0	1	2	3	4	5	6	7
0x700 + Id Node	Tx	1	NMT State	-						

Table 24 - HeartBeat Message Structure

This object indicates what action shall be performed when one of the following events occurs:

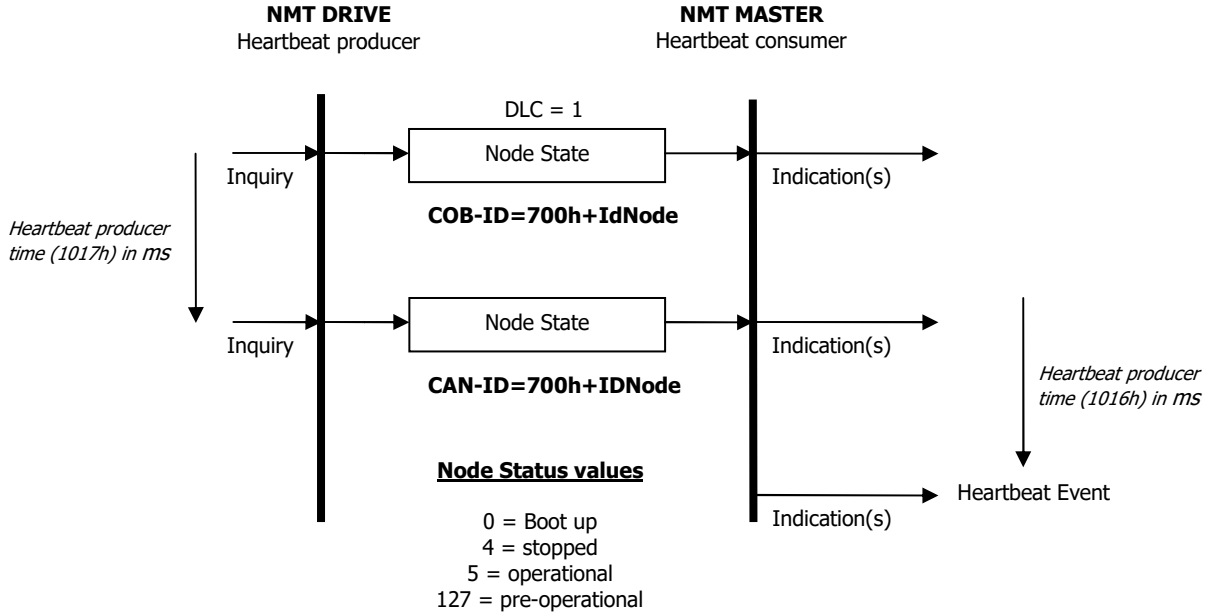


Figure 10 - Heartbeat timeframe

This service is enabled when the value of Producer heartbeat time (1017_h) object is not zero.

The relationship between producer and consumer can be configured with objects. If a consumer does not receive a signal within the period of time set with Consumer heartbeat time (1016_h) it generates an error message (heartbeat event).

If the consumer heartbeat time (1016_h) object equal 0 then the monitoring by a consumer.

information

Referring to "APPENDIX" chapter to read "Heartbeat Mechannism"

CAN Error Communication

It sends an emergency message to describe the cause of the communication error.

The drive goes in FAULT state when it detects the BUS-OFF.

It is possible to have a High sensibility "CAN Error Communication" handling, in this case the drive goes in alarm after timeout when it detects the error PASSIVE. To have this configuration contact the manufacturer.

When the drive is configured by analog mode the drive draws attention with warning message.

Network Management (NMT)

The Network Management (NMT) is one of the service elements of the application layer.

The NMT serves to configure, initialise, and handle errors in a CAN network. NMT commands are used to control the communication state of the servo drive and to broadcast manufacturer messages to all other connected servo drives.

An NMT Slave is uniquely identified in the network by its Node-ID, a value in the range of [1 to 127].

CANopen devices enter the NMT state Pre-operational directly after finishing the CANopen devices initialization. During this NMT state CANopen device parameterization and CAN-ID-allocation via SDO (e.g. using a configuration tool) is possible. Then the CANopen devices may be switched directly into the NMT state Operational.

The Network Management is node oriented and follows a master-slave structure. NMT objects are used for executing NMT services. Through NMT services, nodes are initialised, started, monitored, reset or stopped. All nodes are regarded as NMT slaves.

NMT requires that one device in the network fulfils the function of the NMT Master.

NMT Services:

- **Module Control Services:** Through Module Control Services, the NMT master controls the state of the NMT slaves. The state attribute is one of the values (STOPPED, PRE-OPERATIONAL, OPERATIONAL and INITIALISING). The Module Control Services can be performed with a certain node or with all nodes simultaneously.
- **Error Control Service:** Through Error control services the NMT detects failures in a CAN-based Network. Local errors in a node may e.g. lead to a reset or change of state. Error Control services are achieved principally through periodically transmitting of messages by a device. There exist two possibilities to perform Error Control i.e. Node Guard and Heart Beat Error Control.
- **Boot-up Service:** Through this service, the NMT slave indicates that a local state transition occurred from the state INITIALISING to the state PRE-OPERATIONAL.

NMT state machine

CANopen devices enter the NMT state Pre-operational directly after finishing the CANopen devices initialization. During this NMT state CANopen device parameterization and CAN-ID-allocation via SDO possible. Then the CANopen devices may be switched directly into the NMT state Operational.

The NMT state machine determines the behaviour of the communication function unit.

The coupling of the application state machine to the NMT state machine is CANopen device dependent and falls into the scope of device profiles and application profiles.

The following picture shows the NMT state diagram of a CANopen device is specified.

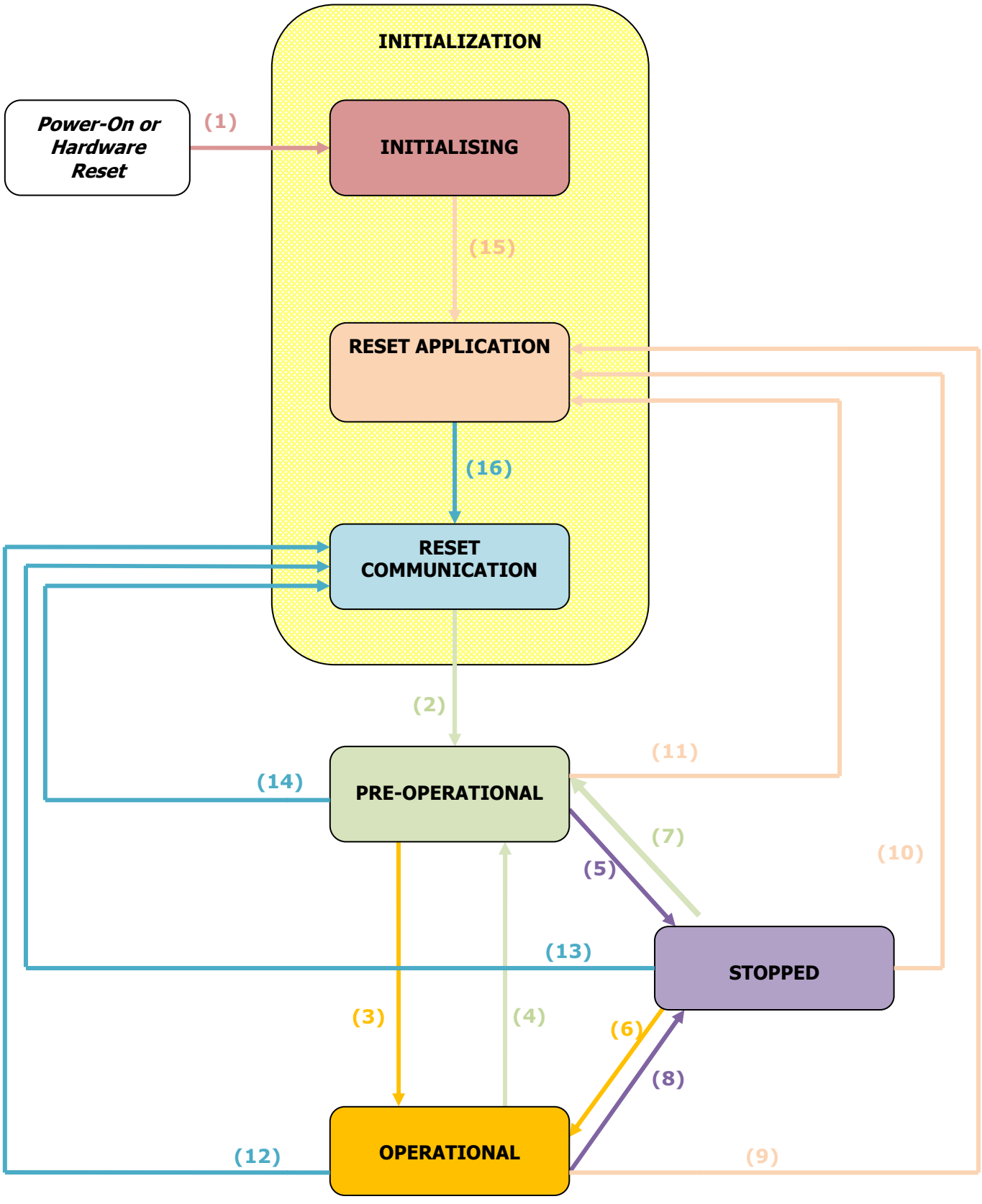


Figure 11 - NMT state machine

The following Table describes the transitions.

Transition	Description
(1), (15), (16)	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

Table 25 - NMT Network Management

The following network communication states are supported, with the following communication type.

State	Description	SDO	PDO	NMT	SYNC
Initialization	Drive is not ready, or it is booting. Drive will not respond to communication and will not transmit anything.	-	-	-	-
Pre-operational	Drive boot sequence is complete, but no command has been received to enter operational mode. The servo drive will respond to SDO and NMT messages, but not to PDOs.	x	-	x	x
Operational	Drive is fully operational, responding to PDO, SDO and NMT messages.	x	x	x	x
Stopped	Servo drive can respond only to NMT objects (including heartbeats).	-	-	x	-

Table 26 -NMT Network Management

Network Initialization:

When powering the drive enter in the state machine Network Management (NMT). The first state after an internal reset or a power cycle is the NMT initialization state.
In this state the drive loads all parameters from the non-volatile memory into the RAM. After finishing the NMT initialisation state the drive enters the pre-operational State. During this state transition the CANOpen drive sends its boot-up message.

The NMT state INITIALIZATION shall be divided into three NMT sub-states in order to enable a complete or partial reset of a CANopen device.

- **Initialising**: This is the first NMT sub-state the CANopen device enters after power-on or hardware reset. After finishing the basic CANopen device initialisation the CANopen device enters autonomously into the NMT sub-state reset application.
- **Reset application**: In this NMT sub-state the parameters of the manufacturer-specific profile area and of the standardized device profile area are set to their power-on values. After setting of the power-on values the NMT sub-state reset communication is entered autonomously.
- **Reset communication**: In this NMT sub-state the parameters of the communication profile area are set to their power-on values. After this the NMT state Initialisation is finished and the CANopen device executes the NMT service boot-up write and enters the NMT state Pre-operational.

Power-on values are the last stored parameters. If storing has not been executed or if the reset was preceded by the command restore defaults, the power-on values are the default values according to the communication and device profile specifications.

Network Pre-Operational state:

In the pre-operational state communication via SDOs is possible, while (PDO) communication is not allowed. Configuration of PDOs and device parameters may be performed. Also the emergency objects and error control service like the CANopen sensors "heartbeat message" occur in this state. The node will be switched into the operational state directly by sending a NMT "start remote node".

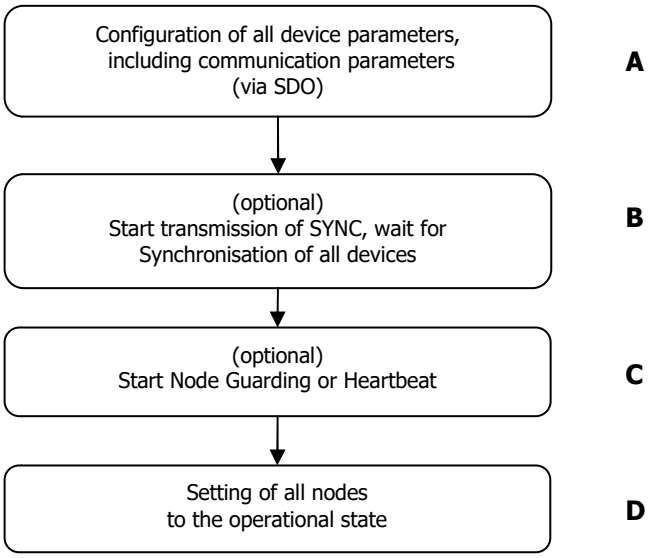
Network Operational State:

In the operational state all communication objects – including PDO handling – are active. Object dictionary access via SDO is possible.

Network Stopped State:

By switching a device into the stopped state it is forced to stop the communication, except node guarding and heartbeat, if active.

Network Initialisation Process:



STEP A): the device is in the node state PRE-OPERATIONAL which is entered automatically after power-on. In this state the devices are accessible via their default-SDO, the configuration of SDOs settings and optionally the setting of COB-IDs may be performed via SDO objects. In many cases a configuration is not even necessary as default values are defined for all application and communication parameters.

STEP B): If the application requires the synchronisation of all or some nodes in the network, the appropriate mechanisms can be initiated in the optional Step B. It can be used to ensure that all nodes are synchronised by the SYNC object before entering the node state OPERATIONAL in step D. The first transmission of SYNC object starts within 1 sync cycle after entering the PRE-OPERATIONAL state.

STEP C): In this step the Node guarding or Heartbeat can be activated using the guarding parameters configured in step A.

STEP D): Now the master controller has to move the drive in OPERATIONAL state. With step D all nodes are enabled to communicate via their PDO objects.

NMT Message

The NMT message contains only 2 data byte, with the following format:

COB-ID	Rx/Tx	DLC	Byte							
			0	1	2	3	4	5	6	7
0x00	Rx	2	Command	Id Node	-					

Table 27 - NMT Message Structure

With the following commands the NMT state can be changed.

Byte 0 value Definition:

Command	Meaning	Description	Transition	Target-State
0x01	Start Remote Node	Through this service the NMT master sets the state of the selected NMT slave(s) to "operational".	(3) (6)	OPERATIONAL
0x02	Stop Remote Node	Through this service the NMT master sets the state of the selected NMT slave(s) to "stopped".	(5) (8)	STOPPED
0x80	Enter Pre-Operational State	Through this service the NMT master sets the state of the selected NMT slave(s) to "pre-operational".	(4) (7)	PRE-OPERATIONAL
0x81	Reset Application	Through this service the NMT master sets the state of the selected NMT slave(s) from any state to the "reset application" sub-state.	(9) (10) (11)	RESET APPLICATION
0x82	Reset Communication	Through this service the NMT master sets the state of the selected NMT slave(s) from any state to the "reset communication" sub-state. After completion of the service, the state of the selected remote nodes will reset communication.	(12) (13) (14)	RESET COMMUNICATION

Byte 1 value Definition:

Id Node	Description
ID Node	set 0x00 for all devices (global mode) set Id-Node (0x01…0x7F) for a specific device

Table 28 - NMT Description Field

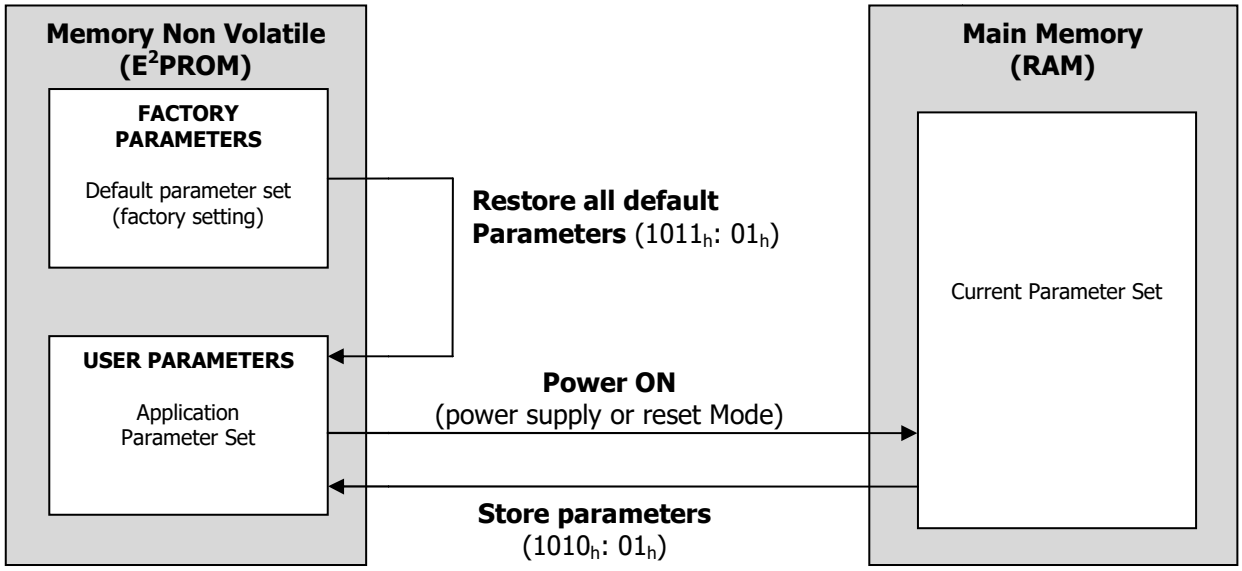
Bootup Message

After power-on or after reset, the LSD controller reports through a Bootup message that the initialising has been finished. Next this message the LSD is in the NMT state preoperational.

COB-ID	Rx/Tx	DLC	Byte							
			0	1	2	3	4	5	6	7
0x700 + Id Node	Tx	1	0x00	-						

Table 29 - BOOTUP Message Structure

Store and Restore



The CiA CANopen protocol specification defines two objects to store and restore parameters:

- Object 1010_h – Store Parameters
- Object 1011_h – Restore Parameters

In order to save all parameters the master writes in the SDO 1010h index the value "save" to one of the subentries of the object. This procedure causes the corresponding set of parameters to be written to non-volatile memory. After the NMT reset node or the NMT reset communication the parameters will be loaded in object dictionary automatically.

The following Objects can be changed and stored in E^2prom by writing in object 1010_h: 2_h (Communication Parameters).

- 1000_h: Device Type
- 1001_h: Error Register
- 1002_h: Manufacturer Status Register
- 1003_h: Predefined Error Field (History List)
- 1005_h: COB-ID Sync
- $100C_h$: Guard Time
- $100D_h$: Life Time Factor
- 1014_h: COB-ID EMCY
- 1017_h: Producer Heartbeat Time
- 1018_h: Identity Object
- 1029_h: Error Behaviour
- 1400_h: RxPDO1 Parameter
- 1401_h: RxPDO2 Parameter
- 1402_h: RxPDO3 Parameter

- 1403_h: RxPDO4 Parameter
- 1600_h: RxPDO1 Mapping
- 1601_h: RxPDO2 Mapping
- 1602_h: RxPDO3 Mapping
- 1603_h: RxPDO4 Mapping
- 1800_h: TxPDO1 Parameter
- 1801_h: TxPDO2 Parameter
- 1802_h: TxPDO3 Parameter
- 1803_h: TxPDO4 Parameter
- $1A00_h$: TxPDO1 Mapping
- $1A01_h$: TxPDO2 Mapping
- $1A02_h$: TxPDO3 Mapping
- $1A03_h$: TxPDO4 Mapping

The following Objects can be changed and stored in E^2prom by writing in object 1010_h: 3_h (Application Parameters).

- 6073_h: Max Current
- $607E_h$: Polarity(*)
- $607F_h$: Max Profile Velocity
- 6080_h: Max Motor Speed
- 6083_h: Profile Acceleration (*)
- 6084_h:Profile Deceleration (*)
- 6096_h: Velocity Factor
- 6097_h: Acceleration Factor
- $60C5_h$: Max Acceleration
- $60C6_h$: Max Deceleration
- 6086_h: Motion Profile Type
- 6072_h: Max torque ***(not implemented yet)***
- 6073_h: Max Current
- $60E0_h$: Positive torque limit value
- $60E1_h$: Negative torque limit value
- 6088_h: Torque Profile Type
- 6075_h: Motor Rated Current
- 6076_h: Motor Rated torque ***(not implemented yet)***
- 6087_h: Torque Slope (*)

The following Objects can be changed and stored in E^2prom by writing in object 1010_h: 4_h (Manufacturer Parameters).

- 2000_h: Id Node
- 2001_h: Baudrate
- 3002_h: Brake Parameters (*)

- 3007_h: Dynamic Brake Parameters (*)
- 3200_h: Pid Current (*)
- 3201_h: Pid Velocity (*)
- 3202_h: Pid Positioner (*)

The objects marked with (*) can be changed in run time. If the drive is disconnected the value modified are lost.
All parameters can be stored in E^2prom, the changes are not accepted until either the voltage supply is briefly disconnected or the CANopen message RESET COMM (NMT) is sent to the motor.

In order to avoid the restoring of default parameters by mistake, it is possible loading the factory parameters. The master sends the SDO 1011h and writes the signature "load" to one of sub-index.

Function mode restore factory parameters:

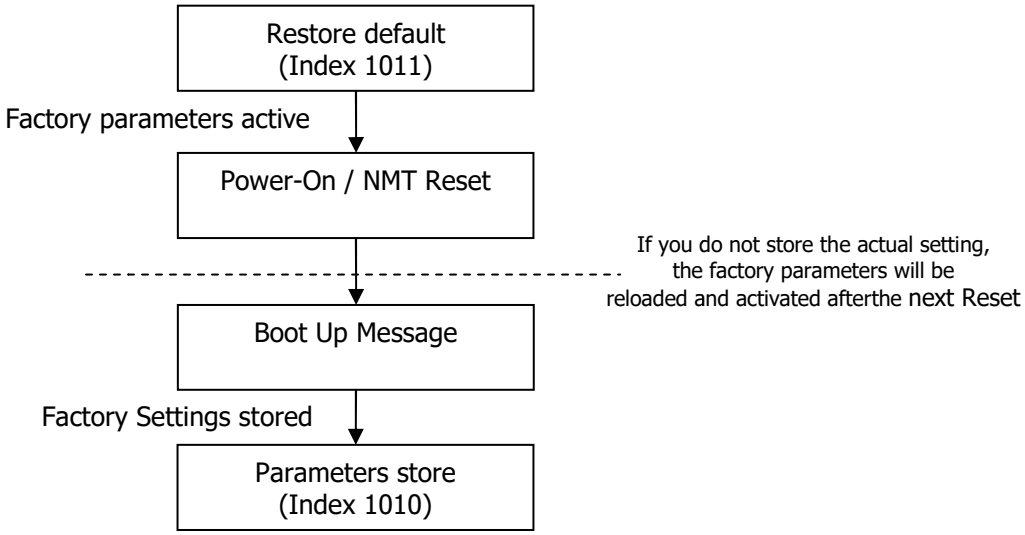


Figure 12 - Restore Flow Chart

Communication Parameters are these "DEFAULT COMMUNICATION":

NAME	Index	Sub Index	Value Field	Default Parameters
P301 DEV TYPE	0x1000	0	Device Type	0xFF7A0192
P301 ERR REG	0x1001	0	Error Register	0
P301 MANUF STATUS REG	0x1002	0	Manufacturer Status Register	0
P301 PREDEF ERR FIELD	0x1003	0	Number of Errors	15
		1	history[1]	0
		2	history[2]	0
		3	history[3]	0
		4	history[4]	0
		5	history[5]	0
		6	history[6]	0
		7	history[7]	0
		8	history[8]	0
		9	history[9]	0
		10	history[10]	0
		11	history[11]	0
		12	history[12]	0
		13	history[13]	0
		14	history[14]	0
		15	history[15]	0
P301 COBID SYNC	0x1005	0	COB-ID SYNC	COB-ID = 80000080h +Id
P301 GUARD TIME	0x100C	0	Guard Time	0 = Disabled

P301 LIFETIME FACTOR	0x100D	0	Life Time Factor	0 = Disabled
P301 COBID EMERGENCY	0x1014	0	COB-ID EMCY	COB-ID = 80h+ID
P301 PRODUCER HB TIME	0x1017	0	Producer Heartbeat Time	0 = Disabled
P301 IDENTITY OBJECT	0x1018	0	number entries	4
		1	Vendor Id	0x01FB
		2	Product Code	0
		3	Revision number	0
		4	Serial number	0
P301 ERR BEHAVIOR	0x1029	0	Number of Entries	1
		0	Communication Error	0
P301 RXPDO 1 PARAM	0x1400	0	Number of Entries	3
		1	COB-ID	COB-ID = 200h+ID, Receive PDO enabled
		2	Transmission Type	0xFE =Asynchronous
		3	Inhibit Time	0x5 = 100us
P301 RXPDO 2 PARAM	0x1401	0	Number of Entries	3
		1	COB-ID	COB-ID = 300h+ID, Receive PDO enabled
		2	Transmission Type	0xFE =Asynchronous
		3	Inhibit Time	0x5 = 100us
P301 RXPDO 3 PARAM	0x1402	0	Number of Entries	3
		1	COB-ID	COB-ID = 400h+ID, Receive PDO enabled
		2	Transmission Type	0xFE =Asynchronous
		3	Inhibit Time	0x5 = 100us
P301 RXPDO 4 PARAM	0x1403	0	Number of Entries	3
		1	COB-ID	COB-ID = 500h+ID, Receive PDO enabled
		2	Transmission Type	0xFE =Asynchronous
		3	Inhibit Time	0x5 = 100us
P301 RXPDO 1 MAPPING	0x1600	0	Number of Entries	3
		1	Mapping Entry 1	0x60400010 = Controlword
		2	Mapping Entry 2	0x60600008 = Mode of operation
		3	Mapping Entry 3	0x60FE0120 = Digital output
P301 RXPDO 2 MAPPING	0x1601	0	Number of Entries	2
		1	Mapping Entry 1	0x60400010 = Controlword
		2	Mapping Entry 2	0x607A0020 = Target Position
P301 RXPDO 3 MAPPING	0x1602	0	Number of Entries	2
		1	Mapping Entry 1	0x60400010 = Controlword
		2	Mapping Entry 2	0x60FF0020 = Target Velocity
P301 RXPDO 4 MAPPING	0x1603	0	Number of Entries	2
		1	Mapping Entry 1	0x60400010 = Controlword
		2	Mapping Entry 2	0x60710010 = Target Torque
P301 TXPDO 1 PARAM	0x1800	0	Number of Entries	3
		1	COB-ID	COB-ID = 180h+ID, Receive PDO enabled
		2	Transmission Type	0xFD = Asynchronous – RTR only
		3	Inhibit Time	0x5 = 100us
P301 TXPDO 2 PARAM	0x1801	0	Number of Entries	3
		1	COB-ID	COB-ID = 280h+ID, Receive PDO enabled
		2	Transmission Type	0xFD = Asynchronous – RTR only
		3	Inhibit Time	0x5 = 100us
P301 TXPDO 3 PARAM	0x1802	0	Number of Entries	3
		1	COB-ID	COB-ID = 380h+ID, Receive PDO enabled
		2	Transmission Type	0xFD = Asynchronous – RTR only
		3	Inhibit Time	0x5 = 100us
P301 TXPDO 4 PARAM	0x1803	0	Number of Entries	3
		1	COB-ID	COB-ID = 480h+ID, Receive PDO enabled
		2	Transmission Type	0xFD = Asynchronous – RTR only
		3	Inhibit Time	0x5 = 100us
P301 TXPDO 1 MAPPING	0x1A00	0	Number of Entries	3
		1	Mapping Entry 1	0x60410010 = Statusword
		2	Mapping Entry 2	0x60610008 = Mode of operation display
		3	Mapping Entry 3	0x60FD0020 = Digital input
P301 TXPDO 2 MAPPING	0x1A01	0	Number of Entries	3
		1	Mapping Entry 1	0x60410010 = Statusword
		2	Mapping Entry 2	0x60640020 =Position Actual Value
P301 TXPDO 3 MAPPING	0x1A02	0	Number of Entries	3
		1	Mapping Entry 1	0x60410010 = Statusword
		2	Mapping Entry 2	0x606C0020 = Velocity Actual Value
P301 TXPDO 4 MAPPING	0x1A03	0	Number of Entries	3
		1	Mapping Entry 1	0x60410010 = Statusword
		2	Mapping Entry 2	0x60770010 = Torque Actual Value

Table 30 - Communication Parameters

TABLE OF IDENTIFIERS

The following table gives a survey of the used identifiers.

Object-Type	Identifier (hexdecimal)
SDO (MASTER to LSD)	0x600 + IdNode
SDO (LSD to MASTER)	0x580 + IdNode
TPDO1	0x180 + IdNode
TPDO2	0x280 + IdNode
TPDO3	0x380 + IdNode
TPDO4	0x480 + IdNode
RPDO1	0x200 + IdNode
RPDO2	0x300 + IdNode
RPDO3	0x400 + IdNode
RPDO4	0x500 + IdNode
SYNC	0x80
EMCY	0x80 + IdNode
HEARTBEAT	0x700 + IdNode
BOOTUP	0x700 + IdNode
NMT	0x00

Table 31 - Table Of Identifiers

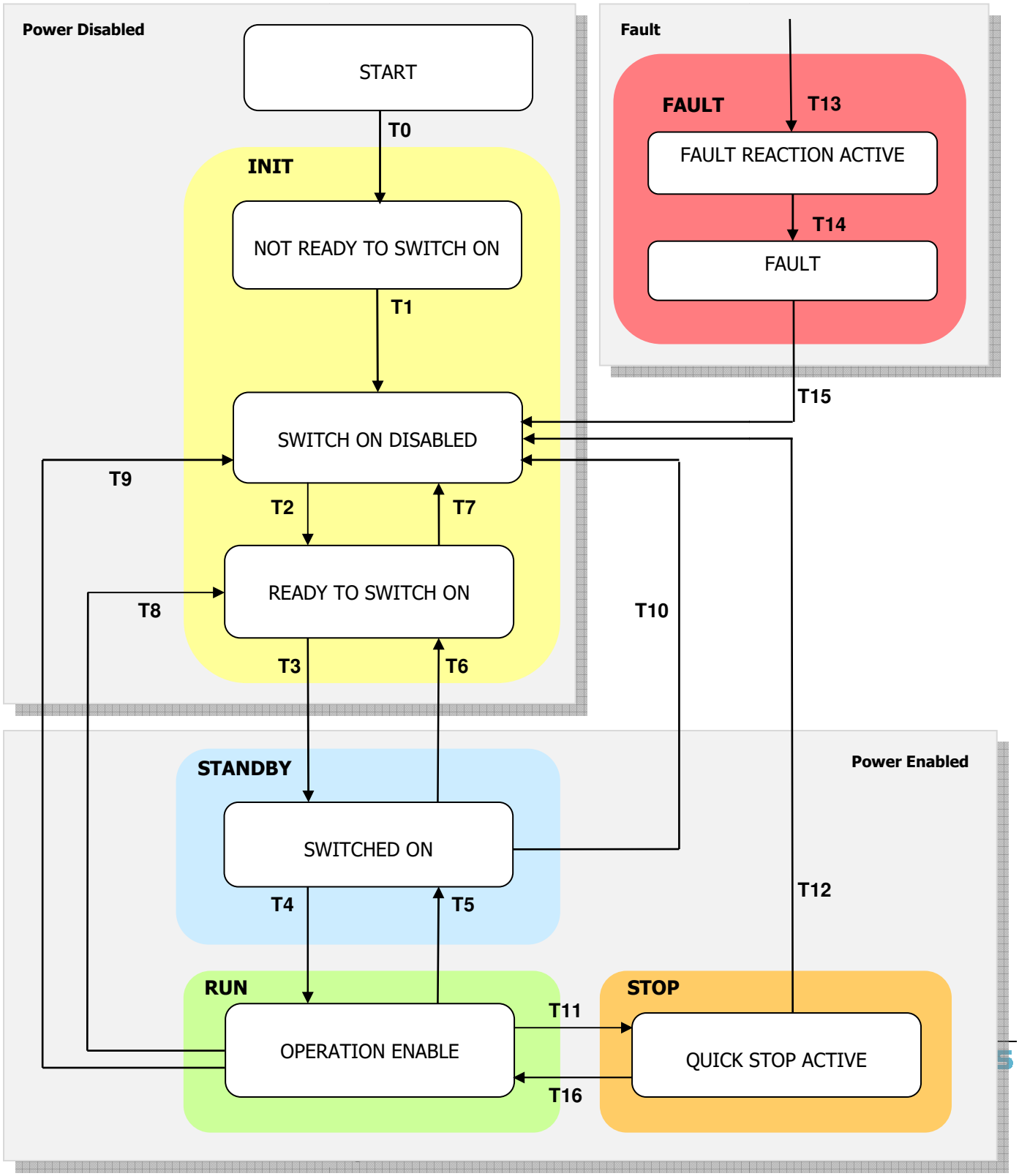
PROFILE DSP402

information

For Additional Information please refer to CiA DS402 standard.

State Machine Profile DSP402

The drive is checked and controlled by a state machine according t DSP402.



State changes are triggered by internal events such as the occurrence of an error or external demand by means of Controlword (6040_h). The object Statusword (6041_h) gives feedback about the actual state.

After power-up and initialisation, the drive switches to the state "Switch On Disabled" automatically. In this state the device waits a controlword command. In the state "Operation Enabled" the drive is fully operational.

SAFETY state is not implemented in DSP402. The state "SAFETY" has been added to protect and to define the drive when the emergency is applied. (See chapter "SAFETY")

Actual State may be read by statusword, with standard coding (defined by CiA DSP402):

STATUS	Description
NOT READY TO SWITCH ON	**INIT STATE:** The drive is being initialized and is running the self test. The drive function is disabled. This state is an internal state in which communication is enabled only at the end. The user can neither retrieve nor monitor this state.
SWITCH ON DISABLED	**INIT STATE:** No power applied. Drive initialization is completed. The drive parameters have been set up. Drive parameters may be changed. The drive function is disabled. The parameters can be saved in E^2prom. *SWITCH ON DISABLED is the minimum state to which a user may switch.*
READY TO SWITCH ON	**INIT STATE:** No power applied. The drive parameters have been set up. Drive parameters may be changed. The drive function is disabled. The parameters can be saved in E^2prom.
SWITCHED ON	**STANDBY STATE:** No power applied. The power stage is ready to RUN condition (to "operation enable"). The drive parameters may be changed. The drive function is disabled. The parameters can be saved in E^2prom.
OPERATION ENABLE	**RUN STATE:** *(This corresponds to normal operation of the drive)* No faults have been detected. Power applied to the motor. The drive function is enabled. The drive parameters may be changed. If automatically brake is enabled than it is released, in according to the brake parameter timing. ***The drive parameters can't be saved and restored in E^2prom.***

QUICK STOP ACTIVE	**STOP STATE:** No faults have been detected. Power applied to the motor. The drive function is enabled. The drive parameters may be changed. The drive stops the motion and either stays in quick stop with torque applied. ***The drive parameters can't be saved and restored in E^2prom.***
FAULT REACTION ACTIVE	**FAULT STATE:** The drive parameters may be changed. A fault has occurred in the drive. The fault reaction function is being executed. The drive function is disabled. This parameter cannot be retrieved by the user. The parameters can be saved in E^2prom.
FAULT	**FAULT STATE:** The drive parameters may be changed. A fault has occurred in the drive. The drive function is disabled. The parameters can be saved in E^2prom.

Table 32 - Status Word

The follow table shows the Led Codes referring the status Drive of "Lafert Servo Drive" and the correspondent state DSP402 state machine.

MACRO DRIVE STATE	CANOpen STATE	STATUS 1 LED GREEN	STATUS 2 LED YELLOW	LED VIEW
INIT	Not Ready To Switch On	"BLINK" simultaneously	"BLINK" simultaneously	 1 simultaneously 2 simultaneously
	Switch On Disabled Ready to Switch On	"BLINK" alternately	"BLINK" alternately	 1 alternately 2 alternately
STANDBY	Switched On	"BLINK"	OFF	 1 BLINK 50% 2 OFF
FAULT	Fault Fault reaction fault	"BLINK" [x]	"BLINK" [y]	 1 see fault 2 chapter
RUN (RUNV / RUNC)	Operation Enabled	ON	OFF	 1 ON 2 OFF
STOP	Quick Stop Active	ON	ON	 1 ON 2 ON
SAFETY	-	OFF	"BLINK"	 1 OFF 2 BLINK
COMMUNICATION CAN ERROR	-	OFF	ON	 1 OFF 2 ON

Table 33 - Drive Status

MODE OF OPERATION

Different operation modes are available with the CiA 402 profile:

- **Profile position mode**: ***it is not available yet.***
- **Profile velocity mode**: Reference velocity assignment by a controller. The drive calculates the necessary motion profiles independently. The movement profile is defined by velocity and acceleration/decelerations commands.
- **Profile torque mode**: The profile torque mode allows control device to transmit the target torque value, which is processed via the trajectory generator. The torque slope and torque profile type parameters are required.
- **Homing Mode**: ***it is not available yet.***

CANOpen Run Sequence Velocity Mode

See picture below to the flow chart of running sequence

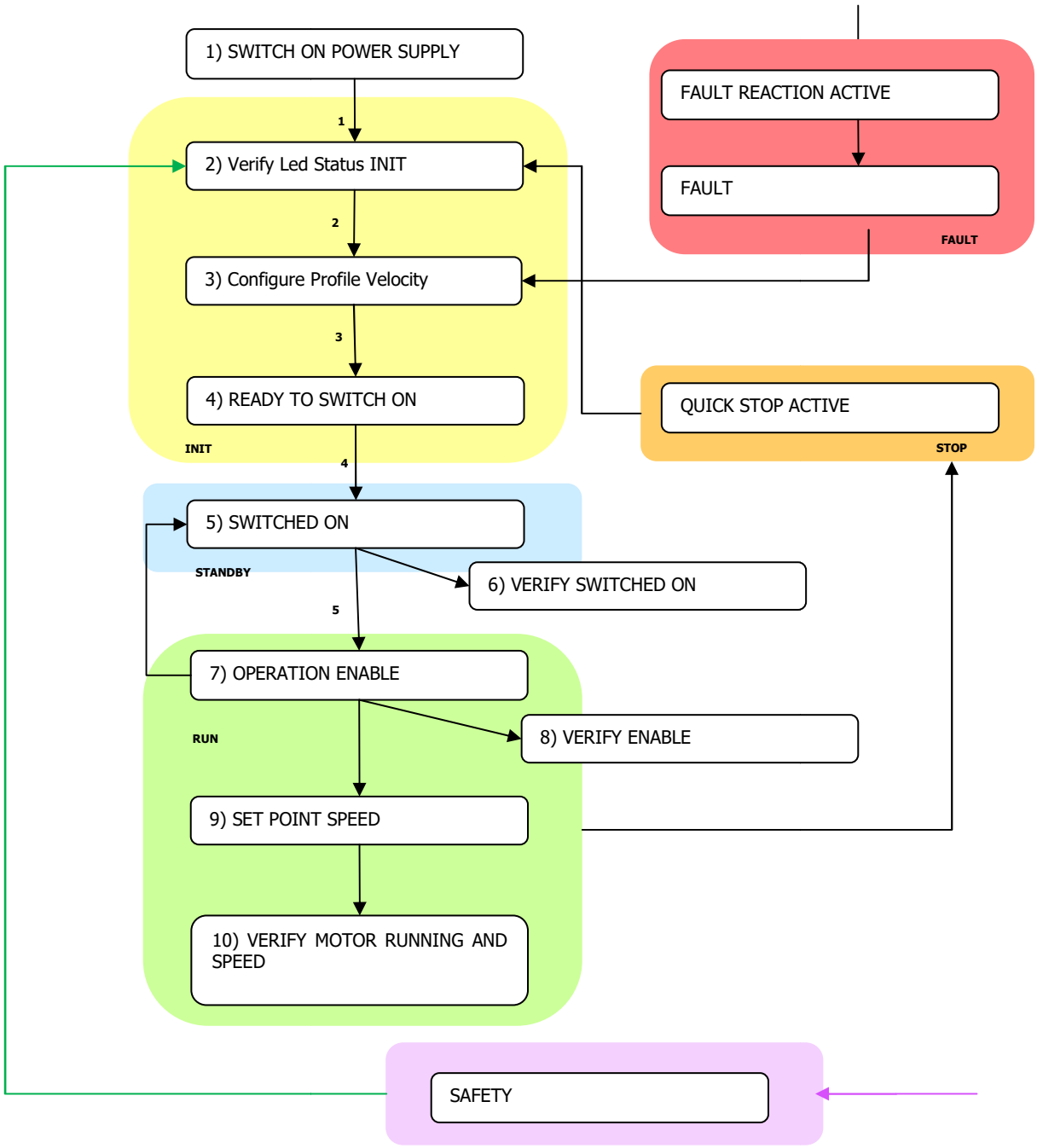


Figure 14 - CANOpen Run Sequence Velocity Mode

NOTE:

- The STO (SAFETY) command may can stop the running command immediately
- The STOP Command can stop the running command immediately
- A FAULT (see table in Diagnostic) can stop the running command immediately

- Switch ON Power Supply
- Verify LED Status 1/2 in INIT Mode
- Configure Profile Velocity 0x6060 → 0x03
- Set **READY TO SWITCH ON** State: write Control Word 0x6040 → 0x06
- Set **SWITCHED ON** State: write Control Word 0x6040 → 0x07
- Verify that the Drive is in SWITCHED ON: read Status Word 0x6041
- Set **OPERATION ENABLED** State: write Control Word 0x6040 → 0x0F
- Verify LED STATUS ENABLED
- Verify that the Drive is in ENABLED: read Status Word 0x6041
- Verify that the brake is released
- Write Speed Set Point: 0x60FF → 0x03E8 (for example set 1000 rpm)
- Verify if the motor is running
- Verify the motor speed (after ramp): read 0x606C→ 0x03E8 (for example set 1000 rpm)

3. | MEASURING UNIT CONVERSION

The Lafert Drive is used in different applications. For setting parameters easily in different applications, our clients could use the internal measuring unit conversion module to converse any users parameters into drive's internal unit.

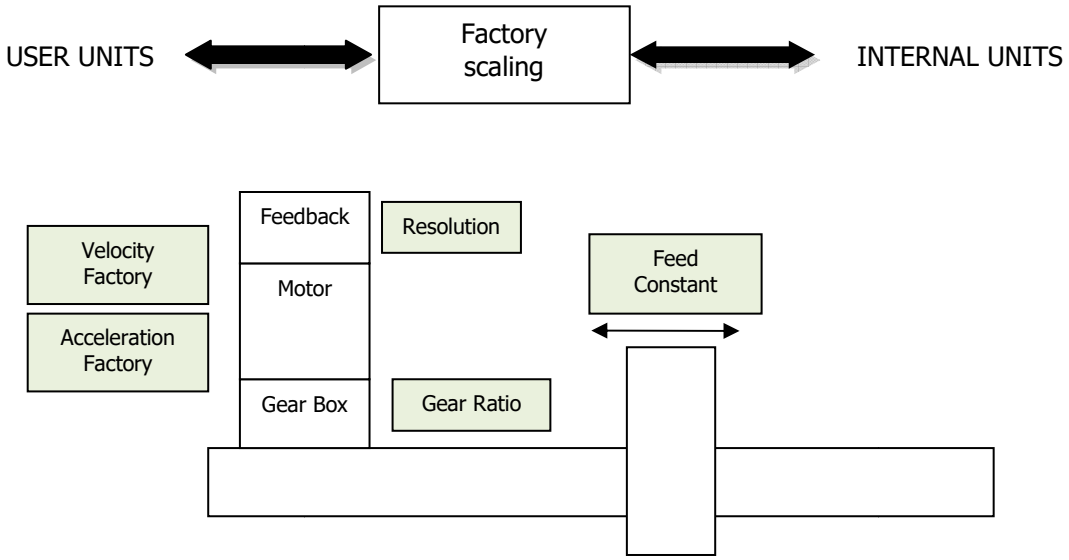


Figure 15 - Factory group

The objects of the Factor Group are used to convert internal position values, speed values and acceleration values into user-defined units.
Internal position values are entered in increments and are dependent on the resolution of the encoder used.
User-defined units depend on the encoder resolution and on attached linear reduction.

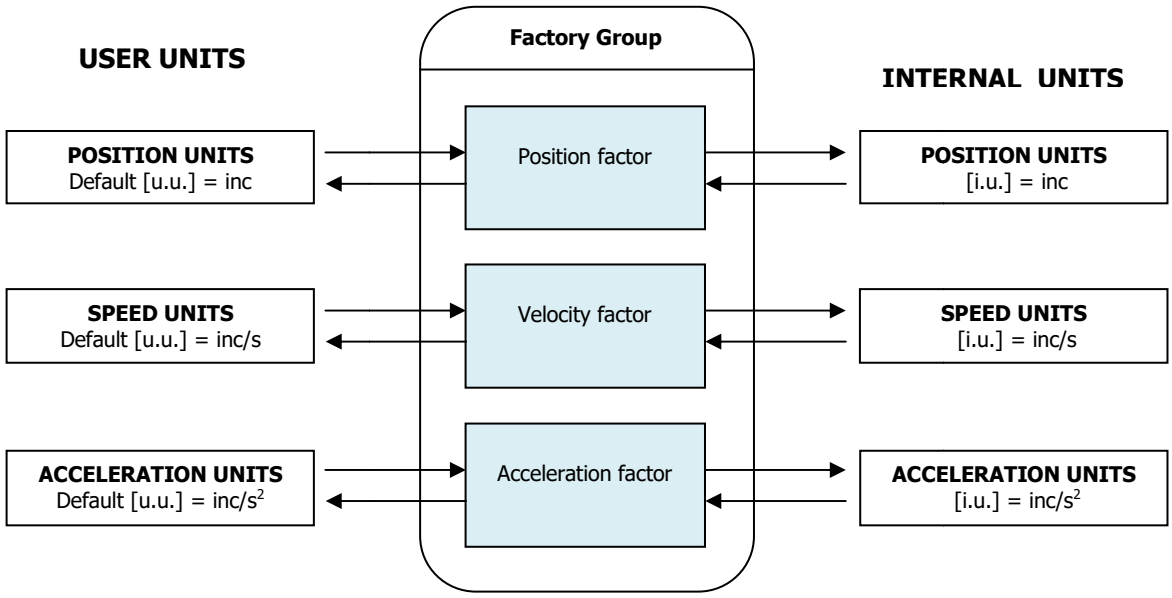


Figure 16 - Factory group units

All parameters are storage in internal units and parameters can be converted in user units using value of factor group.
The default values are following:

Object	Name	Internal Unit	Default User Unit
Length	Position Units	Inc	Inc
Speed	Speed Units	Inc/s	Inc/s
Acceleration	Acceleration Units	Inc/s^2	Inc/s^2

The factors defined in the factor group set up a relationship between device-internal units (increments) and physical units.

It defines [***u.u.***] as *user unit* and [***i.u.***] as *internal units*.

Caution

The drive has as a default unit the increments. If you want to change the units see the procedure. After the units changing It is mandatory modify also some objects (for example profile acceleration/deceleration, max acceleration/deceleration ...)

information

Referring to "APPENDIX" chapter to know the "How to change the user units

MEASURING UNIT CONVERSION PARAMETER:

The factors are the result of the calculation of two parameters called dimension index and notation index.

Index	Name	Object Code	Data Type	Attr.	
$608F_h$	Position encoder resolution	ARRAY	UNSIGNED32	rw	not used
6090_h	Velocity Encoder Resolution	ARRAY	UNSIGNED32	rw	not used
6091_h	Gear Ratio	ARRAY	UNSIGNED32	rw	not used
6092_h	Feed Constant	ARRAY	UNSIGNED32	rw	not used
6096_h	Velocity Factor	ARRAY	UNSIGNED32	rw	used
6097_h	Acceleration Factor	ARRAY	UNSIGNED32	rw	used

Object 6096_h: Velocity factor

This object can be used to match the velocity units to the user-defined velocity units.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6096_h	Velocity Factor	Array	UNSIGNED32	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Value default
00_h	Highest sub-index supported	ro	no	2	2
01_h	Numerator	rw	no	[1 ... 2147483647]	1
02_h	Divisor	rw	no	[1 ... 2147483647]	1

Numerator and divisor of the Velocity Factor has to be entered separately.

$$VelocityFactor = \frac{Numerator}{Divisor}$$

The default value of user unit is [inc/s]: the numerator and the divisor are set "1".

$$Velocity[i.u.] = Velocity[u.u.] \times \left(\frac{60}{Resolution}\right) \times \left(\frac{Numerator}{Divisor}\right)$$

The resolution is the number of measuring segments or units in one revolution of an encoder shaft or 1 in/mm of a linear scale.

Example:

The speed-set point provision is to be made in revolutions per minute (rpm).

$$Velocity[inc/sec] = Velocity[rpm] \times \left(\frac{60}{Resolution}\right) \times \left(\frac{Numerator}{Divisor}\right)$$

If the resolution of encoder is 213 =16384 then the Numerator is 16384 and the Divisor is 60

The factor group used for the following objects:

- $60FF_h$: Target Velocity
- $606D_h$: Velocity Window
- $606F_h$: Velocity Threshold
- 6081_h: Profile Velocity (for Profile Positioner Mode)
- 6082_h: End Velocity (for Profile Positioner Mode)

Caution

If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B19

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)

Object 6097_h: Accelerator factor

This object can be used to match the acceleration units to theuser-defined acceleration units.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6097_h	Acceleration Factor	Array	UNSIGNED32	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Value default
00_h	Highest sub-index supported	ro	no	2	2
01_h	Numerator	rw	no	[1 ... 2147483647]	1
02_h	Divisor	rw	no	[1 ... 2147483647]	1

Numerator and divisor of the Acceleration Factor has to be entered separately.

$$AccelerationFactor = \frac{Numerator}{Divisor}$$

The default value of user unit is *[inc/s^2]:* the numerator and the divisor are set "1".

$$Acceleration[i.u.] = Acceleration\ [u.u.] \times \left(\frac{60}{Resolution}\right) \times \left(\frac{Numerator}{Divisor}\right)$$

The resolution is the number of measuring segments or units in one revolution of an encoder shaft or 1 in/mm of a linear scale.

Example:

The acceleration-set point prevision is to be made in revolutions per minute per second (rpm/s).

$$Acceleration\ [(inc/sec^2] = Acceleration\ [rpm/s] \times \left(\frac{60}{Resolution}\right) \times \left(\frac{Numerator}{Divisor}\right)$$

If the resolution of encoder is 2^{13} =16384 then the Numerator is 16384 and the Divisor is 60

The factor group used for the following objects:

- 6083_h: Profile Acceleration
- 6084_h:Profile Deceleration
- $60C5_h$: Max acceleration
- $60C6_h$: Max deceleration

Caution

If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B1A

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)

4. | SAFETY

The drive moves in the state SAFETY from all states.

To exit by Safety State it is necessary to send the controlword with value "Disable Voltage".

SAFETY OBJECT

The Object 4000_h "Safety State" communicates if the drive is in the safety state and what is the function safety that is occurred. At the moment the STO function is the only safety feature implemented.

Object 4000_h: Safety State

This object is used to communicate the state Safety of the drive. It is only read.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
4000h	Safety State	ARRAY	UNSIGNED16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
0	Number Of Entries	ro	no	-	2
1	Safety State	ro	no	[0,1]	-
2	STO Function	ro	no	[0,1]	-

Value Definition:

Sub-Index	Field	Configuration	Definition
1	Safety State	0_b 1_b	Drive isn't in safety Drive in safety
2	STO Function	0_b 1_b	STO Safety State is not happened STO Safety State is happened

information

Referring to "FUNCTIONS" chapter to know the "SAFETY" function

STATE MACHINE DSP402 WITH SAFETY STATE

The following picture shows the safety state. This state is added in the state machine DSP402. To exit by Safety State it is necessary to send the controlword with value "Disable Voltage".

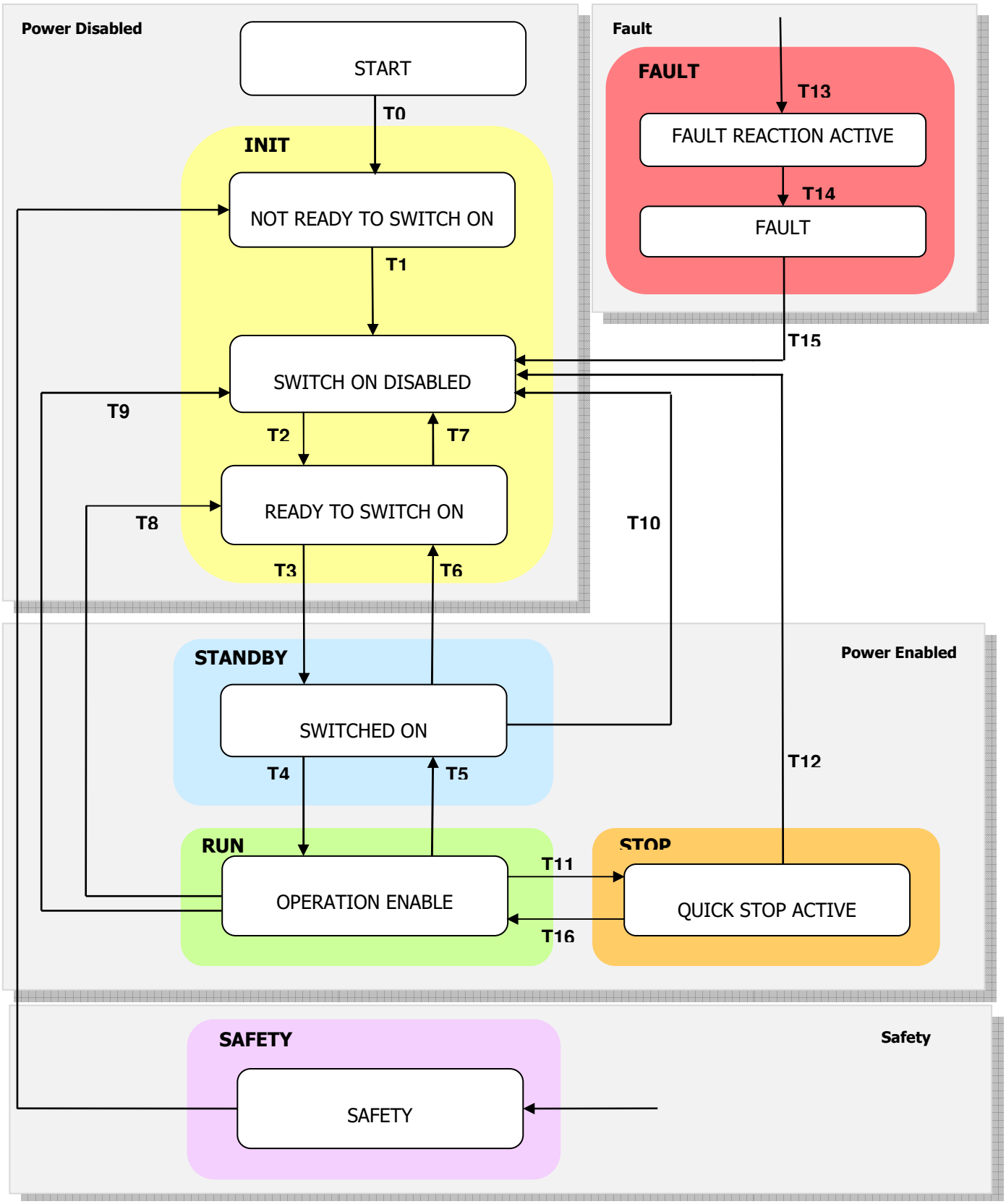
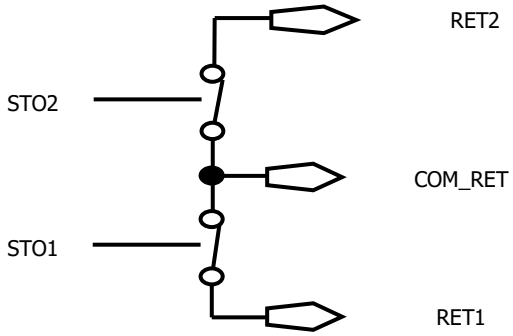


Figure 17 - State Machine DSP402 with Safety State

STO feature

The STO circuit concept uses a two channel architecture. This architecture is shown in the system block diagram below.



The two isolated differential STO inputs have to be connected at 24V voltage to allow that the motor operates. The STO digital inputs status are written in the object digital input $60FD_h$.

Input 1	Input 2	Output 1	Output 2	Output SW
STO1	STO2	RET1	RET2	STATUS
0	0	CLOSE	CLOSE	SAFETY
24V	0	OPEN	CLOSE	SAFETY
0	24V	CLOSE	OPEN	SAFETY
24V	24V	OPEN	OPEN	NORMAL MODE

In the Safety State, the drive will not produce torque or force in the motor. The STO function achieves and maintains a safe state by disabling the ability of the attached motor to produce torque/force.

This both halts any drive induced acceleration already in process and prevents initiation of motion.
The expectation is that an inability of the motor to produce torque/force translates into a reduction of risk of hazardous motion for the larger system.

Caution

The Drive cannot hold the load with the STO function activated because the motor no longer supplies any torque.

- If the STO function is activated during operation, the drive will stop in an uncontrolled manner.
- If the drive has the Safety Torque OFF (STO), verify that this circuit is correctly supplied before all operation functions.

5. | CANOPEN OBJECT DICTIONARY

GENERAL OBJECTS (DS301)

Object 1000_h: Device Type

Object Description:

Index	Name	Object Code	Data Type	Category
1000h	Device Type	VAR	U8	O

Bit MSB 31	Bit LSB 0
Additional Information	Device Profile Number

Default value for Lafert Drive is **0xFF7A0192**, the number 0192h means the device uses the profile 402.

Object 1001_h: Error Register

The error register is a field of 8 bits, each for a certain error type.
If an error occurs the bit has to be set.

BIT	Meaning
0	Generic Error
1	Current
2	Voltage
3	Temperature
4	Communication Error (overrun, error state)
5	Device Profile Specific
6	Reserved
7	Manufacturer Specific

Object 1002_h: Manufacturer status register

This object shall provide a common status register for manufacturer-specific purposes. In thisspecification only the size and the location of this object are defined.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
1002_h	Manufacturer Status Register	VARIABLE	U32	O

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Manufacturer Status Register	ro	no	-

Object 1003_h: Pre-defined Error Field

This object contains an error stack with up to eight entries. It holds errors that have occurred on the device and have been signalled via Emergency Object. It is an error history.

Writing to sub index 0 deletes the entire error history.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
1003_h	Pre-defined Error Field	VARIABLE	U32	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Number of Errors	rw	no	-
01_h	Error Code last alarm occurred	ro	no	-
02_h	Error Code before last alarm	ro	no	-
03_h... FF_h	Error Code Older Alarm	ro	no	-

If a new error occurs, it is entered in sub-index 1. The already existing entries in sub-indices 1 to 15 are moved back one position. The error in sub-index 15 is thereby removed.

The number of errors that have already occurred can be read from the object with sub-index 0.

If no error is currently entered in the error stack, it is not possible to read one of the 15 sub-indices 1–15 and an error is sent in response. The drive responses with an SDO abort message (abort code: 0800 0024_h).

The pre-defined error field has the following structure

Bit MSB 31 – 24	23 – 16	Bit LSB 15 – 0
Manufacturer-specific error code	Error register	Error code

Writing 00h to sub-index 00h shall delete the entire error history (empties the array).
Other values than 00h are not allowed and shall lead to an abort message (error code: 0609 0030_h).

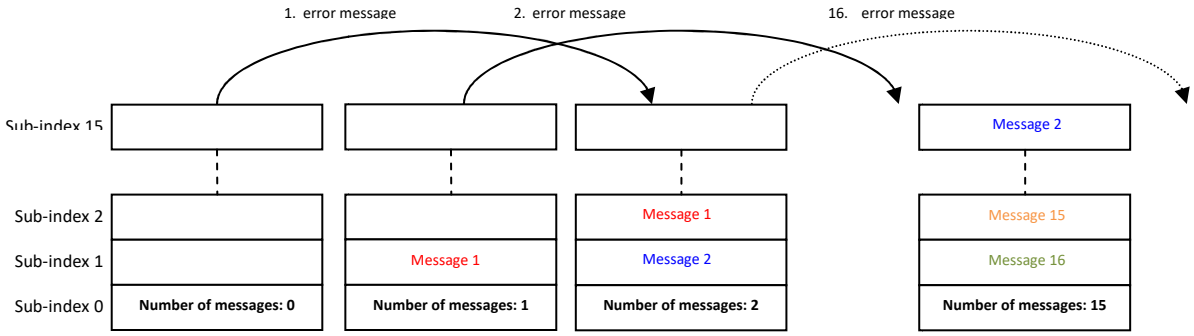


Figure 18 - History Message List

information

Referring to "FUNCTIONS" chapter to know the "Emergency History" function

Object 1005_h: COB-ID SYNC

COB-ID of the synchronization object. The device generates a SYNC message if bit 30 is set. The meaning of other bits is equal to the other communication objects.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
1005_h	COB-ID SYNC	VARIABLE	UNSIGNED32	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	COB-ID SYNC	rw	no	80_h

Object 1008_h: Manufacturer Device Name

This object contains the device name.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
1008_h	Manufacturer Device Name	VARIABLE	STRING (4 char)	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Manufacturer device name	cost	no	-

Object 1009_h: Manufacturer Hardware Version

This object contains the device hardware version.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
1009_h	Manufacturer Hardware Name	VARIABLE	STRING (4 char)	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Manufacturer Hardware name	cost	no	-

Object $100A_h$: Manufacturer Software Version

This object contains the device software version.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
$100A_h$	Manufacturer Software Name	VARIABLE	STRING (4 char)	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Manufacturer Software name	cost	no	-

Object $100C_h$: Guard Time

This entry contains the guard time in milliseconds. The value 0 switches node guarding off.

The guard time multiplied with the life time factor object 100Dh gives the life time for the life guarding protocol.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
$100C_h$	Guard Time	VARIABLE	U16	optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Guard Time	rw	no	[0 - 65535]	0

Caution

The Heartbeat protocol has a higher priority than Node guarding. If both protocols are activated simultaneously, the Node Guarding Timer is suppressed, but no EMCY message is sent either.

Object $100D_h$: Life Time Factor

The Life Time Factor multiplied by the Guard Time Object 100Ch gives the Life Time for the Node Guarding. The value 0 switches the Node Guarding off.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
$100A_h$	Life Time Factor	VARIABLE	U8	optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Life Time Factor	rw	no	[0 - 255]	0

Object 1010_h: Store Parameters Field

This object supports the saving of parameters in non volatile memory. By read access the device provides information about its saving capabilities.

Several parameter groups are distinguished.

- *Sub index 0:* contains the largest Sub-Index that is supported
- *Sub index 1:* refers to all parameters that can be stored on the device.
- *Sub index 2:* refers to communication related parameters (Index 1000h - 1FFFh manufacturer specific communication parameters).
- *Sub index 3:* refers to application related parameters (Index 6000h - 9FFFh manufacturer specific application parameters).
- *Sub index 4 - 127:* manufacturers may store their choice of parameters individually.
- *Sub-Index 128 – 254:* are reserved for future use.

This command can only be carried out if the module isn't in "operation enabled" or "Quick Stop". If the command can't be processed then the drive sends a warning message and set a '1' the third bit of warning object (see Object 2003h: Warning)

Object Description:

Index	Name EDS	Object Code	Data Type	Category
1010_h	Store Parameter Field	ARRAY	U32	optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Number of Entries	c	no	5
01_h	Save all Parameters	rw	no	0
02_h	Save Communication Parameters	rw	no	0
03_h	Save Application Parameters	rw	no	0
04_h	Save Manufacturer Parameters	rw	no	0
05_h	Reserved	rw	no	0

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate Sub-Index. The signature is "save": 0x65766173.

Storage writing access structure:

Signature ISO 8859 ("ASCII")	e	v	a	s
hex	65h	76h	61h	73h

On reception of the correct signature in the appropriate sub-index the CANopen device shall storethe parameter and then it shall confirm the SDO transmission (SDO download initiate response).

If the storing failed, the CANopen device shall respond with the SDO abort transfer service (abort code: 0606 0000h). If a wrong signature is written, the CANopen device shall refuse to store and it shall respond with the SDO abort transfer service (abort code: 0800 002xh).

On read access to the appropriate sub-index the CANopen device shall provide information about its storage functionality with the following format:

Bit MSB		Bit LSB
31 2	1	0
Reserved	Auto	Cmd

Structure of read access:

Bit	Field	Configuration	Definition
0	Cmd	0_b 1_b	CANopen device does not save parameters on command CANopen device saves parameters on command
1	Auto	0_b 1_b	CANopen device does not save parameters autonomously CANopen device saves parameters autonomously

Object 1011_h: Restore default parameters

This entry supports restoring of default parameters. With a read access the device provides information about its capabilities to restore these values.

Several parameter groups are distinguished.

- *Sub index 0:* contains the largest Sub-Index that is supported

- *Sub index 1:* Restore all factory settings
- *Sub index 2:* Restore all factory settings for communications parameters (0x0000 to 0x1FFF)
- *Sub index 3:* Restore all factory settings for application parameters (from 0x2000)
- *Sub index 4- 127:* manufacturer defined parameters

Object Description:

Index	Name EDS	Object Code	Data Type	Category
1010_h	Store Parameter Field	ARRAY	U32	optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Number of Entries	c	no	5
01_h	Restore all Default Parameters	rw	no	0
02_h	Restore Communication Default Parameters	rw	no	0
03_h	Restore Application Default Parameters	rw	no	0
04_h	Restore Manufacturer Parameters	rw	no	0
05_h	Reserved	rw	no	0

The object "Restore Default Parameters" loads the standard configuration parameters. The standard configuration parameters are either that as delivered or those last saved. Read access supplies information about the restore options. For restoring the signature "load" (**0x64616f6c**) must be written.

"Load" signature:

Signature ISO 8859 ("ASCII")	d	a	o	l
hex	64h	61h	6Fh	6Ch

On reception of the correct signature in the appropriate sub-index the CANopen device shall restore the default parameters and then it shall confirm the SDO transmission (SDO download CANopen application layer and communication profile initiate response).

If the restoring failed, the CANopen device shall respond with the SDO abort transfer service (abort code: 0606 0000h). If a wrong signature is written, the CANopen device shall refuse to restore the defaults and shall respond with the SDO abort transfer service (abort code: 0800 002xh).

The default values shall be set valid after the CANopen device is reset (NMT service reset node for sub-index from 01h to 7Fh, NMT service reset communication for sub-index 02h) or power cycled.

On read access to the appropriate sub-index the CANopen device shall provide information about its default parameter restoring capability with the following format:

Bit MSB 31 ... 1	Bit LSB 0
Reserved	CMD

Structure of read access:

Bit	Field	Configuration	Definition
0	Cmd	0_b 1_b	CANopen device does not restore default parameters CANopen device restores parameters

Object 1014_h: COB-ID Emergency Message

Object Description:

Index	Name	Object Code	Data Type	Category
1010_h	COB-ID EMCY	VAR	U32	optional

Entry Description:

Sub-Index	Access	PDO mapping	Default Value
00_h	ro	no	0x80 + Id Node

Object 1017_h: Producer Heartbeat Time

The producer heartbeat time defines the cycle time of the heartbeat. If the time is 0 it is not used. The time has to be a multiple of 1 msec.

Object Description:

Index	Name	Object Code	Data Type	Category
1017_h	Producer Heartbeat Time	VAR	U16	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Producer Heartbeat Time	rw	no	[0 – 65535]	0

Data Byte for NMT state evaluation of the HeartBeat producer:

- 0 (00h): "Boot-Up"
- 4 (04h): "Stopped"
- 5 (05h): "Operational"
- 127 (7Fh) "Pre-operational

Object 1018_h: Identity object

This object shall provide general identification information of the CANopen device.

- **Sub-index 01_h**: shall contain the unique value1 that is allocated uniquely to each vendor of a CANopen device. The value 0000 0000h shall indicate an invalid vendor-ID.
- **Sub-index 02_h:** shall contain the unique value that identifies a specific type of CANopen devices. The value of 0000 0000_h shall be reserved.
- **Sub-index 03_h:** shall contain the major revision number and the minor revision number of the revision of the CANopen device. The major revision number shall identify a specific CANopen behaviour. That means if the CANopen functionality is different, the major revision number shall be incremented. The minor revision number shall identify different versions of CANopen device with the same CANopen behaviour. The value of 0000 0000_h shall be reserved.
- **Sub-index 04_h:** shall contain the serial number that identifies uniquely a CANopen device within a product group and a specific revision. The value of 0000 0000h shall be reserved.

Object Description:

Index	Name	Object Code	Data Type	Category
1018_h	Identity object	RECORD	Identity	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Highest sub-index supported	ro	no	4
01_h	Vendor-ID	ro	no	$000001FB_h$
02_h	Product code	ro	no	reserved
03_h	Revision number	ro	no	reserved
04_h	Serial number	ro	no	reserved

Object 1400_h: Receive PDO1 Communication Parameter

It contains the communication parameters of the current PDO1 the device is able to receive.

Object Description:

Index	Name	Object Code	Data Type	Category
1400_h	Receive PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object 1401_h: Receive PDO2 Communication Parameter

It contains the communication parameters of the current PDO2 the device is able to receive.

Object Description:

Index	Name	Object Code	Data Type	Category
1401_h	Receive PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object 1402_h: Receive PDO3 Communication Parameter

It contains the communication parameters of the current PDO3 the device is able to receive.

Object Description:

Index	Name	Object Code	Data Type	Category
1402_h	Receive PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF

02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object 1403_h: Receive PDO4 Communication Parameter

It contains the communication parameters of the current PDO4 the device is able to receive.

Object Description:

Index	Name	Object Code	Data Type	Category
1403_h	Receive PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object 1600_h: Receive PDO1 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
1600_h	Receive PDO1 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	rw	no	UNSIGNED8	0..8
01_h	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03_h	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04_h	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05_h	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06_h	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07_h	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08_h	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

Object 1601_h: Receive PDO2 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
1601_h	Receive PDO2 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	rw	no	UNSIGNED8	0..8
01_h	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03_h	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04_h	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05_h	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06_h	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07_h	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08_h	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

Object 1602_h: Receive PDO3 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
1602_h	Receive PDO3 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	rw	no	UNSIGNED8	0..8
01_h	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03_h	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04_h	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05_h	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06_h	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07_h	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08_h	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

Object 1603_h: Receive PDO4 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
1603_h	Receive PDO4 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	rw	no	UNSIGNED8	0..8
01_h	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03_h	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04_h	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05_h	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06_h	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07_h	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08_h	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

Object 1800_h: Transmit PDO1 Communication Parameter

It contains the communication parameters of the current PDOn the device is able to transmit.

Object Description:

Index	Name	Object Code	Data Type	Category
1800_h	Transmit PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object 1801_h: Transmit PDO2 Communication Parameter

It contains the communication parameters of the current PDOn the device is able to transmit.

Object Description:

Index	Name	Object Code	Data Type	Category
1801_h	Transmit PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object 1802_h: Transmit PDO3 Communication Parameter

It contains the communication parameters of the current PDOn the device is able to transmit.

Object Description:

Index	Name	Object Code	Data Type	Category
1802_h	Transmit PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object 1803_h: Transmit PDO4 Communication Parameter

It contains the communication parameters of the current PDOn the device is able to transmit.

Object Description:

Index	Name	Object Code	Data Type	Category
1803_h	Transmit PDO Communication Parameter	PDO_COMM_PAR	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	ro	no	UNSIGNED8	5
01_h	COB-ID	ro	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Transmission Type	ro	no	UNSIGNED8	0..0xFF
03_h	Inhibit Time	ro	no	UNSIGNED16	0..0xFFFF
04_h	Compatibility Entry	ro	no	UNSIGNED8	0..0xFF
05_h	Event Timer	ro	no	UNSIGNED16	0..0xFFFF

Object $1A00_h$: Transmit PDO1 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
$1A00_h$	Transmit PDO1 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	rw	no	UNSIGNED8	0..8
01_h	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03_h	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04_h	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05_h	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06_h	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07_h	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08_h	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

Object $1A01_h$: Transmit PDO2 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
$1A01_h$	Transmit PDO2 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Number of Entries	rw	no	UNSIGNED8	0..8
01_h	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02_h	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03_h	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04_h	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05_h	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06_h	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07_h	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08_h	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

Object 1A02$_h$: Transmit PDO3 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
1A02$_h$	Transmit PDO3 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00$_h$	Number of Entries	rw	no	UNSIGNED8	0..8
01$_h$	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02$_h$	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03$_h$	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04$_h$	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05$_h$	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06$_h$	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07$_h$	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08$_h$	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

Object 1A03$_h$: Transmit PDO4 Mapping Parameter

The sub-indices contain the indices, the sub-indices and the lengths of the mapped sub-indices.

Their structure is as follows: Value: index (16bit) + sub-index (8bit) + length (8bit)

Object Description:

Index	Name	Object Code	Data Type	Category
1A03$_h$	Transmit PDO4 Mapping Parameter	PDO_MAPPING	RECORD	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00$_h$	Number of Entries	rw	no	UNSIGNED8	0..8
01$_h$	Mapping Entry 1	rw	no	UNSIGNED32	0..0xFFFFFFFF
02$_h$	Mapping Entry 2	rw	no	UNSIGNED32	0..0xFFFFFFFF
03$_h$	Mapping Entry 3	rw	no	UNSIGNED32	0..0xFFFFFFFF
04$_h$	Mapping Entry 4	rw	no	UNSIGNED32	0..0xFFFFFFFF
05$_h$	Mapping Entry 5	rw	no	UNSIGNED32	0..0xFFFFFFFF
06$_h$	Mapping Entry 6	rw	no	UNSIGNED32	0..0xFFFFFFFF
07$_h$	Mapping Entry 7	rw	no	UNSIGNED32	0..0xFFFFFFFF
08$_h$	Mapping Entry 8	rw	no	UNSIGNED32	0..0xFFFFFFFF

MANUFACTURER OBJECTS - SETTINGS PARAMETERS

Object 2000_h: Id-Node

The object allows the user to set the CAN IdNode of the Node, the change takes effect at next power cycle.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2000_h	IdNode	VAR	UNSIGNED8	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	CAN IdNode	rw	no	1 ... 127	1

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)

This object can be changed and saved in e^2prom memory.

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 2000_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

information

Referring to "APPENDIX" chapter to know "How to change Id-Node"

Object 2001_h: CAN Baud Rate

The object allows the user to set the CAN bit rate of the Node, the change takes effect at next power cycle.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2001_h	CAN Baudrate	VAR	UNSIGNED16	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	CAN Baudrate	rw	no	See table	0x03E8

Valid entries:

BaudRate	Entry	Lafert Servo Drive
10 kBit/s	0x000A	Available
20 kbit/s	0x0014	Available
50 kbit/s	0x0032	Available
100 kbit/s	0x0064	Available
125 kbit/s	0x007D	Available
250 kbit/s	0x00FA	Available
500 kbit/s	0x01F4	Available
800 kbit/s	0x0320	Available
1000 kbit/s	0x03E8	Available

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop"
- Write the new value in SDO object 2001_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

information

Referring to "APPENDIX" chapter to know "How to change BaudRate"

Object 3001_h: Absolute Limits Parameters

This object describes the Absolute Limits. These parameters are only in reading because they are set by manufacturer.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3001_h	Absolute Limits Parameters	ARRAY	UNSIGNED32	Optional

Entry Description:

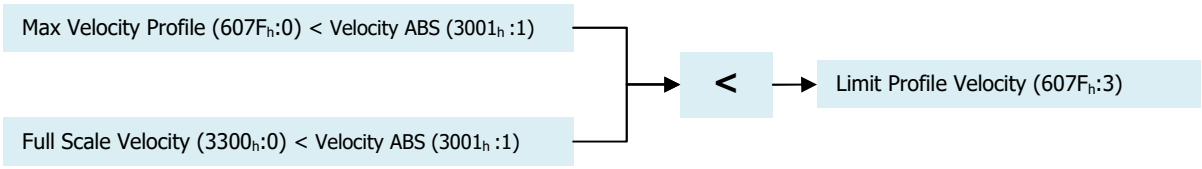
Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	ro	no	-	5
1	Velocity ABS	ro	no	[0 - 2147483647] rpm	defined by application
2	Acceleration ABS	ro	no	[0 - 2147483647] rpm/s	defined by application
3	Limit Profile Velocity	ro	no	[0 - 65535] rpm	defined by application
4	free	ro	no	-	-
5	free	ro	no	-	-

Value Definition:

Sub-Index	Field	Configuration	Definition
1	Velocity ABS	[rpm]	It is maximum absolute value of Velocity profile. It is a limit for $607F_h$ (Max Velocity Profile).
2	Acceleration ABS	[rpm /s]	It is maximum absolute value of acceleration profile. It is a limit for $60C5_h$ (Max Acceleration) and $60C6_h$ (Max Deceleration).
3	Limit Profile Velocity (Min Value)	[rpm]	It is a Limit Velocity for Profile Mode. It is a minimum between $607F_h$ (Max Velocity Profile) and 3300_h (Full Scale Velocity)

These parameters are the maximum rating of drive and they are only reading.

- The velocity parameters have to be lower than the "velocity ABS" object (607Fh:1).
  For example: if the "Velocity ABS" is 4500 rpm then the "Max Velocity Profile" (6071fh:0) will be smaller or equal 4500 rpm.
- The acceleration parameters have to be lower than the acceleration "ABS object" (607Fh:2). For example: if the "Acceleration ABS" is 2228 rpm/s then the "Max Acceleration" (60C5h:0) will be smaller or equal 2228 rpm/s.
- The "Limit Profile Velocity" (607Fh:3) is the limit value of profile velocity, in fact this object is the minimum between 607Fh (Max Velocity Profile) and 3300h (Full Scale Velocity). Therefore, the "Target Velocity" (60FFh:0) will be limited by the "Limit Profile Velocity" (607Fh:3).



The drive will sent the follow abort codes:

- 0x06090011 = sub-index does not exist

Object 3002h: Motor Brake Parameters

This object describes the parameters of Brake Configuration.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3002_h	Brake Parameters	ARRAY	INTEGER16	Mandatory IF

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	ro	no	-	7
01_h	Motor Brake Option	rw	no	[0,1]	defined by application
02_h	Motor Brake Delay	rw	no	[1... 32767]	defined by application
03_h	Unlock Motor Brake	rw	no	[1 ... 32767]	defined by application
04_h	Brake timeout	rw	no	[1 ... 32767]	defined by application
05_h	Automatic/Manual Mode Configuration	rw	no	[0,1]	defined by application
06_h	Motor Brake Status	ro	no	[0,1]	-
07_h	Brake Type	ro	no	[1,2]	defined by application

Value Definition:

Sub-Index	Field	Configuration	Definition
1	Motor Brake Option (*)	0_b 1_b	Motor Brake disabled or Motor Brake is not present Motor Brake enabled
2	Motor Brake Delay	[ms * 10]	Delay open command. This timeout is the delay between STBY Status and unlock brake.
3	Unlock Brake time	[ms * 10]	Delay between STOP and RUN mode before unlock Brake. This timeout depends by kind of motor brake.
4	Brake timeout	[ms * 10]	Only without Dynamic Brake (see object 0x3007) Max time programmed for natural Inertia deceleration. At the end of this timeout the brake is locked and drive will be in STBY status.

5	Automatic/Manual Mode Configuration	0_b 1_b	Automatic Mode Activated Manual Mode Activated
6	Motor Brake Status	0_b 1_b	Brake Status: activated → Motor is locked Brake Status: released → Motor is not locked
7	Brake Type	1 2	Magnetic Brake Spring Brake

Caution

(*) If the motor does not have the brake, this value has 0 as default value.
The user CAN'T enable because it is not present.

The follow graph describes in Automatic Mode the timing of the brake when the drive moves from "Switched-On" (STANDBY state) to "Operation Enabled" (RUN state).

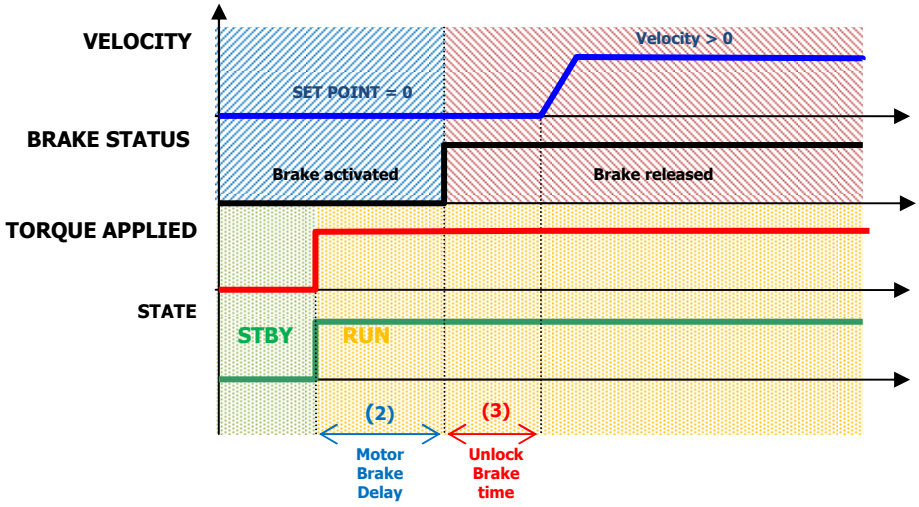


Figure 19 - Brake timeframe "Switched-On" state to "Operation Enabled" State

The follow graph describes in Automatic Mode the timing of the brake when the drive moves from "Operation Enabled" (RUN state) to "Switched-On" (STANDBY state).

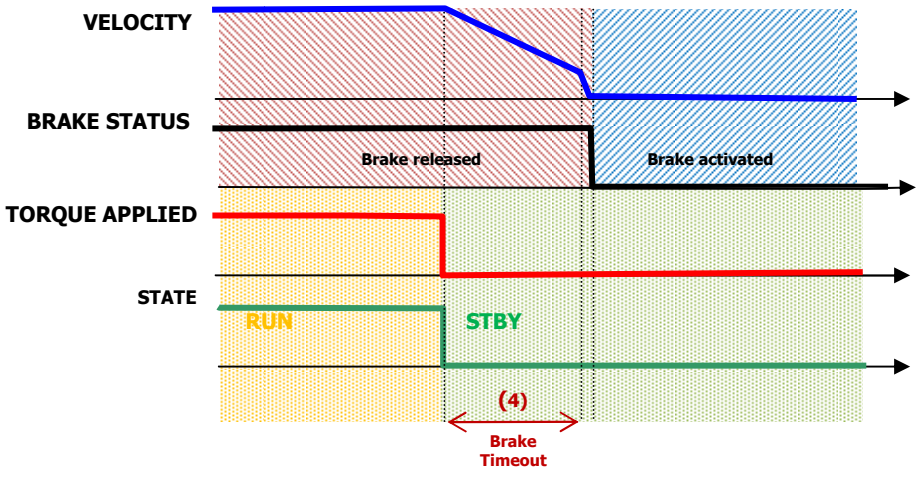


Figure 20 - Brake timeframe "Operation Enabled" State to "Switched-On" State

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)
- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06010002 = written is not permitted because the value is only READ (for object 0x3002:6 and 0x3002:7)
- 0x06090011 = sub-index does not exist

It is possible to change the Brake Parameters in run time.

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop"
- Write the new value in SDO object 3002_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

information

Referring to "FUNCTIONS" chapter to know the management "Motor Brake Management"

Object 3007_h: Dynamic Brake Parameters

This object describes the parameters of Dynamic Brake.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3007_h	Dynamic Brake Parameters	ARRAY	INTEGER16	Mandatory IF

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	ro	no	-	7
01_h	Dynamic Brake Option	rw	no	[0,1]	1
02_h	Holding Torque Time	rw	no	[1 ... 32767]	defined by application
03_h	Dynamic Brake Status	ro	no	[0,1]	defined by application
04_h	Decrement step ramp	rw	no	[1 ... 8191]	defined by application
05_h	Max Timeout Dynamic Brake	rw	no	[1 ... 32767]	defined by application

Value Definition:

Sub-Index	Field	Configuration	Definition
1	Dynamic Brake Option	0_b 1_b	Dynamic Brake Mode Activated Dynamic Brake Mode Deactivated
2	Holding Torque Time	[ms * 10]	This time is the delay between STOP Status and unlock brake, at the end of deceleration ramp, before to stay in STBY status.
3	Dynamic Brake Status	0_b 1_b	Drive is not in Dynamic Brake Drive is in Dynamic Brake

4	Decrement step ramp	[rpm*100/sec]	This number is the step to decrement the Set Point during the transition from Run to Standby with Dynamic Brake activated
5	Max Timeout Dynamic Brake	[ms * 10]	Max Dynamic brake Timeout is the maximum time to exit from condition dynamic brake. It must be higher than "Decrement Step Ramp"

The follow graph describes the timing stop of drive when the ELECTRONIC DYNAMIC BRAKE is applied.

This condition will be present when the drive state move from RUN state (operation enabled in DSP402) to STANDBY state (Switched ON in DSP402).

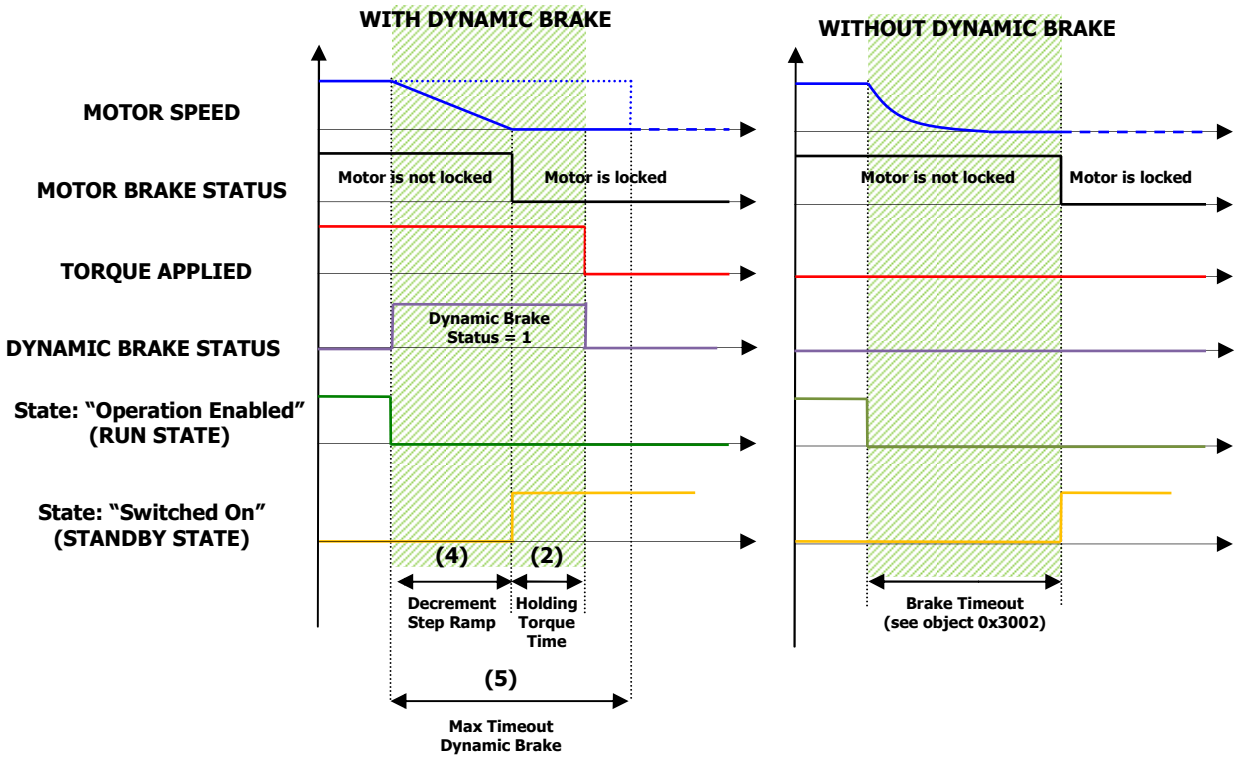


Figure 21 - Dynamic Brake timeframe

The movement from RUN state in dynamic brake (if it is activated) is possible when

- Set the 6040_h (controlword) object in "disable operation" (move to standby state)
- Set the "input 3 Emergency Enable" if the option Digital Input 3 is applied (see 3008_h object)
- the alarm is occurred

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)
- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06090031 = Value of parameter written too high (for object 0x3007:4 because it must be smaller than 0x3007:5)
- 0x06090032 = Value of parameter written too low (for object 0x3007:5 because it must be greater than 0x3007:4)
- 0x06010002 = written is not permitted because the value is only READ (for object 0x3007:3)

- 0x06090011 = sub-index does not exist

It is possible to change the Brake Parameters in run time.
This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop"
- Write the new value in SDO object 3007_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

information

Referring to "FUNCTIONS" chapter to know the "Dynamic Brake Management" function

Object 3008_h: Emergency Enable Parameters

This object describes the parameters to enable the feature of digital input number 3.

The digital input number 3 can be configured as enabling signal hardware to move from "Operation Enable" state [RUN] to "Switched On" state [STANDBY].

It can be considered as emergency signal but it isn't safety certificated (for disabling the power in safety certificated condition referred to STO chapter of Drive User Guide).

If the function "emergency enable" is implemented than the digital input 3 is used to move in the state machine of DSP402:

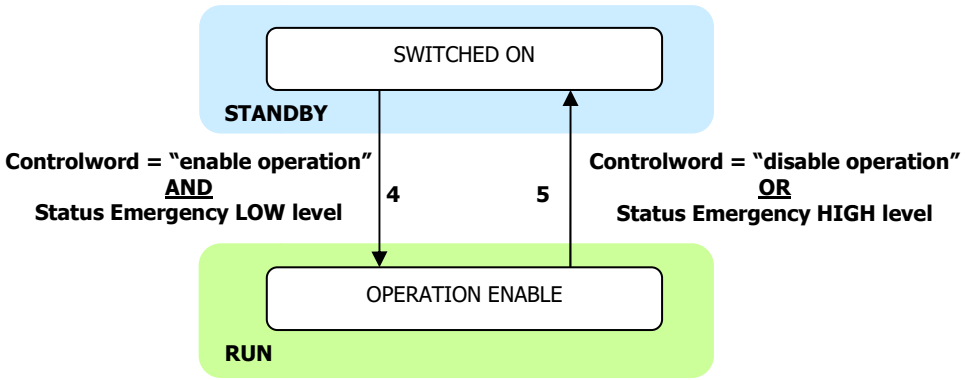


Figure 22 - Emergency enable configuration

CANopen State Transition:

Transition 4: SWITCHED ON → OPERATION ENABLE
To define in [Controlword: 6040_h] with "Enable Operation" value AND digital Input 3 in low level hardware:

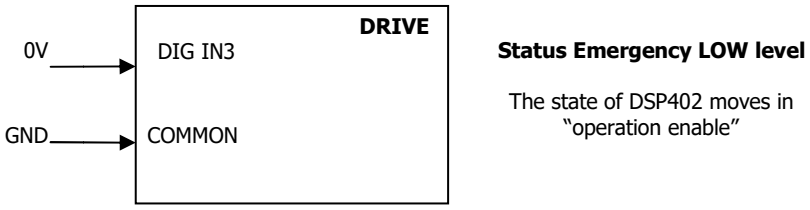


Figure 23 - Emergency Enable Status Low Level

Transition 5: OPERATION ENABLE → SWITCHED ON
To define in [Controlword: 6040_h] with "Disable Operation" value OR digital Input 3 in High Level Level

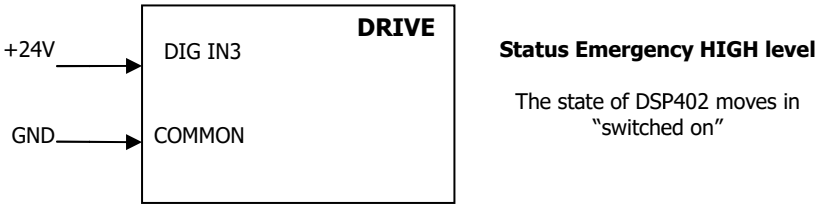


Figure 24 - Emergency Enable Status High Level

If the function "emergency enable" is not used than the digital In 3 is configured as general purpose input. It can be changed the configuration level.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3008_h	Emergency Enabling Input Parameters	ARRAY	INTEGER16	Mandatory IF

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	ro	no	-	7
01_h	Emergency Enable Option	rw	no	[0,1]	1
02_h	Emergency Input Neg	rw	no	[0,1]	0
03_h	Emergency Input Status	ro	no	[0,1]	0
04_h	free	rw	no		-
05_h	free	rw	no		-
06_h	free	rw	no		-
07_h	free	rw	no		-

Value Definition:

Sub-Index	Field	Configuration	Definition
1	Emergency Enable Option	0_b 1_b	Digital Input 3 is configured as General Purpose Digital Input 3 is configured as Emergency Enable
2	Emergency Input Neg	0_b 1_b	None inversion of input emergency level Inversion of input emergency level
3	Emergency Status	0_b 1_b	Status Low Level: Emergency not active Status High Level: Emergency active

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)
- 0x06010002 = written is not permitted because the value is only READ (for object 0x3007:3)
- 0x06090011 = sub-index does not exist

This object can be changed and saved in e^2prom memory.

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Acrive"
- Write the new value in SDO object 3008_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

information

Referring to "FUNCTIONS" chapter to know the management "Input Emergency Enable"

Object 3050_h: Analog Output 1

This object describes the analog output parameters. The analog output capability of the drive is [0÷10]V.

The analog output configuration can be set by user:

- 0 = analog output disabled
- 1 = analog output configured as "general purpose". The analog output gets a value from 0 to 10V following [0 ÷ 4095] bit. The digital input value can be written via CANopen (sub-index 2).
- 2 = analog output configured as "Velocity monitoring". The analog output gets a proportional signal voltage of speed monitoring. The output signal is [0 - 10] V and it matches with the value range [-Velocity Full Scale ÷ Velocity Full Scale] rpm (object 3300_h).
- 3 = analog output configured as "Current monitoring". The analog output gets a proportional signal voltage of current absorbited. The output signal is [0 - 10] V and it matches with the value range [0 ÷ Peak Current] (object 3303_h: 2).

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3050_h	Analog Output 1	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	3	3
01_h	Configuration	rw	no	[0 ... 2]	-
02_h	Digital Value	rw	no	[0 ... 4095]	-
03_h	Output	ro	no	[0 ... 4095]	-

Value Definition:

Sub-Index	Field	Value	Definition
1	Configuration	0 1 2 3	Disabled (Analog Output is 0) General purpose (Analog Output is proportional to digital Value) Velocity Monitoring (Analog Output is proportional to Actual Velocity) Current Monitoring (Analog Output is proportional toActua current absorted)
2	Digital Value	0 ... 4095	Value to set analog output [0 ...4095] → [0 ... 10] V
3	Output DAC	0 ... 4095	Output Monitoring in BIT

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)
- 0x06040030 = the value is out of range
- 0x05040001 = command is invalid because the configuration is not "general purpose" type
- 0x06010002 = written is not permitted because the value is only READ
- 0x06090011 = sub-index does not exist

This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop"
- Write the new value in SDO object 3050_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

information

Referring to "FUNCTIONS" chapter to know the management "DAC Monitoring"

Object 3200_h: Current PID

The object controls equivalent of PID current parameters.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3200_h	Current Pid	ARRAY	INTEGER16	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	ro	no	6	6
01_h	PidCur Kp	rw	no	[1 ... 32767]	defined by application
02_h	PidCur Ki	rw	no	[1 ... 32767]	defined by application
03_h	PidCur Kv	rw	no	[1 ... 32767]	defined by application
04_h	PidCur Kd (reserved)	ro	no	[1 ... 32767]	(reserved)
05_h	PidCur N (reserved)	ro	no	[1 ... 32767]	(reserved)
06_h	PidCur FF (reserved)	ro	no	[1 ... 32767]	(reserved)

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)
- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06090011 = sub-index does not exist

It is possible to change the Current Pid in run time.
This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 3200_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 3201_h: Speed PID

The object controls equivalent of PID speed parameters.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3201_h	Speed PID	ARRAY	INTEGER16	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	ro	no	6	6
01_h	PidVel Kp	rw	no	[1 ... 32767]	defined by application
02_h	PidVel Ki	rw	no	[1 ... 32767]	defined by application
03_h	PidVel Kv	rw	no	[1 ... 32767]	defined by application
04_h	PidVel Kd (reserved)	ro	no	[1 ... 32767]	(reserved)
05_h	PidVel N (reserved)	ro	no	[1 ... 32767]	(reserved)
06_h	PidVel FF (reserved)	ro	no	[1 ... 32767]	(reserved)

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)
- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06090011 = sub-index does not exist

It is possible to change the Speed Pid in run time.
This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 3201_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 3202_h: Position PID

The object controls equivalent of PID position parameters.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3202_h	Position Pid	ARRAY	INTEGER16	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	ro	no	9	9
01_h	PidPos Kp	rw	no	[1 ... 32767]	defined by application
02_h	PidPos Ki	rw	no	[1 ... 32767]	defined by application
03_h	PidPos Kv	rw	no	[1 ... 32767]	defined by application
04_h	PidPos FF Ra V (reserved)	ro	no	[1 ... 32767]	(reserved)
05_h	PidPos FF Ra A (reserved)	ro	no	[1 ... 32767]	(reserved)
06_h	PidPos FF Vr V (reserved)	ro	no	[1 ... 32767]	(reserved)
07_h	PidPos FF Rd A (reserved)	ro	no	[1 ... 32767]	(reserved)
08_h	PidPos FF Rd V (reserved)	ro	no	[1 ... 32767]	(reserved)
09_h	PidPos Tc (reserved)	ro	no	[1 ... 32767]	(reserved)

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)
- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06090011 = sub-index does not exist

It is possible to change the Position Pid in run time.
This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 3202_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 3300_h: Velocity Full Scale

That is value is the Full Scale of Velocity.
In analog mode it is tha maximum value of reference voltage for Speed Set Point.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3300_h	Velocity Full Scale	VARIABLE	UNSIGNED16	Mandatory

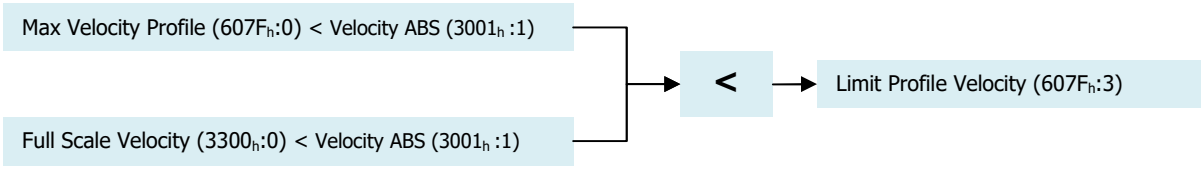
Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number of Entries	rw	no	[0-32767]	Defined by Application

In Analog Mode this object defines the range to set the target of velocity. The analog set point is the input 0 to 10V where the range is defined by the +/- "Velocity Full scale" (3300_h).

You should program the "Velocity Full Scale" to be smaller than "Velocity Absolute Maximum Rating" (3001_h: 1).

In Profile Velocity Mode via CAN this object, together the "Max Profile Velocity" ($607F_h$), defines the limit of Speed. The scheme to set the limit is the follow:



The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06090031 = Value of parameter written too high (it must be smaller than 0x3001:1)

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 3300_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

MANUFACTURER OBJECTS – RUNTIME MONITORING DATA

Object 2002_h: Drive Control State

This object communicates the drive's state. This object defines exactly the PWM control motor.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2002h	Drive Control State	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range
00_h	Drive Control State	ro	no	See table

Valid entries:

VALUE	BIT	Name	Description
0x0001	1	Run Velocity	The motor runs in velocity control mode
0x0002	2	Standby	The drive is in stand-by. The PWM is OFF.
0x0004	3	Stop	The drive is in stop. It is stationary with torque applied.
0x0008	4	Off	Not used

0x0010	5	Alarm	The drive has detected an alarm
0x0020	6	Run Current	The motor runs in Current control mode.
0x0040	7	Init	The drive is in initialization state. The PWM is OFF.
0x0080	8	Safe	The drive is in safe with STO applied. The PWM is OFF.
0x0100	9	Run Positioner	The drive is positioner control mode

Object 2003_h: Warning

This object logs the drive's warnings. To clear the warnings, set fault reset bit (#7) in Controlword (6040_h).

The warnings defined are the following:

- Warning communications CAN
- Warning i^2t Limit
- Warning E^2prom
- Warning Update Parameters
- Warning Factory Parameters
- Warning CANopen parameters
- Warning DAC configuration
- Warning Temperature
- Warning Limitation Torque Funcion

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2003_h	Warning	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range
00_h	Drive Warning	ro	no	See table

Warning List:

Bit	Name	Description
0	Node Guarding	Master losts Node Guarding Message
1	I2T Limit	Drive is in limitation i2T
2	command store/restore/load E^2prom	command store/restore/load are disabled
3	Update Parameters Manufacturer	Request update by canopen is not permission (only RS232)
4	Factory parameters Writing	Factory parameters area is "free": It must be written
5	Alarm CANopen Disabled	alarms canopen are disabled
6	Init object CanOpen	init configuration CANopen object
7	DAC configuration	Configuration DAC is not valid
8	Heat Sink Temperature	Warning Heat Sink Temperature
9	Logic Board Temperature	Warning Logic Board Temperature
10	Motor Temperature	Warning Motor Temperature
11	Can Bus Communication Error	Warning Error Bus CAN (Passive or Busoff)
12	Function Torque Limitation	Warning Function Torque Limitation Activated
13	Torque in Limitation	Warning Tq Limitation
14...31	free	free

Object 2004_h: State Lafert Servo Drive Machine

This object describes exactly the drive state. The drive follows a finite state machine proprietary Lafert Serve Drive that it is compliant with profile DSP402.

State Value Definition:

- 0 = Lafert Servo Drive state INIT - p402 state not ready to switch on
- 1 = Lafert Servo Drive state SAFETY
- 2 = Lafert Servo Drive state STOP - p402 state quick stop active
- 3 = Lafert Servo Drive state RUN - p402 state operation enabled
- 4 = Lafert Servo Drive state STANDBY - p402 state switched on
- 5 = Lafert Servo Drive state DYNAMIC BRAKE
- 6 = Lafert Servo Drive state Reserved
- 7 = Lafert Servo Drive state Reserved
- 10= Lafert Servo Drive state INIT - p402 state switch on disabled
- 11= Lafert Servo Drive state INIT - p402 state ready to switch on
- 16 = Lafert Servo Drive state FAULT - p402 state fault reaction active
- 17 = Lafert Servo Drive state FAULT - p402 state fault
- 18 = Lafert Servo Drive state FAULT - p402 state error

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2004h	Drive Status LSD	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range
00_h	Drive State Lafert Servo Drive	ro	no	See table

Valid entries:

Value	Lafert Servo Drive State	State DSP402
0	**INIT**	**not ready to switch on**
1	**SAFETY**	**-**
2	**STOP**	**quick stop active**
3	**RUN**	**operation enabled**
4	**STANDBY**	**switched on**
5	**DYNAMIC BRAKE**	**-**
6	Reserved	-
7	Reserved	-
8	-	-
9	-	-
10	**INIT**	**state switch on disabled**
11	**INIT**	**ready to switch on**
12	-	-
13	-	-
14	-	-
15	-	-
16	**FAULT**	**fault reaction active**
17	**FAULT**	**fault**
18	**FAULT**	**error**

Object 2030_h: Temperature Drive

This object communicates the drive temperature.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2030_h	Temperature Drive	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Unit
00_h	Drive temperature	ro	no	[-150 ... 1250]	[°C /10]

Object 2031_h: Temperature Motor

This object communicates the motor temperature.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2031_h	Temperature Motor	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Unit
00_h	Motor temperature	ro	no	[-400 ... 1300]	[°C /10]

Object 2032_h: Temperature Heat Sink

This object communicates the Heat Sink temperature.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2032_h	Temperature Heat Sink	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Unit
00_h	Heat Sink temperature	ro	no	[-400 ... 1300]	[°C /10]

Object 2041_h: Voltage Bus

This object communicates the value of voltage Bus.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2041_h	Voltage Bus	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Unit
00_h	Voltage Bus	ro	no	[0 ... 11000]	[V/100]

Object 2050_h: Torque Current

This object communicates the value of Torque Current.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2050_h	Torque Current	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Unit
00_h	Torque Current	ro	no	[-32767... 32767]	[A/100]

Object 2051_h: Power Drive

TBD

Object 2052_h: Power Motor

TBD

Object 2053_h: Velocity Filtered

This object communicates the value of Velocity filtered.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2053_h	Velocity Filtered	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Unit
00_h	Velocity Filtered	ro	no	[-32767... 32767]	[rpm/4]

Object 2060_h: Impulse

This object is the electrical angle (it depends on number of motor pole pairs) with increment units, the max value is the feedback's resolution.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
2060_h	Impulse	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Unit
00_h	Impulse	ro	no	[-32767... 32767]	[0 – Max Resolution]

Object 3004_h: Feedback Parameters

This object defines the specifics characteristics of Feedback. It is only READ.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3004_h	FeedBack Parameters	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	2	2
01_h	Feedback Type	ro	no	[0 ... 32767]	defined by application
02_h	Resolution	ro	no	[0 ... 32767]	defined by application

Value Definition:

Sub-Index	Field	Configuration	Definition
01_h	Feedback Type	[0 ... 2]	0 = Resolver 1 = Incremental Encoder 2 = Sin/Cos Encoder
02_h	Resolution	[0 – 32767]	Feedback Resolution

The drive will sent the follow abort codes:

- 0x06090011 = sub-index does not exist

Object 3006_h: Motor Specific Settings

This object defines the specifics characteristics of motor. It is only READ.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3006_h	Motor Specific Settings	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	3	3
01_h	Motor Part Number	ro	no	[0 – 32767]	defined by application
02_h	Max Motor Speed	ro	no	[0 – 32767]	defined by application
03_h	N Pole	ro	no	[0 – 32767]	defined by application
04_h	Motor Kt	ro	no	[0 – 32767]	**Not available yet**
05_h	Motor Sense Type	ro	no	[0 – 32767]	defined by application

Motor Sense Type can be the following values:

1. KTY83
2. PT1000
3. KTY84
4. PTC ON/OFF

Object 3020_h: Digital Input Function

This object describes the function of digital Inputs.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3020_h	Digital Input Function	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	6	6
01_h	Configuration Enable	ro	no	[0, 32767]	defined by application
02_h	State Function	ro	no	[0, 32767]	defined by application
03_h	Level Function	ro	no	[0, 32767]	defined by application
04_h	free	-	-	-	-
05_h	free	-	-	-	-

Every bit of value is the function that the digital input can be used.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	POL	V/C	VEL3	VEL2	VEL1	DCCW	DCW	RST	EMGY	STOP	RUN

Where

- BIT 0 - function "RUN": in Analog Mode (manufacturer Mode without CANopen communication) this input is the command to move the drive in RUN state.
- BIT 1 - function "STOP": in Analog Mode (manufacturer Mode without CANopen communication) this input is the command to move the drive in STOP state.
- BIT 2 – function "EMERGENCY INPUT ENABLE": when the option of digital input 3 is defined "Emergency Input Enable" this input is the command to move the drive in STANDBY state.
- BIT 3 – function "RESET": when this function is enabled the input configured can put the drive in reset (it is an hardware reset). If the digital input is configured as reset, the reset has a filter with 100ms.
- BIT 4 – function "DCW": this function is to configure the input with an actuator for clockwise. If the input state is 1 the drive goes in STOP state.
- BIT 5 – function "DCCW": this function is to configure the input with an actuator for counter clockwise. If the input state is 1 the drive goes in STOP state
- BIT 6 – function "VEL1": ***this function is not implemented yet***
- BIT 7 – function "VEL2": ***this function is not implemented yet***
- BIT 8 – function "VEL3": ***this function is not implemented yet***
- BIT 9 – function "V/C": ***this function is not implemented yet***
- BIT 10 – function "POL": ***this function is not implemented yet***

Value Definition:

Sub-Index	Field	Configuration	Definition
1	Configuration Function	Every bit can be '0' or '1'	Every bit is the function configuration: 0 = the function is not enabled 1 = the function is enabled
2	Function State	Every bit can be '0' or '1'	Every bit is the state of the function: 0 = the function actives 1 = the function doesn't active
3	Function Level	Every bit can be '0' or '1'	Every bit is the level of the input function: 0 = positive edge (standard configuration) 1 = negative configuration

Caution

- RUN and STOP are function for digital Input for Analog Mode
- EMERGENCY INPUT ENABLE is the function to move the drive in "Switched On" state of DSP402 (or "STANDBY" state of the MSM of LSD) from the "Operation Enabled" state of DSP402 (or "RUN" state of the Macro State Machine of LSD). This command is used for emergency stop. See object 3008h "Emergency Enable Parameters"
- In Analog Mode it is mandatory to have one digital input configured in "Run" function

Object 3021_h: Digital Input 1

This object describes the digital Input 1 configuration. This object is only READ. To change digital Input configuration contact manufacturer.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3021_h	Digital Input 1	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	5	5
01_h	Configuration	ro	no	[0..6]	defined by application
02_h	State	ro	no	[0,1]	defined by application
03_h	Level	ro	no	[0,1]	defined by application
04_h	free				
05_h	free				

Value Definition:

Sub-Index	Field	value	Definition
1	Configuration	[0 .. 6]	0 = none function 1 = Digital Input configured as "RUN" function 2 = Digital Input configured as "STOP" function 3 = Digital Input configured as "EMERGENCY" function 4 = Digital Input configured as "RESET" function 5 = Digital Input configured as "DCW" function 6 = Digital Input configured as "DCCW" function
2	State	[0,1]	0 = Low Level Digital Input Not active 1 = High Level Digital Input Active
3	Level	[0,1]	0 = positive edge (standard configuration) 1 = negative edge

The drive will sent the follow abort codes:

- 0x06090011 = sub-index does not exist

Object 3022_h: Digital Input 2

This object describes the digital Input 2 configuration. This object is only READ. To change digital Input configuration contact manufacturer.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3022_h	Digital Input 2	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	5	5
01_h	Configuration	ro	no	[0..6]	defined by application
02_h	State	ro	no	[0,1]	defined by application
03_h	Level	ro	no	[0,1]	defined by application
04_h	free				
05_h	free				

Value Definition:

Sub-Index	Field	value	Definition
1	Configuration	[0 .. 6]	0 = none function 1 = Digital Input configured as "RUN" function 2 = Digital Input configured as "STOP" function 3 = Digital Input configured as "EMERGENCY" function 4 = Digital Input configured as "RESET" function 5 = Digital Input configured as "DCW" function 6 = Digital Input configured as "DCCW" function
2	State	[0,1]	0 = Low Level Digital Input Not active 1 = High Level Digital Input Active
3	Level	[0,1]	0 = positive edge (standard configuration) 1 = negative edge

The drive will sent the follow abort codes:

- 0x06090011 = sub-index does not exist

Object 3023_h: Digital Input 3

This object describes the digital Input 3 configuration. This object is only READ. To change digital Input configuration contact manufacturer.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3023_h	Digital Input 3	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	5	5
01_h	Configuration	ro	no	[0..6]	defined by application
02_h	State	ro	no	[0,1]	defined by application
03_h	Level	ro	no	[0,1]	defined by application
04_h	free				
05_h	free				

Value Definition:

Sub-Index	Field	value	Definition
1	Configuration	[0 .. 6]	0 = none function 1 = Digital Input configured as "RUN" function 2 = Digital Input configured as "STOP" function 3 = Digital Input configured as "EMERGENCY" function 4 = Digital Input configured as "RESET" function 5 = Digital Input configured as "DCW" function 6 = Digital Input configured as "DCCW" function
2	State	[0,1]	0 = Low Level Digital Input Not active 1 = High Level Digital Input Active
3	Level	[0,1]	0 = positive edge (standard configuration) 1 = negative edge

The drive will sent the follow abort codes:

- 0x06090011 = sub-index does not exist

Object 3024_h: Digital Input 4

This object describes the digital Input 4 configuration. This object is only READ. To change digital Input configuration contatct manufacturer.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
3024_h	Digital Input 4	ARRAY	INTEGER16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Number Of Entries	ro	no	5	5
01_h	Configuration	ro	no	[0..6]	defined by application
02_h	State	ro	no	[0,1]	defined by application
03_h	Level	ro	no	[0,1]	defined by application
04_h	free				
05_h	free				

Value Definition:

Sub-Index	Field	value	Definition
1	Configuration	[0 .. 6]	0 = none function 1 = Digital Input configured as "RUN" function 2 = Digital Input configured as "STOP" function 3 = Digital Input configured as "EMERGENCY" function 4 = Digital Input configured as "RESET" function 5 = Digital Input configured as "DCW" function 6 = Digital Input configured as "DCCW" function
2	State	[0,1]	0 = Low Level Digital Input Not active 1 = High Level Digital Input Active
3	Level	[0,1]	0 = positive edge (standard configuration) 1 = negative edge

The drive will sent the follow abort codes:

- 0x06090011 = sub-index does not exist

Object 6402_h: Motor Type

This object indicates the type of motor attached to and driven by the drive device.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6402_h	Motor Type	VARIABLE	UNSIGNED16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Motor Type	rw	no	0 – 0xFFFF	-

Value Definition:

Sub-Index	Field	Definition
0000_h	non-standard motor	-
0001_h	phase modulated DC motor	-
0002_h	frequency controlled DC motor	-
0003_h	PM synchronous motor	-
0004_h	FC synchronous motor	AC synchronous sinewave wound field

0005_h	switched reluctance motor	AC synchronous reluctance switched
0006_h	wound rotor induction motor	AC asynchronous induction polyphase wound rotor
0007_h	squirrel cage induction motor	AC asynchronous induction squirrel cage
0008_h	stepper motor	AC synchronous step
0009_h	micro-step stepper motor	-
$000A_h$	sinusoidal PM BL motor	AC synchronous sinusoidal PM
$000B_h$	trapezoidal PM BL motor	AC synchronous brushless PM trapezoidal
$000C_h$	AC synchronous reluctance sync	-
$000D_h$	DC commutator PM	-
$000E_h$	DC commutator wound field series	-
$000F_h$	DC commutator wound field shunt	-
0010_h	DC commutator wound field compound	-
0011_h to $7FFE_h$	Reserved	
$7FFF_h$	no motor type assigned	-
8000_h-$FFFF_h$	manufacturer-specific	-

Object 6403_h: Motor Catalogue Number

This object indicates the motor catalogue number (nameplate number) provided by the motor manufacturer. If the number is not assigned yet, this object shall indicate this by /0 (empty string).

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6403_h	Motor Catalogue Number	VARIABLE	STRING	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Motor Catalogue Number	rw	no	4	'000'

Valid entries:

Value	Size
'-'	
'S'	Small
'M	Medium
'L'	Large
'C1'	Custom 1
'C2'	Custom 2
'C3'	Custom 3
'C4'	Custom 4

Object 6404_h: Motor Manufacturer

This object indicates the name of the motor manufacturer. If the name is not assigned yet, this object shall indicate this by /0 (empty string).

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6404_h	Motor Manufacturer	VARIABLE	STRING	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Motor Manufacturer	rw	no	4	'000'

Valid entries:

Value string	Size	Motor product code

'NaN'	None	Not defined
'B40'	Medium	B40E4J – C1078
'B63'	Large	B6304K – H32mm – 48Vdc
'B71'	Small	B7108Q – H40mm – 48Vdc

Object 6502_h: Supported Drive Modes

This object provides information on the supported drive modes.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6502_h	Supported Drive Modes	VARIABLE	U32	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Supported Drive Modes	ro	no	[0 – 32767]	4

Valid entries:

	Value	Lafert Servo Drive
bit 0	profile position mode	
bit 1	velocity mode	
bit 2	profile velocity mode	SUPPORTED
bit 3	profile torque mode	SUPPORTED
bit 4	reserved	
bit 5	homing mode	
bit 6	interpolated position mode	
bit 7	cyclic synchronous position mode	
bit 8	cyclic synchronous velocity mode	
bit 9	cyclic synchronous torque mode	
bit 10-15	reserved	
bit 16	manufacturer-specific – Analog Mode	SUPPORTED
bit 17-31	manufacturer-specific	

PROFILE OBJECTS DSP402

Object 603F$_h$: Error code

This object shall provide the error code of the last error which occurred in the drive device.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
603F$_h$	Error code	VAR	U16	Optional

Entry Description:

Sub-Index	Name	Access	PDO mapping	Value Range	Default Value
00$_h$	Error code	ro	no	See table Emergency	-

The 603F$_h$ object is the error code of (last) alarm occurred. The meaning is described in the Table 21 - Emergency Description of Emergency Chapter. The column "Error Code" is the corresponding value.

Object 6040$_h$: Controlword

This object is used to control the CiA-402 FSA, CiA-402 modes and manufacturer-specific entities.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6040$_h$	Controlword	VAR	UNSIGNED16	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00$_h$	Control word	rw	YES (default)	See table	-

This object is organized bit-wise. The bits have the following meaning:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ms					r	oms	h	fr	oms			eo	qs	ev	so
MSB															LSB

Bits	Definition	Name
0	so	Switch ON
1	ev	Enable Voltage
2	qs	Quick Stop
3	eo	Enable Operation
4, 5, 6,9	oms	Operation mode specific
7	fr	Fault Reset
8	h	Halt
9	oms	Operation mode specific
10	r	reserved
11, 12, 13, 14, 15	oms	manufacturer specific

Commands description:

Command	Bit of the controlword					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
	Fault Reset	Enable Operation	Quick Stop	Enable Voltage	Switch On	
Shutdown	0	X	1	1	0	2, 6, 8
Switch On	0	0	1	1	1	3
Switch ON	0	1	1	1	1	3 (note 2)
Disable Voltage	0	X	X	0	X	7, 9, 10, 12
Quick Stop	0	X	0	1	X	7, 10, 11
Disable Operation	0	0	1	1	1	5
Enable Operation	0	1	1	1	1	4, 16
Fault Reset (note 1)	↑	X	X	X	X	15

NOTE

- (note 1) Reset Fault occurred to exit from FAULT state - **Not Available**
- (note 2) Automatic transition to enable operation state after executing switched on state functionality - **Not Available**

BIT	Manufacturer specific Bits		
	Name Value	VALUE	Description
4, 5, 6,9	Operation mode specific	0_b 1_b	These bit are different meaning as profile mode selected
8	Halt	0_b 1_b	The commanded motion shall be continued if possible. The commanded motion shall beinterrupted
11	Warning Acknowledge	0_b 1_b	If 1 than It cancels the warning bit in the status word
12, 13, 14, 15	manufacturer specific		free

Caution

Between two transitions you wait at least 50ms

Object 6041_h: Statusword

This object is used to indicate the current state of the FSA, the operation mode and manufacturer-specific entities.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6041_h	Statusword	VAR	UNSIGNED16	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Status word	ro	YES (default)	See table	-

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

Bits	Definition	Name
0	rtso	Ready to switch On
1	so	Switched On
2	oe	Operation Enabled
3	f	Fault
4	ve	Voltage Enabled
5	qs	Quick Stop
6	sod	Switch on disabled
7	w	Warning
8	ms	Manufacturer specific
9	rm	Remote
10	tr	Target reached
11	ila	Internal Limit Active
12, 13	oms	operation mode specific
14, 15	ms	Manufacturer Specific

Bits description:

BIT	Manufacturer specific Bits		
	Name Value	VALUE	Description
0, 1, 2, 3, 5, 6	statusword	x0xx 0000 $_b$	Not Ready to switch On
		x1xx 0000 $_b$	Switch On disabled
		x01x 0001 $_b$	Ready to switch on
		x01x 0011 $_b$	Swutched on
		x01x 0111 $_b$	Operation enabled
		x00x 0111 $_b$	Quick Stop Active
		x0xx 1111 $_b$	Fault Reaction Active
		x0xx 1000 $_b$	Fault
4	Voltage Enabled	0_b 1_b	Vbus is smaller than Value of "underVoltage" Vbus is greater than Value of "underVoltage"
5	quick stop	0_b 1_b	The drive is reacting on aquick stop request The drive is not in QUICK STOP
7	Warning	0_b 1_b	No warning is present (Warning is not an error or fault) At least warning is occurred (To refer at warning list in object 2003h to know the warning occurred)
8	Emergency Input Enable	0_b 1_b	Input Emergency Function is not enabled Input Emergency Function is enabled
10	Target Reached	0_b 1_b	The set-point has not been reached yet. The drive has reachedthe set-point.
11	Internal Limit Active	0_b 1_b	Indicate that an i2T limit is not active Indicate that an i2T limit is active
14	Drive Safety	0_b 1_b	drive in NORMAL mode (not safe e not fault) drive in SAFETY mode
15	Drive Fault	0_b 1_b	drive in NORMAL mode (not safe e not fault) drive in FAULT, one alarm is detected

Bits operation mode description:

BIT	Operation Mode					
	Velocity mode	Profile Position mode	Profile Velocity Mode	Profile Torque Mode	Homing Mode	Interpolated Position Mode
12	Reserved	Set-point acknowledge	Speed	Reserved	Homing Attained	Ip mode active
13	Reserved	Following error	Max slippage error	Reserved	Homing Error	reserved

Object 6060_h: Modes of Operation

The operational mode is selectable by this object.
This object shows only the value of the requested operation mode, the actual operation mode of the PDS is reflected in the object [Mode of Operation Display: 6061_h]

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6060_h	Modes of Operation	VAR	INTEGER8	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Mode of operation	rw	YES (default)	See table	-128 to 10

The following value definition is valid:

BIT	Meaning	Lafert Servo Drives
0	no mode change / no mode assigned	
1	profile position mode	
2	velocity mode	
3	profile velocity mode	**Available**
4	profile torque mode	**Available**
5	Reserved	
6	homing mode	
7	interpolated position mode	
8	cyclic synchronous position mode	
9	cyclic synchronous velocity mode	
10	cyclic synchronous torque mode	
-1	manufacturer-specific (analog or hardware control)	**Available**
-2	manufacturer-specific (reserved for test)	**Available**

The Manufacturer-specific is value **(-1)** and It defines the mode operation in analog or hardware control.

The drive will sent the follow abort codes:

- 0x060B0002 = the written is not possible because the drive hah the torque applied (state is "operation enabled" or "Quick stop Active")
- 0x05040001 = command is invalid because the mode is not supported

Object 6061_h: Modes of Operation Display

This object provides the actual operation mode.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
6061_h	Modes of Operation Display	VAR	INTEGER8	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Data Type	Value Range
00_h	Mode of operation display	ro	YES (default)	See table	[-128 to 10]

The following value definition is valid:

BIT	Meaning
0	no mode change / no mode assigned

1	profile position mode
2	velocity mode
3	profile velocity mode
4	profile torque mode
5	Reserved
6	homing mode
7	interpolated position mode
8	cyclic synchronous position mode
9	cyclic synchronous velocity mode
10	cyclic synchronous torque mode
-1	manufacturer-specific (analog or hardware)
-2	manufacturer-specific (reserved for test)

Object $607E_h$: Polarity

This object influences the sign of: [Position Demand Value: 6062h] and/or [Velocity Demand Value: 606Bh]

Object Description:

Index	Name EDS	Object Code	Data Type	Category
$607E_h$	Polarity	VAR	UNSIGNED8	Mandatory

Entry Description:

Sub-Index	Description	Access	PDO mapping	Value Range	Default Value
00_h	Polarity	rw	no	0 ÷ 192	00h

Bits:

BIT	Meaning
0 .. 5	reserved
6	Velocity Polarity
7	Position polarity

The following value definition is valid:

- bit value = 0: multiply the demand value by 1
- bit value = 1: multiply the demand value by -1

The drive will sent the follow abort codes:

- 0x08000002 = the written is disabled by manufacturer (it is a option defined by application)

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object $607E_h$
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object $60FD_h$: Digital inputs

This object shall provide digital inputs. The low word contains the states of the digital inputs as defined by the CANopen 402 profile. The high word displays the states of all digital inputs.

The status of digital inputs is output by object $60FD_h$:

- Limit or reference switch for Homing Profile (not implemented)
- Digital Input 1, 2, 3, 4 programmable or defined by application
- Safe Torque Off (STO)

Object Description:

Index	Name EDS	Object Code	Data Type	Category
$60FD_h$	Digital Inputs	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Name	Access	PDO mapping	Value Range
00_h	Digital Inputs	ro	no	0 ÷ 0xFFFFFFF

Bits Structure:

Bit MSB							**Bit LSB**
31	16	15	4	3	2	1	0
Digital Input Status Manufacturer Specific		reserved		Interlock	Home switch	Pos limit switch	Neg limit switch

Data Description:

BIT	Configuration	Value	Definition	Note
0	Negative limit switch	0_b 1_b	Negative limit switch not reached Negative limit switch reached	If the function "DCW" is configured then this bit is the state of digital input
1	Positive limit switch	0_b 1_b	Positive limit switch not reached Positive limit switch reached	If the function "DCCW" is configured then this bit is the state of digital input
2	Home switch	0_b 1_b	Home switch not reached Home switch reached	Not Used
3	Interlock	0_b 1_b	Interlock not activated Interlock activated	Not Used
4 ... 15	reserved	-	-	
16	Digital Input - DigIn1	0_b 1_b	Read Status: Low Level Read Status: High Level	It depends by Function configured
17	Digital Input – DigIn2	0_b 1_b	Read Status: Low Level Read Status: High Level	It depends by Function configured
18	Digital Input – DigIn3	0_b 1_b	Read Status: Low Level Read Status: High Level	It depends by Function configured
19	Digital Input – DigIn4	0_b 1_b	Read Status: Low Level Read Status: High Level	It depends by Function configured
20	Digital Input – STO1	0_b 1_b	Read Status: Low Level Read Status: High Level	Digital Input connected to STO circuit
21	Digital Input – STO2 (*)	0_b 1_b	Read Status: Low Level Read Status: High Level	(*) Digital Input NOT connected. It is always High Level.
20 ... 31	Digital Input	-	-	Not Available

Object $60FE_h$: Digital outputs

This object shall command the digital outputs. This object shall represent the logical output levels.

Object Description:

Index	Name EDS	Object Code	Data Type	Category
$60FE_h$	Digital Outputs	ARRAY	U32	Optional

Entry Description:

Sub-Index	Name	Access	PDO mapping	Value Range	Default Value
00_h	Highest sub-index supported	c	no	[1, 2]	2
01_h	Physical outputs	rw	possible	0	00000000_h
02_h	Bit Mask	rw	no	0	00000000_h

Bits Structure of sub-index 01_h:

Bit MSB 31 ... 16	15 ... 1	Bit LSB 0
Digital Output Command Manufacturer - specific	reserved	Motor Brake Command

Value Definition for sub-index 01_h:

BIT	Configuration	Value	Definition	Note
0	Motor Brake Command	0_b 1_b	Brake Activated → Motor Locked Brake Released → Motor Free	It is available if the Brake is in "Manual Mode"
1 ... 15	reserved (each bit)	-	Reserved	-
16	Digital Output1 – Status Drive	0_b 1_b	Switched off – Drive is Fault State Switched on – Drive is OK	This Output is connected to Status Drive
17	Digital Output 2	0_b 1_b	Switched off Switched on	Available
18	Digital Output 3	0_b 1_b	Switched off Switched on	Available
19	Digital Output 4	0_b 1_b	Switched off - Brake Activated Switched on – Brake Released	This output is connected to Status Brake

Bits Structure of sub-index 02_h:

Bit MSB 31 ... 16	15 ... 1	Bit LSB 0
Digital Output Enable/Disable Manufacturer - specific	reserved	Motor Brake Management

Value Definition for sub-index 02_h:

BIT	Configuration	Value	Definition	Note
0	Motor Brake Management	0_b 1_b	Disable output Enable output	It is ever enabled
1 ... 15	reserved	-	Reserved	-
16	Enable Digital Output1	0_b 1_b	Disable output Enable output	It is ever enabled
17	Enable Digital Output 2	0_b 1_b	Disable output Enable output	It is ever enabled
18	Enable Digital Output 3	0_b 1_b	Disable output Enable output	It is ever enabled
19	Enable Digital Output 4	0_b 1_b	Disable output Enable output	It is ever enabled

The sub-index 2 is only READ. The outputs are ever enabled.

6. | CANOPEN OPERATION MODES

MODES OF OPERATIONS

The Drive has the modes of operation below:

- **PROFILE POSITION** (Not Available)

The Drive in this mode is able to make movements in relation to a defined target position. Set Value number 1 of "Mode Of Operation" object (6060_h)

- **PROFILE VELOCITY**

The Drive, in this mode, is able to follow a velocity set point without requiring the definition of a target position. Set Value number 3 of "Mode Of Operation" object (6060_h)

- **PROFILE TORQUE**

The Drive, in this mode, is able to follow a Current set point without requiring the definition of a target position. Set Value number 4 of "Mode Of Operation" object (6060_h)

- **PROFILE HOMING** (Not Available)

Use this mode to define an homing position. Set Value number 6 of "Mode Of Operation" object (6060_h)

- **ANALOG MODE**

The Drive in this mode the Drive state is determined by commands transition like mode 'Profile Velocity Mode' but the speed id determined by analog input.
Set Value number -1 of "Mode Of Operation" object (6060_h)

The operating mode is selected with the object 0x6060 whose change is implemented only at speeds zero while in "Profile velocity mode" and "Homing Mode", while only at target reached for the "Position mode".

PROFILE POSITION MODE (1) (not available)

Object 6064_h: Position actual value

This object shall provide the actual value of the position measurement device.
This object is 0 on the power-on.

Object Description:

Index	Object Code	Data Type	Category
6064_h	VAR	INTEGER32	mandatory if pp

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	ro	YES	-	no	[inc]

information

Referring to "APPENDIX" chapter to know the description of "POSITION MONITORING"

PROFILE VELOCITY MODE (3)

In the Profile Velocity Mode (PV) the speed of the drive is controlled by a PID controller. This ensures that the drive is operated without deviation from the specified values, provided it is not overloaded.

Prerequisites for the drive to be operated in PV Mode:

- The Profile Velocity Mode must be set in the "Mode of Operation" (6060_h) parameter (value "3").
- The drive must be in *"Operation Enabled"* state of state machine of DSP402, verify it with the object "Statusword" (6041_h). To move the state machine it uses the object "controlword" (6040_h)
- Target Velocity and parameters of Profile Velocity Mode must be set correctly.

The target velocity is set via the "Target Velocity" ($60FF_h$) object in the object dictionary.

In Profile Velocity Mode the drive directly follows each new transferred set-point value.
At the same time, the set maximum values for acceleration, deceleration ramp and speed are also taken into account.

Controller structure in Profile Velocity Mode:

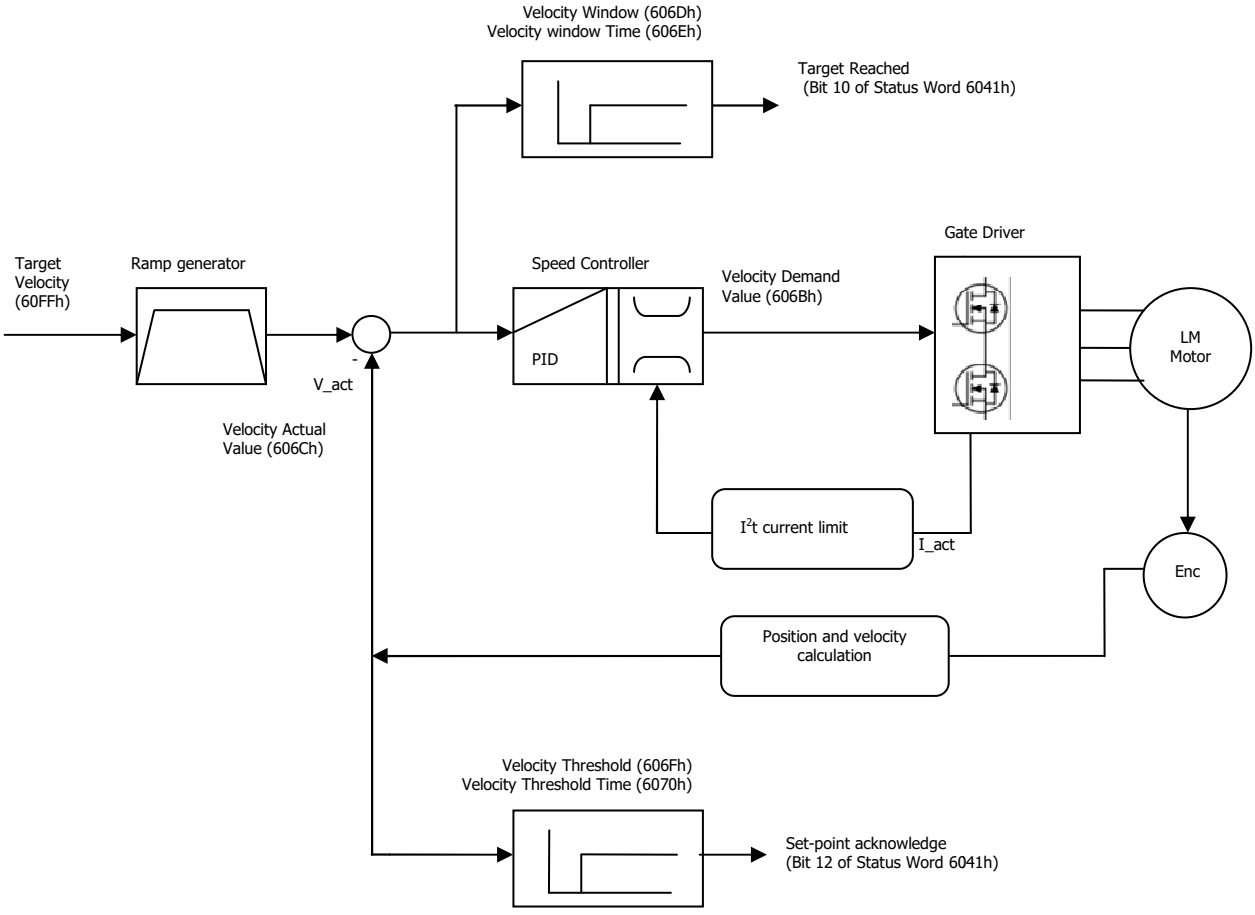


Figure 25 - Controller structure for Profile Velocity

The associated objects to control the drive in Profile Velocity Mode are the following:

Index	Sub Index	Name	READ / WRITE	M/O	Data Type	PDO	Available
0x603F	0	Error Code	RO	O	U16	-	x
0x6040	0	Control Word	R/W	M	U16	RPDO	x
0x6041	0	Status Word	RO	M	U16	TPDO	x
0x6060	0	Modes of Operation	R/W	M	I8	RPDO	x
0x6061	0	Modes of Operation Display	RO	M	I8	TPDO	x
0x60FF	0	Target Velocity	R/W	M	I32	RPDO	x
0x607F	0	Max Profile Velocity	R/W	O	U32	-	x
0x6080	0	Max Motor Speed	R/W	O	U32	-	
0x6083	0	Profile Acceleration	R/W	O	U32	-	x
0x6084	0	Profile Deceleration	R/W	O	U32	-	x
0x60C5	0	Max Acceleration	R/W	O	U32	-	x
0x60C6	0	Max Deceleration	R/W	O	U32	-	x
0x607E	0	Polarity	R/W	O	U8	-	x
0x606B	0	Velocity Demand Value	RO	O	I16	-	x
0x606C	0	Velocity Actual Value	RO	M	I32	TPDO	x
0x606D	0	Velocity Window	R/W	O	U16	-	x
0x606E	0	Velocity Window Time	R/W	O	U16	-	x
0x606F	0	Velocity Threshold	R/W	O	U16	-	x
0x6070	0	Velocity Threshold Time	R/W	O	U16	-	x
0x3300	0	Velocity Full Scale	R/W	O	U16	-	x
0x6086	0	Motion Profile Type	R/W	O	I16	-	
0x60E0	0	Positive torque limit value	R/W	O	U16	-	x
0x60E1	0	Negative torque limit value	R/W	O	U16	-	x

In the Profile Velocity operation mode, the movement profile is defined by velocity and acceleration/deceleration commands.

To initiate a velocity-controlled profile:

- Switch the operation mode to Profile Velocity mode by writing '3' to object "Mode of Operation" (6060_h).
- Use "Controlword" (6040_h) to move in the "Operation Enable" state of Finite State Machine DSP402.
- Set acceleration in object "Profile Acceleration" (6083_h) and the deceleration in object "Profile Deceleration" (6084_h) respectively.
- Start motion by setting the target velocity in object "Target velocity" ($60FF_h$).

If needed, clear Bit 8 in object "Controlword" (6040_h) to start motion.

In this mode the Drive is able to follow a set point of speed "Target velocity" ($60FF_h$). Target velocity can be changed on-the-fly during motion. The set point is reached with the accelerations defined 6083_h and $0x6084_h$.

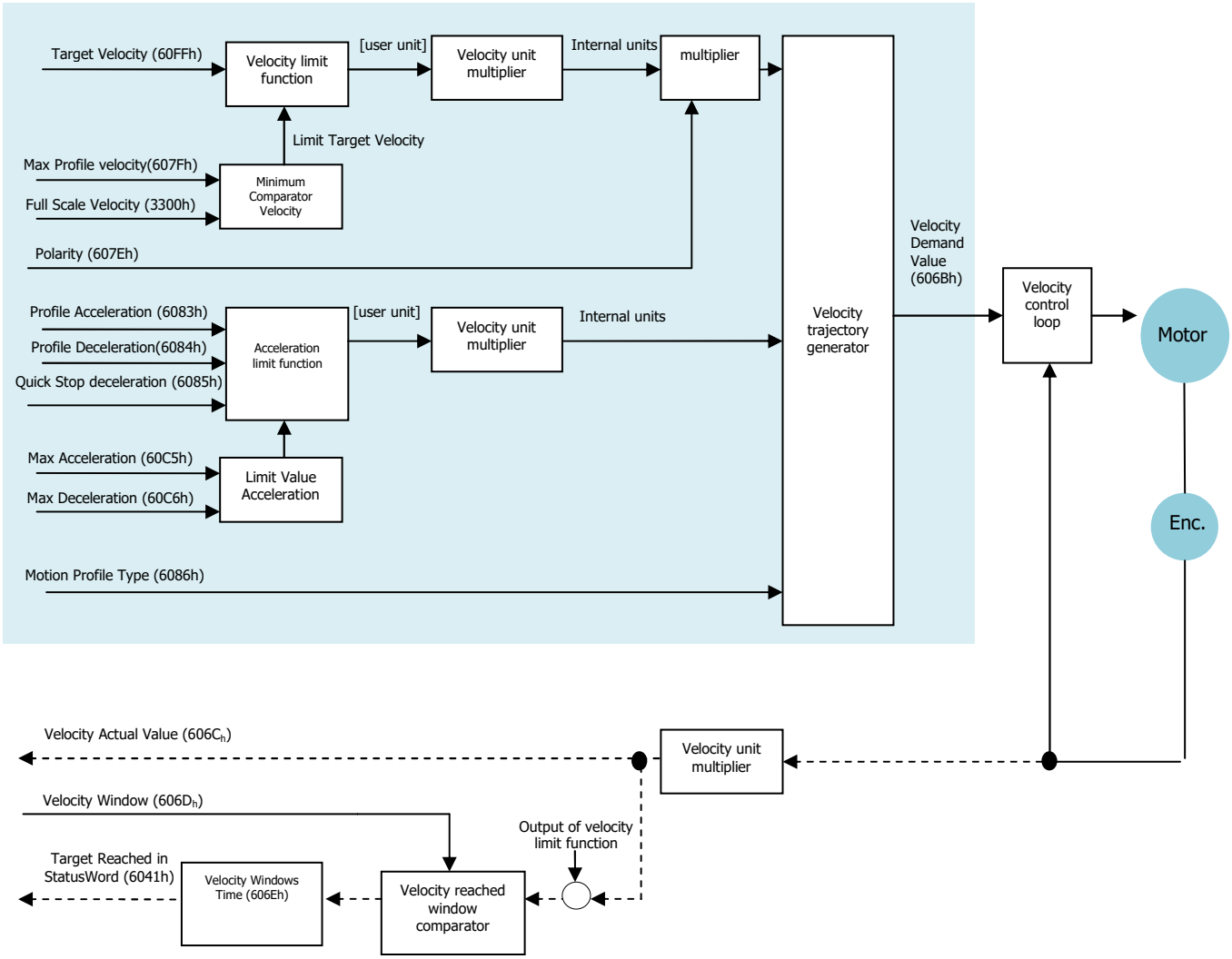


Figure 26 - Profile Velocity Block Diagram

The motion ends when one of the following conditions is met:

- "Target velocity" ($60FF_h$) is set to 0 (in this condition the motor is in torque)
- Stop caused by Halt Bit (8) of "Controlword" (6040_h).
- Stop caused by an error (the drive will move in Fault State)

  Stop to exit Operation Enabled State of DSP402 using command "Disable Operation" or "Disable Voltage" or "Quick Stop" in "Controlword" (6040_h).
- Stop caused by Safety Condition (STO input)

The result of profile Velocity is in the following bits:

- Object "Velocity actual value" ($606C_h$)
- Object "Velocity Windows " ($606D_h$)→ Target Reached Bit 10 of "Statusword" (6041_h)
- Object "Velocity Threshold" ($606F_h$) → Speed Bit 12 of "Statusword" (6041_h)

The following bits in object controlword (6040_h) have a special function:

Bit	Value	Definition
Bit 8 = Halt	0_b 1_b	The motion shall be executed or continued Axis shall be stopped according to the halt option code ($605D_h$) (*)

(*) option code $605D_h$ is not implemented

The following bits in object **6041_h** (statusword) have a special function:

Bit	Value	Definition
Bit 10 = Target Reached	0_b	If Halt (bit 8 in controlword) = 0: Target not reached If Halt (bit 8 in controlword) = 1: Axis decelerates
	1_b	If Halt (bit 8 in controlword) = 0: Target reached If Halt (bit 8 in controlword) = 1: Velocity of axis is 0
Bit 12 = Speed	0_b 1_b	Speed is notgreater than Velocity threshold Speed isgreater than Velocity threshold
Bit 13 = Max Slippage error(*)	0_b 1_b	Maximum slippage not reached Maximum slippage reached

(*) Not managed, It is used only for motor asynchronous

Caution

The torque can be limited with "Positive torque limit value" object $60E0_h$ and "Negative torque limit value" $60E1_h$

OPERATING MODE DESCRIPTION:

In the operating mode Profile Velocity, a movement is made with a desired target velocity.

Procedure:

- Set "Mode of operation" (6060_h) to operating mode Profile Velocity (value 3).
- Set "Profile acceleration" (6083_h) and "Profile deceleration" (6084_h) to the value for the acceleration ramp (user units)
- Set "Target velocity" ($60FF_h$) to the target velocity (user units)
- Set "Controlword" (6040_h) to start the operating mode.

If the power stage is enabled, the new target velocity will become active immediately and the movement will start or set in operating mode with bit halt = 0

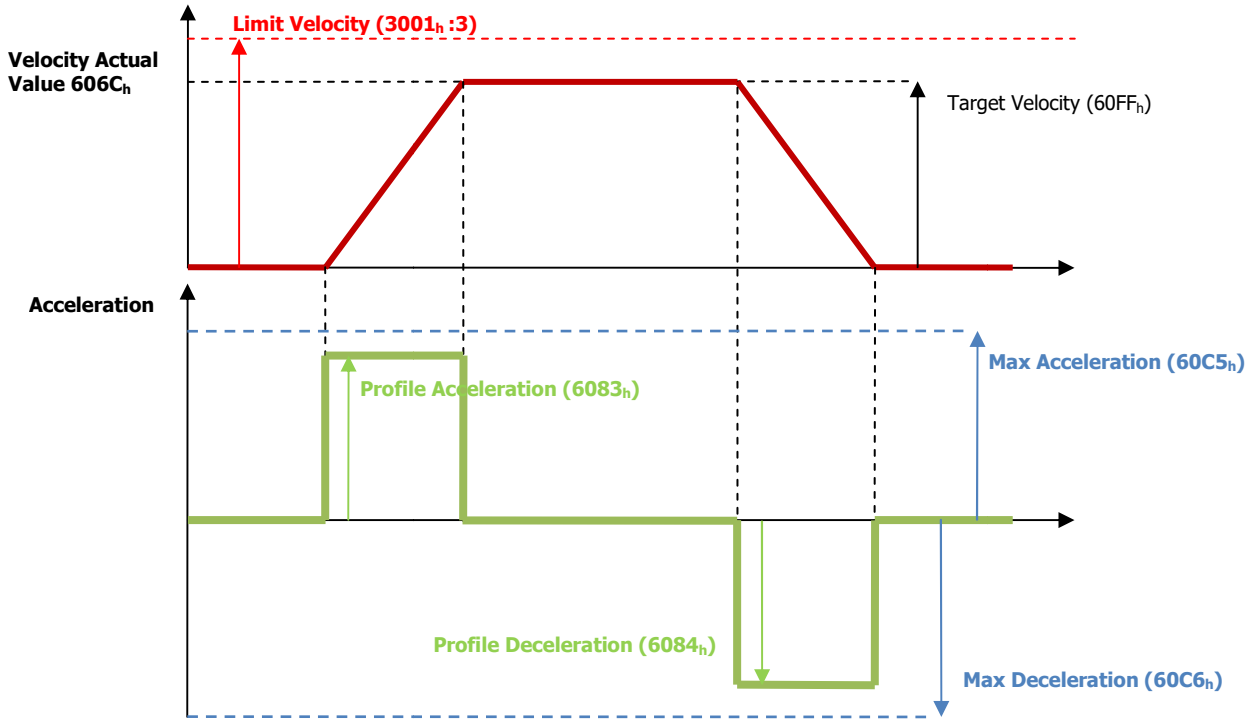


Figure 27 – Velocity Actual

Optional:

- Query "Statusword" (6041_h) to get the device status. The value is reset to zero if the operating mode is changed, the power stage is disabled or a Quick Stop is triggered.
- Query "Velocity demand" value ($606B_h$) to get the reference velocity (user units)
- Query "Velocity actual" value ($606C_h$) to get the actual velocity (user units)
- Query "Velocity window" ($606D_h$) to the value of the velocity window (customer units). It is the step to add ad Target Velocity. With the object "Velocity window" ($606D_h$) a tolerance window for the velocity actual value will be defined for comparing the "Velocity Actual" Value ($606C_h$) with the target velocity "Target velocity" ($60FF_h$). If the difference is smaller than the "Velocity window" ($606D_h$) for a longer time than specified by the object "Velocity window Time" ($606F_h$) bit 10 "Target Reached" will be set in the object "Statusword" (6041_h).

1. Stop Velocity without Halt Bit:

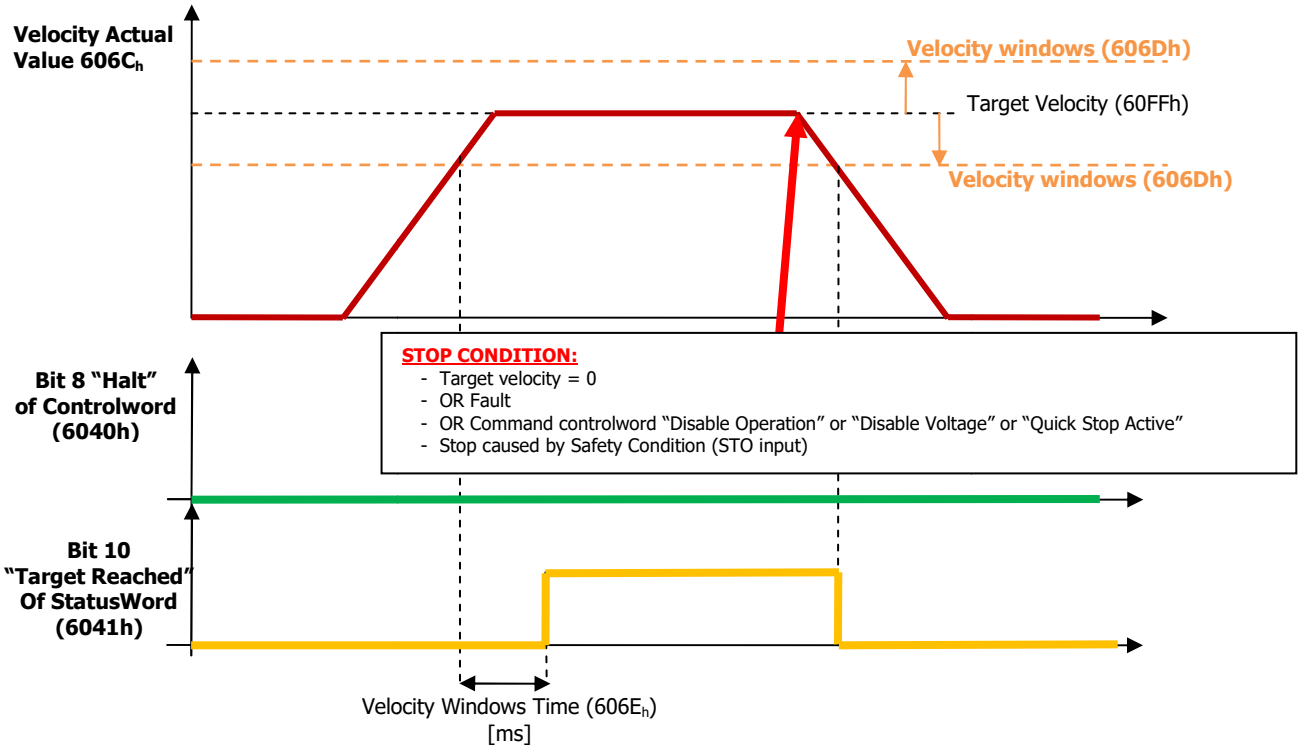


Figure 28 - Velocity Windows without Halt Bit

2. Stop Velocity with Halt Bit = 1

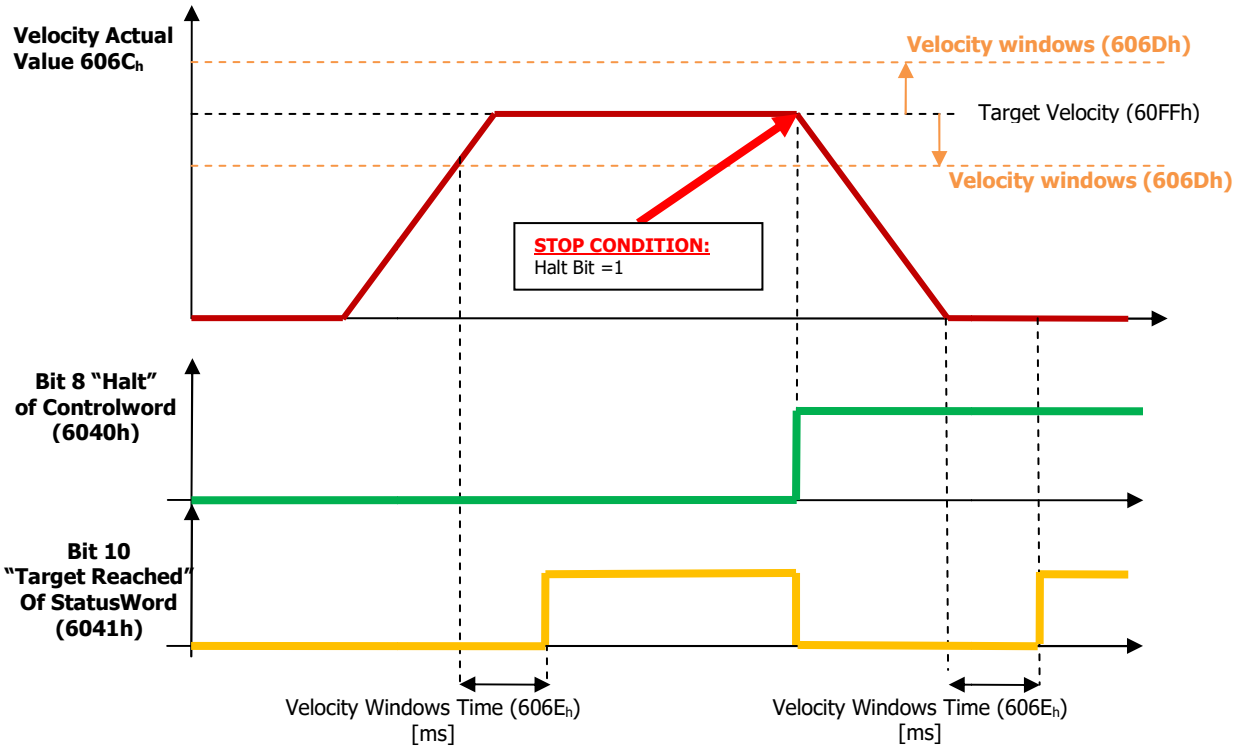


Figure 29 - Velocity Windows with Halt Bit = 1

- Query "Velocity threshold" ($606F_h$) to set the standstill window. The object "Velocity threshold" ($606F_h$) determines the velocity underneath the axis is regarded as stationary. As soon as the "Velocity Actual" Value ($606C_h$) exceeds the "Velocity threshold" ($606F_h$) longer than "Velocity threshold Time" (6070_h) the bit 12 "Speed"vis cleared in the "Statusword" (6041_h).

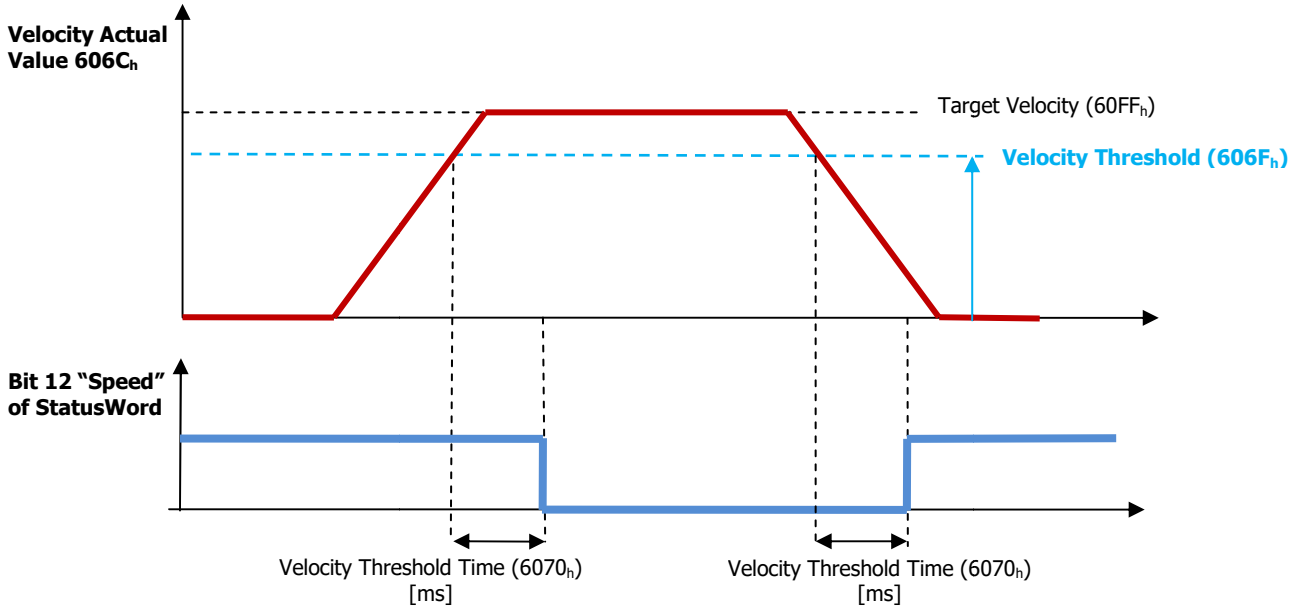


Figure 30 -Velocity threshold

Object $60FF_h$: Target Velocity

This object indicates the configured target velocity and is used as input for the trajectory generator.
Object 60FFh sets the target velocity when using profile velocity mode.
The drive then accelerates or decelerates to that velocity using the acceleration and deceleration set by objects 6083_h and 6084_h.

Object Description:

Index	Object Code	Data Type	Category
$60FF_h$	VAR	INTEGER32	Mandatory

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	YES	[-2147483647 … 2147483647]	Manufacturer Specific	[u.u.]

The drive will sent the follow abort codes:

- 0x06090031 = Value of parameter written too high, the value must be smaller than "Limit Velocity" (3001_h :3)
- 0x06040030 = the value is out of range

Caution

Minimum Target Velocity to move the motor is 0,25 rpm

Object 607F$_h$: Max Profile Velocity

This object indicates the maximal allowed velocity in either direction during a profiled motion.

Object Description:

Index	Object Code	Data Type	Category
607F$_h$	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00$_h$	rw	no	[1 ... 2147483647]	Manufacturer Specific	[u.u.]

This object is used also Profile Positioner.

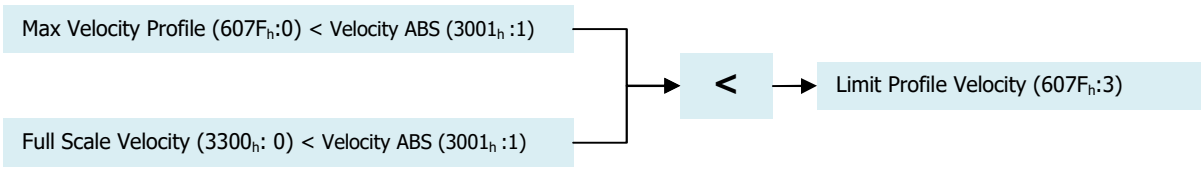
You should program the "Max Profile Velocity" to be smaller than "Velocity Absolute Maximum Rating" (3001$_h$: 1).

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B17

The "Max Profile Velocity" (607F$_h$:0), together "Full Scale Velocity (3300$_h$: 0)", defines the limit of Speed. The scheme to set the limit is the follow:



Value The drive will sent the follow abort codes:

- 0x06090031 = Value of parameter written too high, the value must be smaller than "Velocity ABS" (3001$_h$:1)
- 0x06040030 = the value is out of range

This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 607F$_h$
- Write signature "SAVE" in Store Parameters 1010$_h$ object (pay attention on the processing time)
- NMT Reset Node

Object 6083_h: Profile Acceleration

This object indicates the commanded acceleration.

- The range value admissible is [10 ... 319000] rpm/s.
- The profile deceleration must be smaller than "Max Acceleration" ($60C5_h$:0)

Object Description:

Index	Object Code	Data Type	Category
6083_h	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	no	[1 ... 2147483647]	Manufacturer Specific	[u.u.]

(*This object is used also Profile Positioner*)

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B13

The drive will sent the follow abort codes:

- 0x06090031 = Value of parameter written too high, the value must be smaller than "Max Acceleration" ($60C5_h$:0)
- 0x06040030 = the value is out of range [10 ... 319000] rpm/s

It is possible to change the Acceleration Profile in run time.
This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 6083_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 6084_h: Profile Deceleration

This object indicates the commanded deceleration.

- The range value admissible is [10 ... 319000] rpm/s.
- The profile deceleration must be smaller than "Max Deceleration" ($60C6_h$:0)

Object Description:

Index	Object Code	Data Type	Category
6084_h	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	no	[1 ... 2147483647]	Manufacturer Specific	[u.u.]

(*This object is used also Profile Positioner*)

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B14

The drive will sent the follow abort codes:

- 0x06090031 = Value of parameter written too high, the value must be smaller than "Max Deceleration" ($60C6_h$:0)
- 0x06040030 = the value is out of range [10 ... 319000] rpm/s

It is possible to change the Deceleration Profile in run time.
This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 6084_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object $60C5_h$: Max Acceleration

This object indicates the maximal acceleration. It is used to limit the acceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed.

- The range value admissible is [10 ... 319000] rpm/s.
- The Max Acceleration must be smaller than "Acceleration ABS" (3001_h :2)

Object Description:

Index	Object Code	Data Type	Category
$60C5_h$	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	no	[1 ... 2147483647]	Manufacturer Specific	[u.u.]

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B15

The drive will sent the follow abort codes:

- 0x06090031 = Value of parameter written too high, the value must be smaller than "Acceleration ABS" (3001_h :2)
- 0x06040030 = the value is out of range [10 ... 319000] rpm/s

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object $60C5_h$
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object $60C6_h$: Max Deceleration

This object indicates the maximal deceleration. It is used to limit the deceleration to an acceptable value in order to prevent the motor and the moved mechanics from being destroyed.

- The range value admissible is [10 ... 319000] rpm/s.
- The Max Deceleration must be smaller than "Acceleration ABS" (3001_h :2)

Object Description:

Index	Object Code	Data Type	Category
$60C6_h$	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	no	[1 ... 2147483647]	Manufacturer Specific	[u.u.]

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B16

The drive will sent the follow abort codes:

- 0x06090031 = Value of parameter written too high, the value must be smaller than "Acceleration ABS" (3001_h :2)
- 0x06040030 = the value is out of range [10 ... 319000] rpm/s

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object $60C6_h$
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object $606B_h$: Velocity Demand Value

This object provides the output value of the trajectory generator.

Object Description:

Index	Object Code	Data Type	Category
$606B_h$	VAR	INTEGER32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	ro	no	[-2147483648 ...2147483648]	Manufacturer Specific	[u.u.]

Object $606C_h$: Velocity Actual Value

This object provides the actual velocity value derived either from the velocity sensor or the position sensor.

Object Description:

Index	Object Code	Data Type	Category
$606C_h$	VAR	INTEGER32	Conditional: mandatory if pv or csv is supported

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	ro	YES (default)	[-2147483647 ...2147483647]	Manufacturer Specific	[u.u.]

Object $606D_h$: Velocity Window

This object indicates the velocity window.

Object Description:

Index	Object Code	Data Type	Category
$606D_h$	VAR	UNSIGNED16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	YES (default)	[1 ...65535]	Manufacturer Specific	[u.u.]

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B1B

The drive will sent the follow abort codes:

- 0x05040001 = command is invalid because the value is 0

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object $606D_h$

- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object $606E_h$: Velocity Window Time

This object indicates the velocity window time.

Object Description:

Index	Object Code	Data Type	Category
$606E_h$	VAR	UNSIGNED16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	no	[1 ...65535]	Manufacturer Specific	[ms]

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B1C.

The drive will sent the follow abort codes:

- 0x06090032 = Value of parameter written too low

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object $606E_h$
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 606Fh: Velocity Threshold

This object indicates the velocity threshold.

Object Description:

Index	Object Code	Data Type	Category
$606F_h$	VAR	UNSIGNED16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00_h	rw	no	[0 ...65535]	Manufacturer Specific	[u.u.]

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B1D

This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 606F$_h$
- Write signature "SAVE" in Store Parameters 1010$_h$ object (pay attention on the processing time)
- NMT Reset Node

Object 6070$_h$: Velocity Threshold Time

This object indicates the velocity threshold time.

Object Description:

Index	Object Code	Data Type	Category
6070h	VAR	UNSIGNED16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default Value	Unit
00h	rw	no	[1 ...65535]	Manufacturer Specific	[ms]

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B1E.

The drive will sent the follow abort codes:

- 0x06090032 = Value of parameter written too low

This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 6070$_h$
- Write signature "SAVE" in Store Parameters 1010$_h$ object (pay attention on the processing time)
- NMT Reset Node

PROFILE TORQUE MODE (4)

In the profile Torque operating mode (PT), the motor executes a movement according to a target torque or current sent by the master controller. The current regulator (torque control) is specified a current proportional to the target torque.

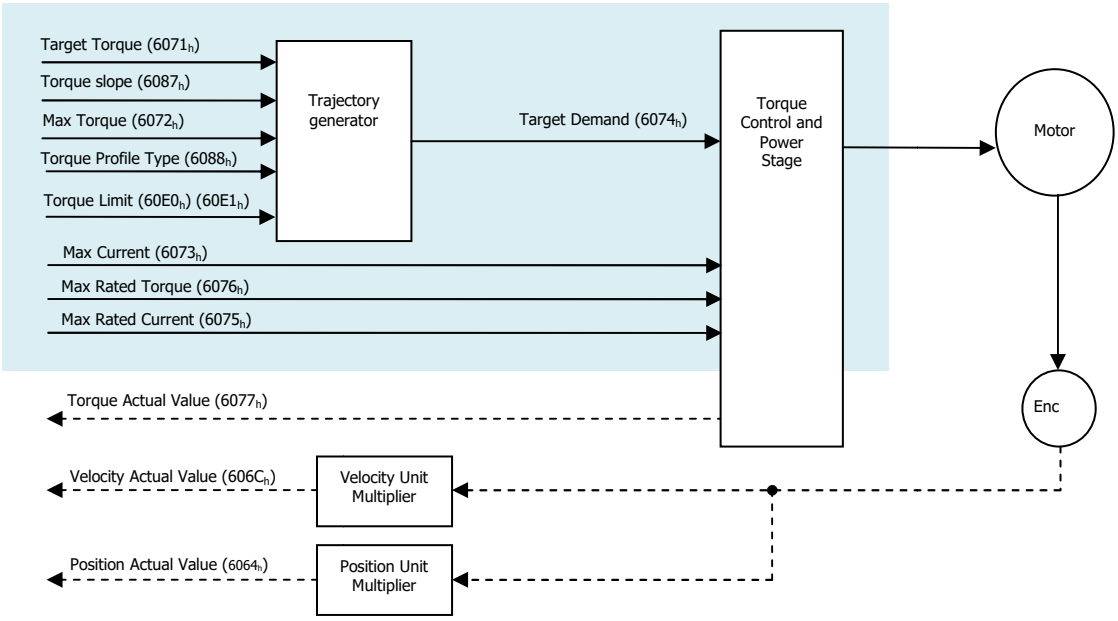


Figure 31 – Torque Profile Block Diagram

Prerequisites for the drive to be operated in Torque Profile Mode:

- The Torque Profile Mode must be set in the "Mode of Operation" (6060_h) parameter (value "4"). The mode operation can be verified using "Mode of Operation Display" (6061_h) which is updated when the current operation is accepted.
- The drive must be in *"Operation Enabled"* state of state machine of DSP402, verify it with the object "Statusword" (6041_h). To move the state machine it uses the object "controlword" (6040_h)
- Target Torque and parameters of torque must be set correctly.

The target Torque is set with object "Target Torque" (6071_h) of the object dictionary.

The objects associated to move the drive in Torque Profile Mode are the following:

Index	Sub Index	Name	READ / WRITE	M/O	Data Type	PDO	Available
0x603F	0	Error Code	RO	O	U16	-	x
0x6040	0	Control Word	R/W	M	U16	RPDO	x
0x6041	0	Status Word	RO	M	U16	TPDO	x
0x6060	0	Modes of Operation	R/W	M	I8	RPDO	x
0x6061	0	Modes of Operation Display	RO	M	I8	TPDO	x
0x6071	0	Target torque	R/W	M	I16	RPDO	x
0x6087	0	Torque slope	R/W	O	U32	-	x
0x6072	0	Max torque	R/W	O	U16	-	
0x6073	0	Max current	R/W	O	U16	-	x
0x6076	0	Motor rated torque	R/W	O	U32	-	
0x6075	0	Motor rated current	R/W	O	U32	-	x

0x6077	0	Torque actual value	RO	O	I16	TPDO	**x**
0x6078	0	Current actual value	RO	O	I16	-	**x**
0x6079	0	DC link circuit voltage	RO	O	U32	-	**x**
0x60E0	0	Positive torque limit value	R/W	O	U16	-	**x**
0x60E1	0	Negative torque limit value	R/W	O	U16	-	**x**
0x6074	0	Torque demand	RO	O	I16	-	
0x6088	0	Torque profile type	R/W	O	I16	-	**x**

The motion ends when one of the following conditions is met:

- "Target Torque" (6071_h) is set to 0 (in this condition the motor is in torque equal 0)
- Stop caused by Halt Bit (8) of "Controlword" (6040_h).
- Stop caused by an error (the drive will move in Fault State)
- Stop to exit Operation Enabled State of DSP402 using command bit "Disable Operation" or "Disable Voltage" or "Quick Stop" in "Controlword" (6040_h).
- Stop caused by Safety Condition (STO input)

The result of profile torque is in the following bits:

- Object "Torque actual value" (6077_h)
- Object "Current actual value" (6078_h)
- Target Reached Bit 10 of "Statusword" (6041_h)

The following bits in object controlword (6040_h) have a special function:

Bit	Value	Definition
Bit 8 = Halt	0_b 1_b	The motion shall be executed or continued Axis shall be stopped according to the halt option code ($605D_h$) (*)

(*) option code $605D_h$ is not implemented

The following bits in object **6041_h** (statusword) have a special function:

Bit	Value	Definition
Bit 10 = Target Reached	0_b	If Halt (bit 8 in controlword) = 0: Target not reached If Halt (bit 8 in controlword) = 1: Axis decelerates
	1_b	If Halt (bit 8 in controlword) = 0: Target reached If Halt (bit 8 in controlword) = 1: Velocity of axis is 0

Caution

The "dynamic brake controlled" is not available in Torque profile.

If the dynamic Brake feature is set (see object 3007:1) then the drive is controlled by torque (or current) without the dynamic brake controlled.

OPERATING MODE DESCRIPTION:

In the operating Torque Profile Mode a movement is made with a desired target torque.

Procedure:

- Set "Mode of operation" (6060_h) to operating mode Profile Velocity (value 4).
- Set "Motor Rated Current" (6075_h) to a value according to motor specifications (unit mArms) (this value is saved in e^2prom, follow the procedure to save the new value in e^2prom)
- Set "Torque profile Type" (6078h) to select the type of torque profile
- If the "Torque profile Type" is a Linear Ramp (Trapezoidal profile) set the rate of change of torque objrct "Torque slope" (6087_h)
- Set "Controlword" (6040_h) to activate the operating mode and enable movement. When the operating mode is started, the target torque is set to zero.
- Set "Target Torque" (6071_h) to the set point torque

If the power stage is enabled, the new target velocity will become active immediately and the movement will start or set in operating mode with bit halt = 0.

information

The torque can be limited in percent value with $60E0_h$ and $60E1_h$ object.

The type of profile can be set by "Torque Profile Type" (6088_h):

- Value "**-1**": Immediately
- Value "**0**": Linear Ramp.

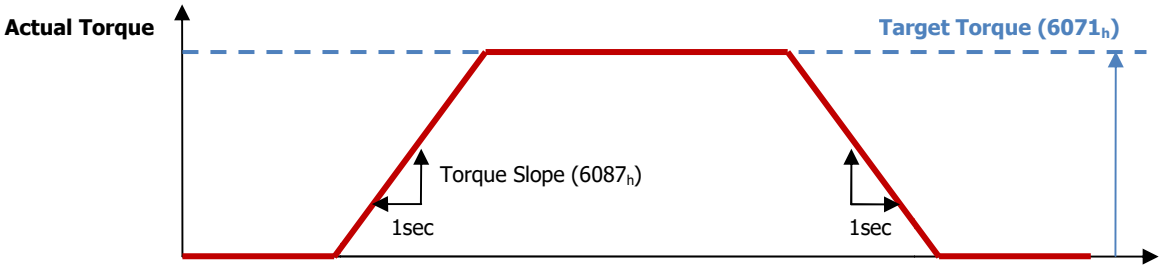


Figure 32 – Diagram Torque Trapezoidal Type

- Value "**1**": Sin^2 Ramp ***(Not Available)***

Optional:

- Query "Statusword" (6041_h) to get the device status. The value is reset to zero if the operating mode is changed, the power stage is disabled or a Quick Stop is triggered.
- Query "Torque Actual Value" value (6077_h) to get the reference instantaneous current in the drive motor.
- Query "Current Actual Value" value (6078_h) to get the reference filtered current in the drive motor.

- Query "Target Reached" value (bit 10) of object "Statusword" (6041_h).

1. Torque Profile Mode without Halt Bit:

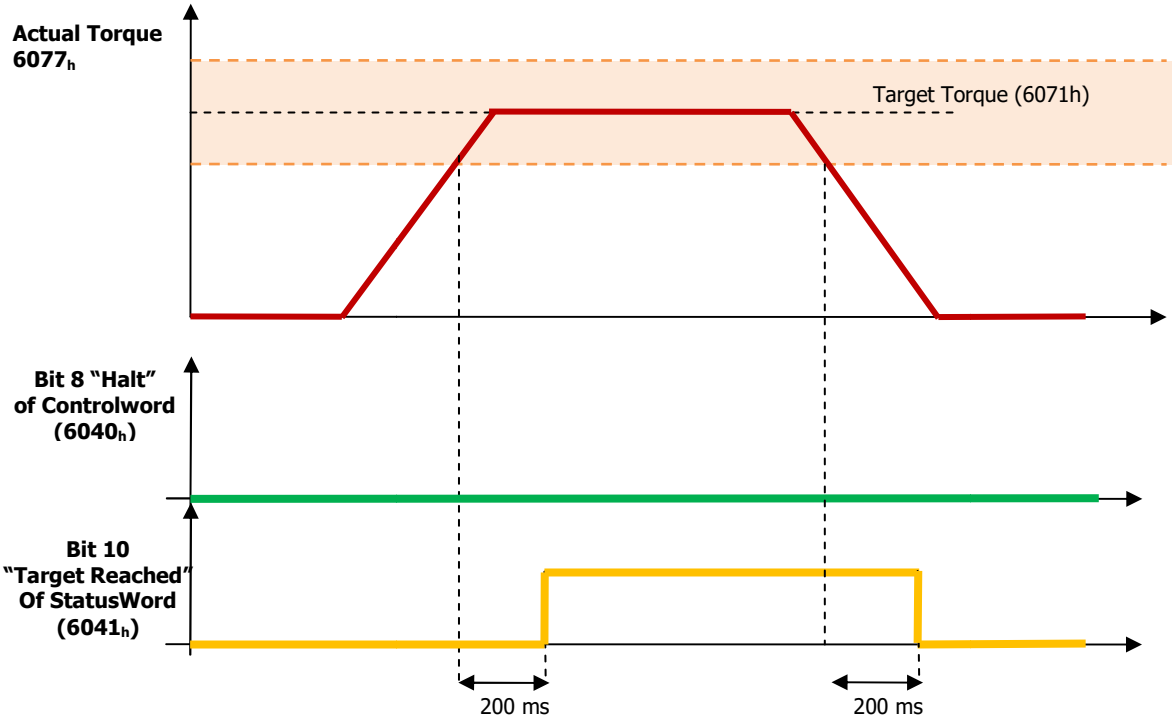


Figure 33 – Torque Reached Bit without Halt Bit

2. Torque Profile Mode with Halt Bit = 1

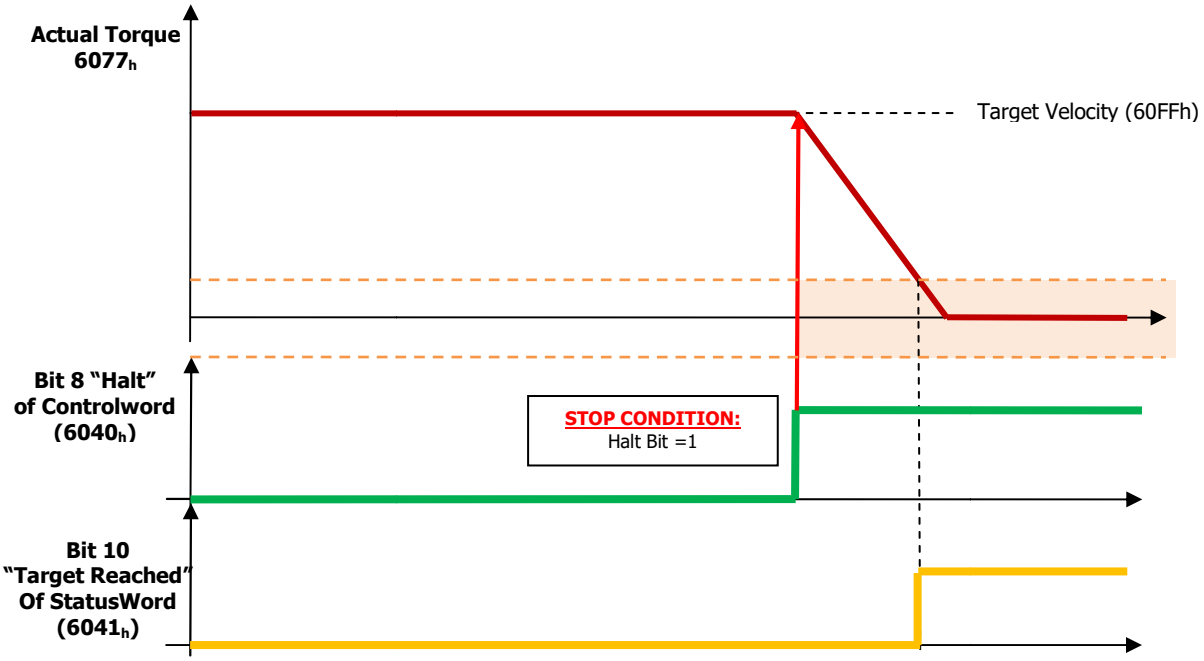


Figure 34 - Velocity Windows with Halt Bit = 1

Object 6071_h - Target Torque

This object shall indicate the configured input value for the torque controller in profile torque mode. The value shall be given per thousand of rated current.
Target Torque is the torque set-point, which is given here as the torque producing current I_q.

Object Description:

Index	Object Code	Data Type	Category
6071_h	VAR	INTEGER16	Mandatory

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	rw	yes	[1 ...65535]	Manuf Specific	[1000/Rated Current]

Example:
If a torque that is relative to current of 2Arms is needed and the object "Motor Rated Current" (6075_h) is 12500 mArms, then:

$$TargetTorque[6071h] = \frac{2000\,mArms\ *\ 1000}{12500\,mArms} = 160$$

This number means 16.0% of Motor Rated Current

The drive will send the follow abort code:

- 0x06090031 = Value of parameter written too high

Object 6075_h – Motor Rated Current

This object shall indicate the configured motor rated current. It is taken from the motor's name-plate. Depending on the motor and drive technology, this current is DC, peak or r.m.s. (root-mean-square) current. All relative current data refers to this value. The value shall be given in mArms.

Object Description:

Index	Object Code	Data Type	Category
6075_h	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	rw	no	[1 ... 2147483647]	Manuf. Specific	[mArms]

The "Motor Rated Current" (6075_h: 0) must be lower or equal than Rated Current of Motor, it is defined in the object "Current Parameters" (3003_h) sub-index 3 (named "Rated Current Motor") or by Motor Parameters Datasheet.

Motor Rated Current (6075_h: 0)	≤	Nominal Rated Current (3003_h: 3) (Datasheet Parameter)

Example:
If Nominal Rated Current (3003_h: 3) is 12500 mArms then the "Motor Rated Current" (6075_h: 0) must be ≤ 12500 mArms.

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06090031 = Value of parameter written too high (because it must be smaller than 0x3003: 3)

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B1F

This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 6075_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 6073_h – Max Current

This object shall indicate the configured maximum permissible torque creating current in the motor. The value shall be given per thousand of rated current.

Object Description:

Index	Object Code	Data Type	Category
6073_h	VAR	UNSIGNED16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	rw	no	[1 ... 32767]	Manuf. Specific	[1000/Rated Current]

The value "Max Current" (6073_h :0) converted in mArms must be lower or equal than Peak Current [mArms]. The Peak current is defined in the object "Current Parameters" (3003_h) sub-index 2 (named "Peak Current Motor") or by Motor Parameters Datasheet.

Max Current (6073_h: 0) → [mArms] ≤ Peak Current (3003_h: 2) - Datasheet Parameter

Example:
If Peak Rated Current (3003_h: 3) is 41000 mArms and the Motor Rated Current is 12500 mArms then the "Max Current" (6073_h: 0) must be:

$$Max\ Current[6073h] = \frac{41000\ mArms\ *\ 1000}{12500\ mArms} = 3280$$

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)
- 0x06090031 = Value of parameter written too high (the value converted must be smaller than 0x3003:2)

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B22

This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 6073_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 6087_h - Torque slope

This object shall indicate the configured rate of change of torque. The value shall be given in units of per thousand of rated torque per second.

Object Description:

Index	Object Code	Data Type	Category
6087_h	VAR	UNSIGNED32	Mandatory IF

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	rw	no	[1 ... 2147483647]	Manuf. Specific	[(1000/Rated Current)/s]

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)

It is possible to change the Deceleration Profile in run time.
This object can be changed and saved in e^{2}prom memory

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B25

This object can be changed and saved in e^{2}prom memory

E^{2}prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 6087_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 6088_h - Torque profile type

This object shall indicate the configured type of profile used to perform a torque change.

Object Description:

Index	Object Code	Data Type	Category
6088_h	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	rw	no	[-32767 ...32767]	Manuf. Specific	-

Value definition

Value	Definition
-1	Immediately
0	Linear ramp (trapezoidal profile) (default)
1	sin2 ramp ***(not available)***

The drive will sent the follow abort codes:

- 0x08000024 = No data available (if the value is the type not available)

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B18

This object can be changed and saved in e^2prom memory

E^2prom Store

- The drive mustn't be in "Operational enabled" or "Quick Stop Active"
- Write the new value in SDO object 6088_h
- Write signature "SAVE" in Store Parameters 1010_h object (pay attention on the processing time)
- NMT Reset Node

Object 6074_h - Torque demand

This object shall provide the output value of the trajectory generator. The value shall be given per thousand of rated current.

Object Description:

Index	Object Code	Data Type	Category
6074_h	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	ro	no	[-32767 ...32767]	Manuf. Specific	[1000/ Rated Current]

Object 6077_h - Torque Actual Value

This object shall provide the actual value of the torque. It shall correspond to the instantaneous torque in the motor. The value shall be given per thousand of rated current.

Object Description:

Index	Object Code	Data Type	Category
6077_h	VAR	INTEGER16	Mandatory

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	ro	si	[-32767 ...32767]	Manuf. Specific	[1000/Rated Current]

Object 6078_h - Torque Actual Current

This object shall provide the actual value of the current. It shall correspond to the current in the motor. The value shall be given per thousand of rated current. This value is filtered.

Object Description:

Index	Object Code	Data Type	Category
6078_h	VAR	INTEGER16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	ro	no	[-32767 ...32767]	Manuf. Specific	[1000/Rated Current]

Object 6079_h – DC Link circuit Voltage

This object shall provide the instantaneous DC link current voltage at the drive device. The value shall be given in mV.

Object Description:

Index	Object Code	Data Type	Category
6079_h	VAR	UNSIGNED32	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	ro	no	[0 ... 4294967296]	Manuf. Specific	[mV]

Object $60E0_h$ – Positive torque limit value

This object shall indicate the configured maximum positive torque in the motor. The value shall be given percent of rated current. Positive torque takes effect in the case of motive operation is positive velocity or regenerative operation is negative velocity.

Object Description:

Index	Object Code	Data Type	Category
$60E0_h$	VAR	UNSIGNED16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	ro	no	[0 ... 100]	100	[%]

This object can be used also with Velocity Profile.

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B23

This object can't be saved in e2prom memory but it can be changed in RAM

Object $60E1_h$ – Negative torque limit value

This object shall indicate the configured maximum negative torque in the motor. The value shall be given percent of rated current. Negative torque takes effect in the case of motive operation is negative velocity or regenerative operation is positive velocity.

Object Description:

Index	Object Code	Data Type	Category
$60E1_h$	VAR	UNSIGNED16	Optional

Entry Description:

Sub-Index	Access	PDO mapping	Value Range	Default	Unit
00_h	ro	no	[0 ... 100]	100	[%]

This object can be used also with Velocity Profile.

The drive will sent the follow abort codes:

- 0x06040030 = the value is out of range (see table Entry Description)

Caution

This object is controlled by some limits of drive. If the value is not correct the drive sends an Abort Code.

If the value is not correct during initialization the drive sends an error messages with Emergency Protocol. See Error Code 0x8B24

This object can't be saved in e2prom memory but it can be changed in RAM

PROFILE HOMING MODE (6) (not available)

ANALOG MODE

In this operation mode the Drive can be piloted with an analog reference -10 Volt +10 Volt. Giving voltage on +VREF and –VREF it is possible to supply to the drive the speed set point.

The motor speed will depend from the reference voltage given on +VREF and –VREF and from the maximum speed available on the drive.
The Speed Set Point is proportional to the voltage supplied on the concerned input.
Giving a +VREF voltage the motor will set to the maximum speed in clockwise rotation, motor front, while giving a -VREF the motor will set to the maximum speed in counter clockwise rotation, motor front.

Example:
Maximum Speed Configured =Max Speed r.p.m.

Input Voltage = +10V →Rotation speed (clockwise) = + Max Speed r.p.m.
Input Voltage = -10V →Rotation speed (counter clockwise) = - Max Speed r.p.m
Input Voltage = +5V →Rotation speed (clockwise) = + 1/2 Max Speed r.p.m
Input Voltage = -5V →Rotation speed (counter clockwise) = - 1/2 Max Speed r.p.m

To move the motor is necessary that the digital Inputs configured as "RUN" and "STOP". See Digital I/O chapter what is the digital Input configuration.

Variable Monitoring

In analog Mode it's possible to monitor a list of variables on drive by CANOpen, connecting a CAN Interface and using Lafert Drive SW:

- Object 2002h: Drive Status Mode
- Object 2003h: Warning
- Object 2004h: State Lafert Servo Drive Machine
- Object 2030h: Temperature Drive
- Object 2031h: Temperature Motor
- Object 2032h: Temperature Heat Sink
- Object 2041h: Voltage Bus
- Object 2050h: Torque Current
- Object 2051h: Power Drive
- Object 2052h: Power Motor
- Object 2053h: Velocity Filtered
- Object 3020h: Function Digital Input
- Object 3021h: Digital Input 1
- Object 3022h: Digital Input 2
- Object 3023h: Digital Input 3
- Object 3024h: Digital Input 4
- Object 4000h: Safety State

7. | CANOPEN OBJECT LIST

INDEX	SUB.	DESCRIPTION	CODE	TYPE	O/M	ATTR.	ARGUMENT
		STANDARD OBJECTS DS301					
1000 h	0	Device Type	COST	UINT32	M	RO	Settings
1001 h	0	Error Register	VAR	UINT32	O	RO	Settings
1002 h	0	Manufacturer Status Register	VAR	UINT32	O	RO	Settings
1003 h	0	Pre-Defined Error Field	ARRAY	UINT32	M	RO	Alarm
	1	History Error Field		UINT32	M	RO	Alarm
	2	History Error Field		UINT32	O	RO	Alarm
	3	History Error Field		UINT32	O	RO	Alarm
	4	History Error Field		UINT32	O	RO	Alarm
	5	History Error Field		UINT32	O	RO	Alarm
	6	History Error Field		UINT32	O	RO	Alarm
	7	History Error Field		UINT32	O	RO	Alarm
	8	History Error Field		UINT32	O	RO	Alarm
	9	History Error Field		UINT32	O	RO	Alarm
	10	History Error Field		UINT32	O	RO	Alarm
	11	History Error Field		UINT32	O	RO	Alarm
	12	History Error Field		UINT32	O	RO	Alarm
	13	History Error Field		UINT32	O	RO	Alarm
	14	History Error Field		UINT32	O	RO	Alarm
	15	History Error Field		UINT32	O	RO	Alarm
1005 h	0	Cob-ID Sync	VAR	UINT32		R/W	Settings
1008 h	0	Manufacturer Device Name	VAR	STRING	M	RO	Communication
1009 h	0	Manufacturer Hardware Version	VAR	STRING	M	RO	Communication
100A h	0	Manufacturer Software Version	VAR	STRING	M	RO	Communication
100C h	0	Guard Time	VAR	UINT16	O	R/W	Settings
100D h	0	Life Time Factor	VAR	UINT8	O	R/W	Settings
1010 h	0	Store Parameter Fields	ARRAY	UINT32	O	R/W	Memory Parameters
	1	Save all Parameters			M	R/W	Memory Parameters
	2	Save Communication Parameters			O	R/W	Memory Parameters
	3	Save Application Parameters			O	R/W	Memory Parameters
	4	Save Manufacturer Parameters			O	R/W	Memory Parameters
	5	Save Data Factory Parameters			O	R/W	(reserved)
1011 h	0	Restore Default Parameter	ARRAY	UINT32	O	R/W	Memory Parameters
	1	Restore all Default Parameters			O	R/W	Memory Parameters
	2	Restore Communication Default Parameters			O	R/W	***not available***
	3	Restore Application Default Parameters			O	R/W	***not available***
	4	Restore Manufacturer Default Parameters			O	R/W	***not available***
	5	Restore Data Factory Parameters			O	R/W	(reserved)
1014 h	0	Cob-ID Emergency Message	VAR	UINT32	O	RO	***not available***

1017 h	0	Producer HeartBeat Time	VAR	UINT16	M	R/W	Settings
1018 h	0	Identity Object	RECORD	UINT32	M	RO	-
	1	Vendor Id			M	RO	Settings
	2	Product Code			O	RO	***not available***
	3	Revision number			O	RO	***not available***
	4	Serial number			O	RO	***not available***
1029 h	0	Error Behaviour	ARRAY	UINT8	O	RO	***not available***
	1	Communication Error			O	R/W	***not available***
1200 h	0	Server SDO Parameter 1			O	R/W	Settings
1280 h	0	Client SDO Parameter 1			O	R/W	Settings
1400 h	0	Receive PDO Communication Parameter 1	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1401 h	0	Receive PDO Communication Parameter 2	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1402 h	0	Receive PDO Communication Parameter 3	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1403 h	0	Receive PDO Communication Parameter 4	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1600 h	0	Receive PDO Mapping Parameter 1	RECORD	UINT8	M	R/W	Settings
	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings
1601 h	0	Receive PDO Mapping Parameter 2	RECORD	UINT32	M	R/W	Settings
	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings

1602 h	0	Receive PDO Mapping Parameter 3	RECORD	UINT32	M	R/W	Settings
	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings
1603 h	0	Receive PDO Mapping Parameter 4	RECORD	UINT8	M	R/W	Settings
	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings
1800 h	0	Transmit PDO Communication Parameter 1	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1801 h	0	Transmit PDO Communication Parameter 2	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1802 h	0	Transmit PDO Communication Parameter 3	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1803 h	0	Transmit PDO Communication Parameter 4	RECORD	UINT8	M	R/W	Settings
	1	COB-ID		UINT32	M	R/W	Settings
	2	Transmission Type		UINT8	M	R/W	Settings
	3	Inhibit Time		UINT16	O	R/W	Settings
1A00 h	0	Transmit PDO Mapping Parameter 1	RECORD	UINT8	M	R/W	Settings
	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings
1A01 h	0	Transmit PDO Mapping Parameter 2	RECORD	UINT8	M	R/W	Settings

	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings
1A02 h	0	Transmit PDO Mapping Parameter 3	RECORD	UINT8	M	R/W	Settings
	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings
1A03 h	0	Transmit PDO Mapping Parameter 4		UINT32	M	R/W	Settings
	1	Mapping Entry 1		UINT32	M	R/W	Settings
	2	Mapping Entry 2		UINT32	M	R/W	Settings
	3	Mapping Entry 3		UINT32	M	R/W	Settings
	4	Mapping Entry 4		UINT32	M	R/W	Settings
	5	Mapping Entry 5		UINT32	M	R/W	Settings
	6	Mapping Entry 6		UINT32	M	R/W	Settings
	7	Mapping Entry 7		UINT32	M	R/W	Settings
	8	Mapping Entry 8		UINT32	M	R/W	Settings
MANUFACTURER OBJECT							
2000 h	0	ID Node	VAR	UINT8	M	R/W	*Settings*
2001 h	0	CAN Baud Rate	VAR	UINT16	M	R/W	*Settings*
2002 h	0	Drive Status	VAR	INT16	O	RO	*TELL*
2003 h	0	Warning	VAR	UINT32	O	RO	*TELL*
2004 h	0	State Lafert Servo Drive Machine	VAR	INT16	O	RO	*TELL*
2030 h	0	Drive Temperature	VAR	INT16	O	RO	*TELL*
2031 h	0	Motor Temperature	VAR	INT16	O	RO	*TELL*
2032 h	0	Heat Sink Temperature	VAR	INT16	O	RO	*TELL*
2041 h	0	Voltage Bus	VAR	INT16	O	RO	*TELL*
2050 h	0	Torque Current	VAR	INT16	O	RO	*TELL*
2051 h	0	Drive Power	VAR	INT16	O	RO	***not available***
2052 h	0	Motor Power	VAR	INT16	O	RO	***not available***
2053 h	0	Velocity Filtered	VAR	INT16	O	RO	*TELL*
3001 h	0	**Limits Parameter**	ARRAY	UINT32	O	**RO**	-
	1	Velocity ABS		UINT32	O	RO	*TELL*
	2	Acceleration ABS		UINT32	O	RO	*Settings*
	3	Limit Velocity Profile		UINT32	O	RO	*TELL*

3002 h	0	**Brake Parameters**	ARRAY	INT16	M IF	RO	*Settings*
	1	Motor Brake Option		INT16	M IF	R/W	*Settings*
	2	Motor Brake Delay		INT16	M IF	R/W	*Settings*
	3	Brake Unlock time		INT16	M IF	R/W	*Settings*
	4	Brake Timeout		INT16	M IF	R/W	*Settings*
	5	Automatic/Manual Mode Configuration		INT16	M IF	R/W	*Settings*
	6	Motor Brake Status		INT16	M IF	RO	*TELL*
3003 h	0	**Drive Size Parameters**	ARRAY	INT16	O	RO	-
	1	Maximum Current		INT16	O	RO	*TELL*
	2	Peak Current		INT16	O	RO	*TELL*
	3	Rated Current		INT16	O	RO	*TELL*
	4	I2T		INT16	O	RO	*TELL*
	5	Maximum Peak Current		INT16	O	RO	*TELL*
	6	Maximum Rated Current		INT16	O	RO	*TELL*
	7	Maximum I2T		INT16	O	RO	*TELL*
3004 h	**0**	**FeedBack Parameters**	ARRAY	INT16	O	**RO**	*TELL*
	1	Feedback Type		INT16	O	RO	*TELL*
	2	Resolution		INT16	O	RO	*TELL*
3005 h	0	Filter Parameters	ARRAY	INT16	O	RO	***not available***
3006 h	0	**Motor Specific Settings**	ARRAY	INT16	O	RO	*TELL*
	1	Motor Part Number		INT16	O	RO	*TELL*
	2	Max Motor Speed		INT16	O	RO	*TELL*
	3	N Poli		INT16	O	RO	*TELL*
3007 h	0	**Dynamic Brake Parameter**	ARRAY	INT16	M IF	RO	*Settings*
	1	Dynamic Brake Option		INT16	M IF	R/W	*Settings*
	2	Holding Torque Time		INT16	M IF	R/W	*Settings*
	3	Dynamic Brake Status		INT16	M IF	RO	*TELL*
	4	Decrement step ramp		INT16	M IF	R/W	*Settings*
3008 h	0	Emergency Enable Parameter	ARRAY	INT16	M IF	RO	*Settings*
	1	Emergency Enable Option		INT16	M IF	R/W	*Settings*
	2	Emergency Input Neg		INT16	M IF	R/W	*Settings*
	3	**Emergency Status**		INT16	M IF	RO	*TELL*
3010 h	0	Alarm Option	ARRAY	INT16	O	RO	***not available***
3020 h	0	**Function Digital Input**	ARRAY	INT16	O	RO	*TELL*
	1	Configuration		INT16	O	RO	*TELL*
	2	State		INT16	O	RO	*TELL*
	3	Level		INT16	O	RO	*TELL*
	4	-		INT16	O	RO	*TELL*
	5	-		INT16	O	RO	*TELL*
3021 h	0	**Digital Input 1**	ARRAY	INT16	O	RO	*TELL*
	1	Configuration		INT16	O	RO	*TELL*
	2	State		INT16	O	RO	*TELL*
	3	Level		INT16	O	RO	*TELL*
	4	-		INT16	O	RO	*TELL*

	5	-		INT16	O	RO	*TELL*
3022 h	0	**Digital Input 2**	ARRAY	INT16	O	RO	*TELL*
	1	Configuration		INT16	O	RO	*TELL*
	2	State		INT16	O	RO	*TELL*
	3	Level		INT16	O	RO	*TELL*
	4	-		INT16	O	RO	*TELL*
	5	-		INT16	O	RO	*TELL*
3023 h	0	**Digital Input 2**	ARRAY	INT16	O	RO	*TELL*
	1	Configuration		INT16	O	RO	*TELL*
	2	State		INT16	O	RO	*TELL*
	3	Level		INT16	O	RO	*TELL*
	4	-		INT16	O	RO	*TELL*
	5	-		INT16	O	RO	*TELL*
3024 h	0	**Digital Input 2**	ARRAY	INT16	O	RO	*TELL*
	1	Configuration		INT16	O	RO	*TELL*
	2	State		INT16	O	RO	*TELL*
	3	Level		INT16	O	RO	*TELL*
	4			INT16	O	RO	*TELL*
	5			INT16	O	RO	*TELL*
3030 h	0	Drive Digital Output		INT16	O	RO	*TELL*
3040 h	0	Analog Input		INT16	O	RO	*TELL*
3050 h	0	Analog Output 1		INT16	O	RO	*TELL*
3051 h	0	Analog Output 2		INT16	O	RO	*TELL*
3200 h	0	Current PID	ARRAY	INT16	M	R/W	*Settings*
	1	PidCur Kp		INT16	M	R/W	*Settings*
	2	PidCur Ki		INT16	M	R/W	*Settings*
	3	PidCur Kv		INT16	M	R/W	*Settings*
	4	PidCur Kd		INT16	M	R/W	*Settings*
	5	PidCur N		INT16	M	R/W	*Settings*
	6	PidCur FF		INT16	M	R/W	*Settings*
3201 h	0	Speed PID	ARRAY	INT16	M	R/W	*Settings*
	1	PidVel Kp		INT16	M	R/W	*Settings*
	2	PidVel Ki		INT16	M	R/W	*Settings*
	3	PidVel Kv		INT16	M	R/W	*Settings*
	4	PidVel Kd		INT16	M	R/W	*Settings*
	5	PidVel N		INT16	M	R/W	*Settings*
	6	PidVel FF		INT16	M	R/W	*Settings*
3202 h	0	Position PID	ARRAY	INT16	M	R/W	*Settings*
	1	PidPos Kp		INT16	M	R/W	*Settings*
	2	PidPos Ki		INT16	M	R/W	*Settings*
	3	PidPos Kv		INT16	M	R/W	*Settings*
	4	PidPos FF Ra V		INT16	M	R/W	*Settings*
	5	PidPos FF Ra A		INT16	M	R/W	*Settings*
	6	PidPos FF Vr V		INT16	M	R/W	*Settings*

	7	PidPos FF Rd A		INT16	M	R/W	*Settings*
	8	PidPos FF Rd V		INT16	M	R/W	*Settings*
	9	PidPos Tc		INT16	M	R/W	*Settings*
3300 h	0	Velocity Full Scale		UINT16	O	R/W	*Settings*
4500 h	0	Safety Feature		UINT16	O	RO	*TELL*
	1	Safety State		UINT16	O	RO	*TELL*
	2	STO Function		UINT16	O	RO	*TELL*
4501 h	0	Dummy	ARRAY	INT16	O	RO	***not available***
4502 h	0	DummyTell	ARRAY	INT16	O	RO	***not available***
4503 h	0	DummyTellLong	ARRAY	INT16	O	RO	***not available***
4503 h	0	DummyCANopen	ARRAY	INT16	O	RO	***not available***
STANDARD OBJECTS DSP402							
6007 h	0	Abort Connection Option Code	VAR	UINT16	O	R/W	***not available***
603F h	0	Error Code	VAR	UINT16	O	RO	Alarm
6040 h	0	Control Word	VAR	UINT16	M	R/W	State Machine DS402
6041 h	0	Status Word	VAR	UINT16	M	RO	State Machine DS402
605A h	0	Quick Stop Option Code	VAR	INT16	O	R/W	***not available***
605B h	0	Shutdown Option Code	VAR	INT16	O	R/W	***not available***
605C h	0	Disable Option Code	VAR	INT16	O	R/W	***not available***
605D h	0	Halt Option Code	VAR	INT16	O	R/W	***not available***
605E h	0	Fault Reaction Code	VAR	INT16	O	R/W	***not available***
6060 h	0	Modes of Operation	VAR	INT8	M	R/W	State Machine DS402
6061 h	0	Modes of Operation Display	VAR	INT8	M	RO	State Machine DS402
6062 h	0	Position Demand Value	VAR	INT32	O	RO	***not available***
6063 h	0	Position Actual internal Value	VAR	INT32	O	RO	***not available***
6064 h	0	Position Actual Value	VAR	INT32	M	RO	***not available***
6065 h	0	Following Error Windows	VAR	UINT32	O	R/W	***not available***
6066 h	0	Following Error TimeOut	VAR	UINT16	O	R/W	***not available***
6067 h	0	Position Windows	VAR	UNIT32	O	R/W	***not available***
6068 h	0	Position Window Time	VAR	UINT16	O	R/W	***not available***
6069 h	0	Velocity Sensor Actual Value	VAR	INT32	O	RO	***not available***
606A h	0	Sensor Selection Code	VAR	INT16	O	R/W	***not available***
606B h	0	Velocity Demand Value	VAR	INT32	O	RO	Profile Velocity
606C h	0	Velocity Actual Value	VAR	INT32	M	RO	Profile Velocity
606D h	0	Velocity Window	VAR	UINT16	O	R/W	Profile Velocity
606E h	0	Velocity Window Time	VAR	UINT16	O	R/W	Profile Velocity
606F h	0	Velocity Threshold	VAR	UINT16	O	R/W	Profile Velocity
6070 h	0	Velocity Threshold Time	VAR	UINT16	O	R/W	Profile Velocity
6071 h	0	Target Torque	VAR	INT16	M	R/W	Torque Profile
6072 h	0	Max Torque	VAR	UINT16	O	R/W	***not available***
6073 h	0	Max Current	VAR	UINT16	O	R/W	Torque Profile
6074 h	0	Torque Demand	VAR	INT16	O	RO	Torque Profile
6075 h	0	Motor Rated Current	VAR	UINT32	O	R/W	Torque Profile
6076 h	0	Motor Rated Torque	VAR	UINT32	O	R/W	***not available***

6077 h	0	Torque Actual Value	VAR	INT16	O	RO	Torque Profile
6078 h	0	Current Actual Value	VAR	INT16	O	RO	Torque Profile
6079 h	0	DC Link Circuit Voltage	VAR	UINT32	O	RO	Tell
607A h	0	Target Position	VAR	INT32	M	R/W	***not available***
607B h	0	Position Range Limit	VAR	INT32	O	R/W	***not available***
607C h	0	Home Offset	VAR	INT32	O	R/W	***not available***
607D h	0	Software Position Limit	VAR	INT32	O	R/W	***not available***
607E h	0	Polarity	VAR	UINT8	O	R/W	Profile Velocity e Position
607F h	0	Max Profile Velocity	VAR	UINT32	O	R/W	Profile Velocity e Position
6080 h	0	Max Motor Speed	VAR	UINT32	O	R/W	***not available***
6081 h	0	Profile Velocity	VAR	UINT32	M	R/W	***not available***
6082 h	0	End Velocity	VAR	UINT32	O	R/W	***not available***
6083 h	0	Profile Acceleration	VAR	UINT32	O	R/W	Profile Velocity e Position
6084 h	0	Profile Deceleration	VAR	UINT32	O	R/W	Profile Velocity e Position
6085 h	0	Quick Stop Deceleration	VAR	UINT32	O	R/W	***not available***
6086 h	0	Motion Profile Type	VAR	INT16	O	R/W	***not available***
6087 h	0	Torque Slope	VAR	UINT32	M	R/W	Torque Profile
6088 h	0	Torque Profile Type	VAR	INT16	O	R/W	Torque Profile
608F h	0	Position Encoder Resolution	VAR	VAR	O	R/W	***not available***
6090 h	0	Velocity Encoder Resolution	VAR	VAR	O	R/W	***not available***
6091 h	0	Gear Ratio	VAR	UINT32	O	R/W	***not available***
6092 h	0	Feed Constant	VAR	UINT32	O	R/W	***not available***
6096 h	0	Velocity Factor Group	VAR	UINT32	O	R/W	Settings
	1	Num Velocity Factor		UINT32	O	R/W	Settings
	2	Div Velocity Factor		UINT32	O	R/W	Settings
6097 h	0	Acceleration Factor Group	VAR	UINT32	O	R/W	Settings
	1	Num Acceleration Factor		UINT32	O	R/W	Settings
	2	Div Acceleration Factor		UINT32	O	R/W	Settings
6098 h	0	Homing Method	VAR	INT8	M	R/W	***not available***
6099 h	0	Homing Speeds	VAR	UINT32	M	R/W	***not available***
609A h	0	Homing Acceleration	VAR	UINT32	O	R/W	***not available***
60A2 h	0	Jerk factor	VAR	UINT32	O	R/W	***not available***
60A3 h	0	Profile Jerk Use	VAR	UINT8	O	R/W	***not available***
60A4 h	0	Profile Jerk	VAR	UINT32	O	R/W	***not available***
60A8 h	0	SI Unit Position	VAR	UINT32	O	R/W	***not available***
60A9 h	0	SI unit velocity	VAR	UINT32	O	R/W	***not available***
60B0 h	0	Position Offset	VAR	INT32	O	R/W	***not available***
60B1 h	0	Velocity Offset	VAR	INT32	O	R/W	***not available***
60B2 h	0	Torque Offset	VAR	INT16	O	R/W	***not available***
60C5 h	0	Max Acceleration	VAR	UINT32	O	R/W	Profile Velocity
60C6 h	0	Max Deceleration	VAR	UINT32	O	R/W	Profile Velocity
60E0 h	0	Positive Torque Limit Value	VAR	UINT16	O	R/W	***not available***

60E1 h	0	Negative Torque Limit Value	VAR	UINT16	O	R/W	***not available***
60F2 h	0	Position Option Code	VAR	UINT16	O	R/W	***not available***
60F4 h	0	Following Error Actual Value	VAR	INT32	O	RO	***not available***
60F8 h	0	Max Slippage	VAR	INT32	O	R/W	***not available***
60FA h	0	Control Effort	VAR	INT32	O	RO	***not available***
60FC h	0	Position Demand Internal Value	VAR	INT32	O	RO	***not available***
60FD h	0	Digital Inputs	VAR	UINT32	O	RO	Tell
60FE h	0	Digital Outputs	VAR	UINT32	O	RO	Settings
60FF h	0	Target Velocity	VAR	INT32	**M**	R/W	Profile Velocity
6402 h	0	Motor Type	VAR	UINT16	O	R/W	Tell
6403 h	0	Motor Catalogue Number	VAR	STRING	O	R/W	Tell
6404 h	0	Motor Manufacturer	VAR	STRING	O	R/W	Tell
6407 h	0	Motor Service Period	VAR	UINT32	O	R/W	***not available***
6502 h	0	Supported Drive Modes	VAR	UINT32	**M**	RO	Tell
6503 h	0	Drive Catalogue Number	VAR	STRING	O	R/W	***not available***
6504 h	0	Drive Manufacturer	VAR	STRING	O	R/W	***not available***

8. | FUNCTIONS

RAMP SPEED SET-UP

It's possible to set the drive in ramp mode. This Operation mode makes that the variation of speed can be defined by ramp defined by user.
This ramp operation mode not is active with STOP command or with a Switch Limit.

STOP WITH RAMP

It's possible to set the drive with stop with ramp mode. This Operation mode makes that the variation of speed can be defined by ramp defined by user.

If the stop ramp is active, each variation of speed set point will correspond a ramp with a parameter (in ms) programmable. This Stop with Ramp operation mode is independent of Speed with Ramp mode operation.

DIGITAL I/O

The drive has:

- 4 DIGITAL Input
- 4 DIGITAL Output
- 2 SAFETY Digital Input

Digital Input

The digital Inputs can be configured with different functions (see object 3020_h).

The drive has the digital inputs configured by manufacturer; if the user would change the configuration he must contact the manufacturer.

The standard of level to activate the function in digital input is edge positive.

The functions available are:

- Function "RUN": in Analog Mode (manufacturer Mode without CANopen communication) this input is the command to move the drive in RUN state.
- Function "STOP": in Analog Mode (manufacturer Mode without CANopen communication) this input is the command to move the drive in STOP state.
- Function "EMERGENCY INPUT ENABLE": when the option of digital input 3 is defined "Emergency Input Enable" this input is the command to move the drive in STANDBY state.
- Function "RESET": when this function is enabled the input configured can put the drive in reset (it is an hardware reset). If the digital input is configured as reset, the reset has a filter with 100ms.
- Function "DCW": this function is to configure the input with an actuator for clockwise. If the input state is 1 the drive goes in STOP state.
- Function "DCCW": this function is to configure the input with an actuator for counter clockwise. If the input state is 1 the drive goes in STOP state
- Function "VEL1": ***this function is not implemented yet***
- Function "VEL2": ***this function is not implemented yet***
- Function "VEL3": ***this function is not implemented yet***
- Function "V/C": ***this function is not implemented yet***
- Function "POL": ***this function is not implemented yet***

The default configuration is

- DIG-IN1 = function RUN used by Analog Mode Control
- DIG-IN2 = function STOP used by Mode Control
- DIG-IN3 = function EMERGENCY used to go from RUN State to STANDBY in emergency condition with dynamic brake.
- DIG-IN4 = function RESET HARDWARE (with timeout)

If the drive in controlled by CAN commands the State Machine follows the "controlword" (6060h) and Digital Input are ignored. When the drive is in "ANALOG MODE" so to move the motor are used 2 digital inputs.

Digital Output

- DIG-OUT1 = Drive Status
  - High Level 1= Driver OK
  - Low Level 0 = Fault
- DIG- OUT1 = Waning Status
  - High Level 1= at least on warning
  - Low Level 0 = NO warning
- DIG- OUT1 = free
- DIG- OUT4 = Brake Status
  - High Level 1= brake released, motor free
  - Low Level 0 = brake activated, motor blocked

Digital Safety Input

If it is available the "STO Safety" the STO is active the drive goes in SAFETY status independently to other selection. In case of FAULT the drive goes in FAULT State.

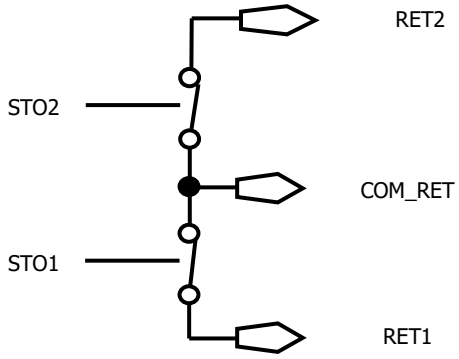


Figure 35 – STO Circuit

The follow picture shows the STO state machine:

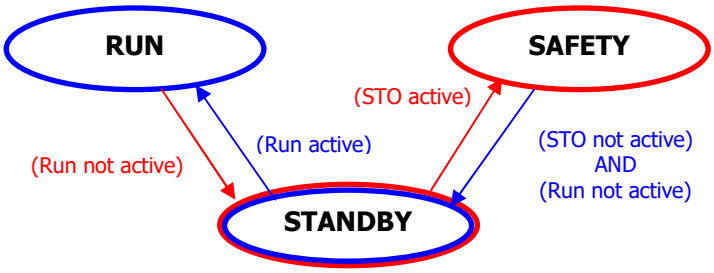


Figure 36 – STO transition State Machine

Caution

To reactive the standby status is mandatory that STO and RUN are not active and do the procedure

- in Analog Mode: SAFETY → STANDBY → RUN
- in CANopen Mode: SAFETY →"SWITCH ON DISABLED"→ "READY TO SWITCH ON" → "SWITCH ON" → "OPERATION ENABLED"

If an application requires controlled braking before using the STO function, the drive must be braked first and the STO function must be activate with a delay:

- Controlled braking of drive
- Once standstill is reached, disable the drive
- In the case of a suspended load, mechanically lock the drive as well
- Activate STO function.

Every single output relay is read.
Safety controller can read every single command corresponding to every relay output: fully monitoring of safety functions.

Caution

The Drive cannot hold the load with the STO function activated because the motor no longer supplies any torque.

- If the STO function is activated during operation, the drive will stop in an uncontrolled manner.
- If the drive has the Safety Torque OFF (STO), verify that this circuit is correctly supplied before all operation functions.

OTHER FUNCTIONALITY

The following paragraphs describe the CANopen command of the additional function of drive.

- Emergency Digital Input Enable
- Safety
- Emergency History
- Dynamic Brake
- Brake Management
- DAC monitoring

Emergency Digital Input Enable

This procedure is necessary to use the feature of digital input 3 as "Emergency Enable".

This feature must be configured. Send the object index 0x3008 and sub-index 1 with "1" value

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 08 30 01 01 00 00 00	WRITE Emergency Enable Input : TO ENABLE
Tx	0x581	60 08 30 01 00 00 00 00	

Now the Emergency Digital Input Enable is active.

Master Control can stop in emergency enabling the Digital Input 3 via hardware.

It means that it is possible to move the drive from RUN state ("Operation Enabled") to standby ("Switched-on") also using a digital Input 3 (High Signal default), if it is connected.

If dynamic brake is enabled the drive is stopping with ramp.

The drive can't go to run state if the digital input 3 continues to stay High Level.

Caution

This feature in not SAFETY function, but it is an additional protection via hardware to exit RUN state

When the digital Input Emergency is High Level then the bit 8 of Status word is 1.

The status of profile 402 is the "Switched On" (STAND-BY) is caused by Emergency Input Enable. Master Controller can read the status of the drive object 0x6041: xxxx xxx**1** x01x 0011b

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read SDO Status Word
Tx	0x581	4b 41 60 00 33 15 00 00	Status is "switched-on" and Emergency Input Enabled active: xxxx xxx1 x01x 0011b

Read the status of "Emergency Input Enable" via SDO (index 0x3008 and sub-index 3)

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read status of "Emergency Input Enable"
Tx	0x581	4b 08 30 03 01 00 00 00	1 = Emergency Input Enabled actives

It is possible to change the level. Write the object Index 0x3008 sub-index 2

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 08 30 02 01 00 00 00	WRITE Emergency Input Neg: Level Negative
Tx	0x581	60 08 30 02 00 00 00 00	Emergency Input Neg accepted

Safety

This procedure is necessary to read the drive Safety Mode.

Master Controller must active the STO Hardware Input to move the drive in safety state (SAFETY).

information

To enable Safety Mode you don't have to connect pin STO1 or STO2 at +24V.

Now the status of drive is the "SAFETY". Read the "Status Word" Object, Index 0x6041 and sub-index 0: xx1x xxxx xxxx xxxxb

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read SDO Status Word
Tx	0x581	4b 41 60 00 23 40 00 00	Bit 14: 0 = No Safety, 1 = Safety

Or read the "Safety State" Object, Index 0x4000 and sub-index 1:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 00 40 01 00 00 00 00	Read State "Safety State"
Tx	0x581	4b 00 40 01 01 00 00 00	0=no Safety, 1=Safety

Read the "Drive Mode" Object, Index 0x2002 and sub-index 0:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 02 20 00 00 00 00 00	Read State "Drive Control State"
Tx	0x581	4b 02 20 00 80 00 00 00	Value = 0x80 means the drive is in SAFETY state with STO applied

Emergency History

This procedure is to read the emergency history

To Read number of errors (sub-Index 0) occurred:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 03 10 00 00 00 00 00	Read number of errors (sub-Index 0)
Tx	0x581	4F 03 10 00 02 00 00 00	Response from CANopen

Byte 5: 02h means there are 2 error messages recorded

To delete the emergency messages by writing 0 to sub-index 0:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	22 03 10 00 00 00 00 00	Delete the emergency messages
Tx	0x581	60 03 10 00 00 00 00 00	Response from CANopen

To Read error message (sub-index 1 ... 15)

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 03 10 01 00 00 00 00	Read error message
Tx	0x581	43 03 10 01 00 FF 81 00	Response from CANopen

The error message code description is in section "Error Code" on the Emergency chapter.

To Read error message (sub-index 1...15) without alarm

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 03 10 01 00 00 00 00	Read error message
Tx	0x581	80 03 10 01 11 00 09 06	Response from CANopen

Request a sub-index without occurred error the following error message will be received

Dynamic Brake

When the Dynamic Brake Function is enabled then the drive exit from "Operation Enabled" (RUN STATE) with a ramp. When the ramp finishes, the drive lock brake motor and stay in STOP (with torque applied) for a delay time programmed with dynamic brake parameter. Finally the drive will turn in "Switched On" (STANDBY STATE) with locked brake (if the brake is automatic mode).

When the Dynamic Brake is disabled the drive will decrease speed with natural inertia. When the speed is zero the drive lock motor brake, but if deceleration speed is greater than delay time in "Brake timeout" (3002_h: 4), however the drive blocks brake.

The drive has default parameters that they depend by application. It is possible to change their parameters.

To change the default parameters then write the new value via SDO in the index object 3007_h and relative sub-index and store in e^2prome (using object 1010_h)

To active the Dynamic Brake Function write 1 in the "Dynamic Brake Option" parameter of Object Index 3007_h and sub-index 1.

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 07 30 01 01 00 00 00	Enable Dynamic Brake Option
Tx	0x581	60 07 30 01 00 00 00 00	

For example modify "Decrement step ramp" parameter, set value 100 [rpm*100/sec]

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 07 30 04 64 00 00 00	Write Decrement Step ramp
Tx	0x581	60 07 30 04 00 00 00 00	

Store parameter in e^2prom

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 10 10 01 73 61 76 65	Store All parameters
Tx	0x581	60 10 10 01 00 00 00 00	

And Reset all Nodes:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x00	81 00	Comand Reset All Nodes
Tx	0x701	00	Boot-up
Tx	0x701	00	Boot-up
Tx	0x81	00 00 00 00 00 00 00 00	Emergency Protocol= NO ERROR

Motor Brake Management

The Motor Brake Management is an output who can drive and supply power directly to a motor brake.

To enable this function, you must write 1 in "motor Brake Option" object 3002_h: 1.

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 02 30 01 01 00 00 00	Enable Brake
Tx	0x581	60 02 30 01 00 00 00 00	

Store parameter in e^2prom

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 10 10 01 73 61 76 65	Store All parameters
Tx	0x581	60 10 10 01 00 00 00 00	

And Reset all Nodes:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x00	81 00	Comand Reset All Nodes
Tx	0x701	00	Boot-up
Tx	0x701	00	Boot-up
Tx	0x81	00 00 00 00 00 00 00 00	Emergency Protocol= NO ERROR

The type of brake available is

- Magnetic Brake
- Spring Brake

It is possible to know what brake is used reading the index object 3002_h: 7.

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 02 30 07 00 00 00 00	Read Type brake
Tx	0x581	4B 02 30 07 01 00 00 00	1 = Magnetic

The Motor Brake can be configured in Automatic Mode or in Manual Mode.

- **Automatic Mode:** the brake will be released (Brake+ = 24V) automatically when the Drive is set in "Operation Enabled" (RUN STATE) and is activated automatically in other states.
- **Manual Mode:** the user can be released the brake using a dedicated command in the object "Digital Outputs" $60FE_h$: 1.

To set the Motor Brake in Manual Mode write 1 the "Automatic/Manual Mode Configuration" object 3002_h: 5

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 02 30 05 01 00 00 00	Write 1 in Automatic/Manual Mode Configuration
Tx	0x581	60 02 30 05 00 00 00 00	

To command the brake It need to use the object "Digital Outputs" $60FE_h$: 1.

Write 0 to active:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 FE 60 01 00 00 00 00	0 = Command Brake Active
Tx	0x581	60 FE 60 01 00 00 00 00	Brake Activated → Motor Locked

Write 1 to release:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 FE 60 01 01 00 00 00	1 = Command Brake Released
Tx	0x581	60 FE 60 01 00 00 00 00	Brake Released → Motor Free

DAC monitoring

It is possible to configure the analog output as a monitoring. The object to set the DAC configuration is the 3050_h

Analog Ouput = 0 is disabled:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 50 30 01 00 00 00 00	**0 = set Analog Output disabled**
Tx	0x581	60 50 30 01 00 00 00 00	

Analog Ouput configured as "General Purpose":

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 50 30 01 01 00 00 00	1 = set General Purpose
Tx	0x581	60 50 30 01 00 00 00 00	

It is possible to set a Digital value in the sub-index 2 and reading in the output.
For example: write 100 value object 3050_h :2

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 50 30 02 64 00 00 00	Value = 100
Tx	0x581	60 50 30 02 00 00 00 00	

Read analog Output (object 3050_h: 3)

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 50 30 03 00 00 00 00	Read Analog Output
Tx	0x581	4B 50 30 03 F3 00 00 00	0xF3

Analog Ouput configured as "Velocity Monitoring" value (object 3050_h: 2)

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2B 50 30 01 02 00 00 00	1 = set Velocity Monitoring
Tx	0x581	60 50 30 01 00 00 00 00	

Read the output (object 3050_h: 3)

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 50 30 03 00 00 00 00	Read Analog Output
Tx	0x581	4B 50 30 03 80 13 00 00	0x1380

9. | DIAGNOSTIC

MACRO DRIVE STATE	CANOpen STATE	STATUS 1 LED GREEN	STATUS 2 LED YELLOW	LED VIEW
INIT	Not Ready To Switch On	"BLINK" simultaneously	"BLINK" simultaneously	1 simultaneously 2 simultaneously
	Switch On Disabled Ready to Switch On	"BLINK" alternately	"BLINK" alternately	1 alternately 2 alternately
STANDBY	Switched On	"BLINK"	OFF	1 BLINK 50% 2 OFF
FAULT	Fault Fault reaction fault	"BLINK" [x]	"BLINK" [y]	1 see fault 2 chapter
RUN (RUNV / RUNC)	Operation Enabled	ON	OFF	1 ON 2 OFF
STOP	Quick Stop Active	ON	ON	1 ON 2 ON
SAFETY	-	OFF	"BLINK"	1 OFF 2 BLINK
COMMUNICATION ERROR	-	OFF	ON	1 OFF 2 ON

Table 34 - Led Status

Alarm	STATUS 1 CODE LED GREEN 1st Code	STATUS 2 CODE LED YELLOW 2nd Code	Alarm Description
A Group: (Temperature)			
Motor Over Temperature	**1**	**10**	Motor Temperature over threshold. The motor has reached a too high temperature for correct operation.
Heat Sink Over Temperature	**1**	**1**	Heat Sink Temperature over threshold. The Heat Sink has reached a too high temperature for correct operation.
Heat Sink Temperature Out Of Range	**1**	**3**	Heat Sink Temperature Sensor is out of range. Potential malfunction of the temperature sensor. (Contact the supplier).
Board Over Temperature	**1**	**4**	Internal Board Temperature over threshold. Too high a temperature for correct operation inside the drive.
Board Temperature Out Of Range	**1**	**5**	Internal Temperature Sensor out of range. Potential malfunction of the temperature sensor. (Contact the supplier).
Motor Temperature Out Of Range	**1**	**6**	Motor Temperature Sensor is out of range. • Potential malfunction of the temperature sensor. • (Contact the supplier).
B Group: (Feedback)			
Resolver	**2**	**10**	Check resolver connections, connectors and wiring of both

Alarm	STATUS 1 CODE LED GREEN 1st Code	STATUS 2 CODE LED YELLOW 2nd Code	Alarm Description
			sides.
Resolver Initialization	**2**	**4**	Initialization Faultfor Resolver Device. (Contact the supplier).
Encoder	**2**	**5**	Incremental Encoder Fault
SinCosFault	**2**	**6**	SinCos Encoder Fault
Hall	**2**	**7**	Hall Sensors Fault
Distance Hall	**2**	**8**	Hall Sensors Fault
C Group: (Current)			
Offset Current Sensor	**3**	**10**	Offset current sensor isoutof range. (Contact Supplier).
Over Current	**3**	**1**	The current absorbed by the motor is beyond the set limit. Check Phase Motor connection and wire. Look for any short circuits.
D Group: (Voltage)			
Under Voltage	**4**	**1**	DC Bus voltage value lower than the limit threshold. Check mains voltage at terminals +, -.
Over Voltage	**4**	**2**	DC Bus voltage value higher than the limit threshold. Check mains voltage at terminals +, -
E Group: (Functionality)			
Velocity Fault	**5**	**10**	Theactual speed differs fromthe target Speed.
I^2T Overload Protection	**5**	**2**	I2T overload motor protection reached.
Hardware	**5**	**3**	Error Hardware (Contact Supplier)
External HW	**5**	**4**	Error CAN Interface (Contact Supplier)
F Group: (Communication and Configuration)			
E^2prom	**6**	**1**	Parameter Fault stored in E2prom.
CanOpen	**6**	**2**	Communication Faultwith CANOpen
Sincos Fault	**6**	**3**	Internal Communication Fault (Contact Supplier).
Configuration Parameters	**6**	**4**	Configuration Parameters Fault (Contact Supplier).
Profile Generic	**6**	**5**	Error Configuration Profile: Mode Of Operation
Torque Profile	**6**	**6**	Error Torque Profile
Velocity Profile	**6**	**7**	Error Velocity Profile
G-H-L Group: (Programming)			
Program Fault	**7**	**x**	Code Programming Fault (Contact Supplier).
	8	**X**	
	9	**x**	

Table 35 - Diagnostic

10. | APPENDIX - FIRST CONFIGURATION

POWER-ON

On the Power-On the, if the CAN communication is OK, the drive sends these message:

Tx/Rx	ID	VALUE	DESCRIPTION
Tx	0x701	00	Boot-up
Tx	0x701	00	Boot-up
Tx	0x81	00 00 00 00 00 00 00 00	Emergency Protocol= NO ERROR

The drive has default values.

It is possible to change the default value writing via SDO protocol the corresponding index and sub-index object and store in e^2prom.

After that you do not need to write at every power on, and values are updated by memory e^2prom.

Caution

During STORE procedure, drive must not be in "Operation Enabled" state or in "Quick Stop Active" state.

To Store parameter in e^2prom (permanently) it need to send the object 1010_h and reset

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 10 10 01 73 61 76 65	Store All parameters
Tx	0x581	60 10 10 01 00 00 00 00	

and Reset all Nodes (or switch-off/switch-on)

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x00	81 00	Command Reset All Nodes
Tx	0x701	00	Boot-up
Tx	0x701	00	Boot-up
Tx	0x81	00 00 00 00 00 00 00 00	Emergency Protocol= NO ERROR

HOW TO CHANGE ID-NODE

Id-Node has default Value = 1. The following steps describe how to change the Id-Node.

Caution

To Change Id-Node it is mandatory connect one drive on the time with Master Controller

Procedure Set New Id-Node Value (Write SDO)

The Master Control sends SDO message ID = 0x601 (defined 0x600 + Id node)

- Data Value "command" = 0x2F
- Data Value "Index" = 0x2000
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = new Id-Node value (for Example 3)

The drive answers SDO message ID = 0x581 (defined 0x580 + Id node)

- Data Value "command" = 0x60
- Data Value "Index" = 0x2000
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0

The following picture shows the SDO message:

ID	Name	Node	Transfer data	Interpretation	Data
601	CSDO_001	Node1	03	[2000,00] Initiate Download Rq. expedited	2F 00 20 00 03 00 00 00
581	SSDO_001	Node1		[2000,00] Initiate Download Rsp	60 00 20 00 00 00 00 00

Procedure Save New Value in e^{2}prom (Write SDO)

Caution

During STORE procedure, drive must not be in "Operation Enabled" state or in "Quick Stop Active" state.

The Master Control sends SDO message ID = 0x601 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x1010 (Store)
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0x73617665 (means "save" in ASCII code)

The drive answers SDO message ID = 0x581 (defined 0x580 + Id node)

- Data Value "command" = 0x60
- Data Value "Index" = 0x1010
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0

The following picture shows the SDO message:

ID	Name	Node	Transfer data	Interpretation	Data
601	CSDO_001	Node1	73 61 76 65	[1010,01] Initiate Download Rq. expedited "save"	23 10 10 01 73 61 76 65
581	SSDO_001	Node1		[1010,01] Initiate Download Rsp	60 10 10 01 00 00 00 00

The procedure continues ...

after stored parameters continue with reset

Reset All Nodes (NMT Protocol)

The sends message ID = 0x00 (NMT protocol)

- Data Value "command" = 0x81
- Data Value "Index" = 0x00

The following picture shows the SDO message:

Dir	ID	Name	Node	Transfer data	Interpretation	Error	Data
Rx	0	NMTZeroMsg			Reset all nodes		81 00

After Reset (NMT Protocol)

The drive answers message BOOT-UP message ID = 0x703 (defined 0x700 + Id node)

- Data Value "Index" = 0x00

ID	Name	Node	Transfer data	Interpretation	Error	Data
703			Boot-up			00
703	_		Boot-up			00

The procedure continues ...

then the drive sends emergency messages (emergency protocol)

The drive sends message ID = 0x83 (defined 0x80 + Id node)

- Data Value "Error Code" = 0x0
- Data Value "Reg" = 0x0
- "Data" = 0

It means "ERROR RESET or NO ERROR"

ID	Name	Node	Transfer data	Interpretation	Error	Data
83	EMCY_003		00 00 00 00 00 00 00 00	Error reset or no error	E	00 00 00 00 00 00 00 00

The procedure continues ...

Then, to be sure, the drive accepted previous id-node changed.

Procedure Verify New Id-Node (Read SDO)

The Master Control sends SDO message ID = 0x603 (defined 0x600 + New Id node)

- Data Value "command" = 0x40
- Data Value "Index" = 0x2000
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0x00

The drive answers SDO message ID = 0x583 (defined 0x580 + New Id node)

- Data Value "command" = 0x4F
- Data Value "Index" = 0x2000
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0x3

The following picture shows the SDO messages:

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003			[2000,00] Initiate Upload Rq.		40 00 20 00 00 00 00 00
583	SSDO_003		03	[2000,00] Initiate Upload Rsp. expedited		4F 00 20 00 03 00 00 00

HOW TO CHANGE BAUDRATE

BaudRate Default is 1000Kbit. The following steps describe how to change the BaudRate.

Procedure Set New Baudrate Value (Write SDO)

The Master Control sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x2B
- Data Value "Index" = 0x2001
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = new BaudRate (for Example 500K = 0x01F4)

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x60
- Data Value "Index" = 0x2001
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0

The following picture shows the SDO messages:

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003		f4 01	[2001,00] Initiate Download Rq. expedited		2B 01 20 00 F4 01 00 00
583	SSDO_003			[2001,00] Initiate Download Rsp		60 01 20 00 00 00 00 00

Procedure Save New Value In e^{2}prom (Write SDO)

Caution

During STORE procedure, drive must not be in "Operation Enabled" state or in "Quick Stop Active" state.

The Master Control sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x1010 (Store)

- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0x73617665 (means "save" in ASCII code)

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x60
- Data Value "Index" = 0x1010
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0

The following picture shows the SDO message:

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003		73 61 76 65	[1010,01] Initiate Download Rq. expedite...		23 10 10 01 73 61 76 65
583	SSDO_003			[1010,01] Initiate Download Rsp		60 10 10 01 00 00 00 00

The procedure continues ...

After stored parameters proceed with Reset

Reset All Nodes (NMT Protocol)

The Master Control sends message ID = 0x00 (NMT protocol)

- Data Value "command" = 0x81
- Data Value "Index" = 0x00

The following picture shows the SDO message:

Dir	ID	Name	Node	Transfer data	Interpretation	Error	Data
Rx	0	NMTZeroMsg			Reset all nodes		81 00

After Reset (NMT Protocol)

(See CANopen Manual page 31)

The drive answers message BOOT-UP message ID = 0x703 (defined 0x700 + Id node)

- Data Value "Index" = 0x00

ID	Name	Node	Transfer data	Interpretation	Error	Data
703			Boot-up			00
703	_		Boot-up			00

The procedure continues ...

then the drive sends emergency messages (emergency protocol)

The drive sends message ID = 0x83 (defined 0x80 + Id node)

- Data Value "Error Code" = 0x0
- Data Value "Reg" = 0x0
- "Data" = 0

It means "ERROR RESET or NO ERROR"

ID	Name	Node	Transfer data	Interpretation	Error	Data
83	EMCY_003		00 00 00 00 00 00 00 00	Error reset or no error	E	00 00 00 00 00 00 00 00

The procedure continues ...

then, to be sure, the drive accepted previous id-node changed

Procedure Verify New BaudRate (Read SDO)

The Master Control sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x40
- Data Value "Index" = 0x2001
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0x00

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x4B
- Data Value "Index" = 0x2000
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0x01F4

The following picture shows the SDO messages:

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003			[2001,00] Initiate Upload Rq.		40 01 20 00 00 00 00 00
583	SSDO_003		f4 01	[2001,00] Initiate Upload Rsp. expedited		4B 01 20 00 F4 01 00 00

HOW TO CHANGE THE USER UNITS

Lafert Servo Drive has a default unit [inc/s] for velocity objects and [inc/s^2] for acceleration objects. If it is necessary to change the user unit (for example in [rpm] for velocity objects and [rpm/s] for acceleration objects) it has to change the factory group object.

The velocity factory group is

$$Velocity\ Factor = \frac{Numerator}{Divisor}$$

Numerator and divisor of the Velocity Factor has to be entered separately.

The default value is [inc/s]. The numerator and the divisor are set "1" in e^{2}prom.

To change the default unit it has to write the numerator and divisor in the object index 0x6096 and save in e^{2}prom the new value:

$$Velocity[internal\ unit] = Velocity[user\ unit] \times \left(\frac{60}{Resolution}\right) \times \left(\frac{Numerator}{Divisor}\right)$$

Example:

The speed-set point provision is to be made in revolutions per minute (rpm).

$$Velocity[inc/sec] = Velocity[rpm] \times \left(\frac{60}{Resolution}\right) \times \left(\frac{Numerator}{Divisor}\right)$$

If the resolution of encoder is 2^{13} =16384 then the Numerator is 16384 and the Divisor is 60

The Acceleration Factory Group has the same consideration (object index 0x6097)

Procedure Set New Factory Group Values (Write SDO)

Write **NUMERATOR** Velocity Factory Group (value = 16384):

The Master Control sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x6096
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 16384 = 0x4000

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x6096
- Data Value "Index" = 0x01
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0

Write **DIVISOR** Velocity Factory Group (value = 60):

The Master Control sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x6096
- Data Value "Sub-Index" = 0x02
- Data Value "Data" = 60 = 0x3C

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x6096
- Data Value "Index" = 0x02
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0

Write **NUMERATOR** Acceleration Factory Group (value = 16384):

The Master Controller sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x6097
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 16384 = 0x4000

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x6097
- Data Value "Index" = 0x01
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0

Write **DIVISOR** Acceleration Factory Group (value = 60):

The Master Controller sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x6097
- Data Value "Sub-Index" = 0x02
- Data Value "Data" = 60 = 0x3C

The drive answers SDO message ID = 0x583(defined 0x580 + Id node)

- Data Value "command" = 0x6097
- Data Value "Index" = 0x02
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0

The following picture shows the SDO messages:

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003		00 40 00 00	[6096,01] Initiate Download Rq. expedited ".@.."		23 96 60 01 00 40 00 00
583	SSDO_003			[6096,01] Initiate Download Rsp		60 96 60 01 00 00 00 00
603	CSDO_003		3c 00 00 00	[6096,02] Initiate Download Rq. expedited "<..."		23 96 60 02 3C 00 00 00
583	SSDO_003			[6096,02] Initiate Download Rsp		60 96 60 02 00 00 00 00
603	CSDO_003		00 40 00 00	[6097,01] Initiate Download Rq. expedited ".@.."		23 97 60 01 00 40 00 00
583	SSDO_003			[6097,01] Initiate Download Rsp		60 97 60 01 00 00 00 00
603	CSDO_003		3c 00 00 00	[6097,02] Initiate Download Rq. expedited "<..."		23 97 60 02 3C 00 00 00
583	SSDO_003			[6097,02] Initiate Download Rsp		60 97 60 02 00 00 00 00

The procedure continues ...

Now the user units are ready to save in E^2prom, but they aren't available. **It MUST STORE and RESET**.

Procedure Save New Value in e^2prom (Write SDO)

Caution

During STORE procedure, drive must not be in "Operation Enabled" state or in "Quick Stop Active" state.

The Master Controller sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x1010 (Store)
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0x73617665 (means "save" in ASCII code)

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x60
- Data Value "Index" = 0x1010
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0

The following picture shows the SDO message:

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003		73 61 76 65	[1010,01] Initiate Download Rq. expedite...		23 10 10 01 73 61 76 65
583	SSDO_003			[1010,01] Initiate Download Rsp		60 10 10 01 00 00 00 00

The procedure continues ...

After stored parameters proceed with drive reset

Reset All Nodes (NMT Protocol)

The Master Controller sends message ID = 0x00 (NMT protocol)

- Data Value "command" = 0x81
- Data Value "Index" = 0x00

The following picture shows the SDO message:

Dir	ID	Name	Node	Transfer data	Interpretation	Error	Data
Rx	0	NMTZeroMsg			Reset all nodes		81 00

After Reset (NMT Protocol)

The drive answers message BOOT-UP message ID = 0x703 (defined 0x700 + Id node)

- Data Value "Index" = 0x00

ID	Name	Node	Transfer data	Interpretation	Error	Data
703			Boot-up			00
703	_		Boot-up			00

The procedure continues ...

Then the drive sends Emergency Messages (EMERGENCY PROTOCOL)

The drive sends message ID = 0x83 (defined 0x80 + Id node)

- Data Value "Error Code" = 0x0
- Data Value "Reg" = 0x0
- "Data" = 0

It means "ERROR RESET or NO ERROR"

ID	Name	Node	Transfer data	Interpretation	Error	Data
83	EMCY_003		00 00 00 00 00 00 00 00	Error reset or no error	E	00 00 00 00 00 00 00 00

OBJECT WITH DIFFERENT DEFAULT

If the factory group changed than it is mandatory to change in [user unit] all values related to velocity and acceleration/deceleration of velocity profile mode and save in e^2prom.

After reset (or power-on) the drive initializes the object from e^2prom.

Caution

It is important to change before the maximum value.

If after power-on the value is not correct the drive send an emergency message with error code 0x8B06. It has also a manufacture code that defines what object has a wrong value.

Manufacturer specific value describes what index object is failed:

- (Bit 0): Error Init Object 0x6081
- (Bit 1): Error Init Object 0x6082
- (Bit 2): Error Init Object 0x6083
- (Bit 3): Error Init Object 0x6084
- (Bit 4): Error Init Object 0x60C5
- (Bit 5): Error Init Object 0x60C6
- (Bit 6): Error Init Object 0x607F
- (Bit 7): Error Init Object 0x6088
- (Bit 8) : Error Init Object 0x6096
- (Bit 9) : Error Init Object 0x6097
- (Bit 10) : Error Init Object 0x606D
- (Bit 11) : Error Init Object 0x606E
- (Bit 12) : Error Init Object 0x606F
- (Bit 13) : Error Init Object 0x6070
- (Bit 14) : Error Init Object 0x6075

- (Bit 15) : Error Init Object 0x6076
- (Bit 16) : Error Init Object 0x6072
- (Bit 17) : Error Init Object 0x6073
- (Bit 18) : Error Init Object 0x60E0
- (Bit 19) : Error Init Object 0x60E1
- (Bit 20) : Error Init Object 0x6087
- (Bit 21) : Error Init Object 0x6086

It has to change the following objects (example new value):

Index	Sub Index	Name Object	Default Value [internal unit]	New Value [user unit]
0x606D	0	Velocity Window	13653 inc/sec	50 rpm
0x606F	0	Velocity Threshold	1365 inc/s	5 rpm
0x6083	0	Profile Acceleration	273066 inc/s^2	1000 rpm/s
0x6084	0	Profile Deceleration	273066 inc/s^2	1000 rpm/s
0x60C5	0	Max Acceleration	608393 inc/s^2	2228 rpm/s
0x60C6	0	Max Deceleration	608393 inc/s^2	2228 rpm/s

Caution

Pay attention at the order to write the new values.

It is important to define before the max value before the other objects.

Procedure Set New Values in User Unit (Write SDO)

Save Max Acceleration: new value 2228 [rpm/s]

The Master Controller sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x60C5
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 2228 = 0x08B4

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x60C5
- Data Value "Index" = 0x00
- Data Value "Sub-Index" = 0x00
- Data Value "Data" = 0

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003		b4 08 00 00	[60c5,00] Initiate Download Rq. expedited		23 C5 60 00 B4 08 00 00
583	SSDO_003			[60c5,00] Initiate Download Rsp		60 C5 60 00 00 00 00 00

The procedure continues ...

It has to write all objects with new values in [user unit]. It has to use the same procedure.

Procedure Save New Value in E^{2}prom (Write SDO)

Caution

During STORE procedure, drive must not be in "Operation Enabled" state or in "Quick Stop Active" state.

The Master Controller sends SDO message ID = 0x603 (defined 0x600 + Id node)

- Data Value "command" = 0x23
- Data Value "Index" = 0x1010 (Store)
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0x73617665 (means "save" in ASCII code)

The drive answers SDO message ID = 0x583 (defined 0x580 + Id node)

- Data Value "command" = 0x60
- Data Value "Index" = 0x1010
- Data Value "Sub-Index" = 0x01
- Data Value "Data" = 0

The following picture shows the SDO message:

ID	Name	Node	Transfer data	Interpretation	Error	Data
603	CSDO_003		73 61 76 65	[1010,01] Initiate Download Rq. expedite...		23 10 10 01 73 61 76 65
583	SSDO_003			[1010,01] Initiate Download Rsp		60 10 10 01 00 00 00 00

The procedure continues ...

After Stored Parameters Proceed With reset drive

Reset All Nodes (NMT Protocol)

The Master Controller sends message ID = 0x00 (NMT protocol)

- Data Value "command" = 0x81
- Data Value "Index" = 0x00

The following picture shows the SDO message:

Dir	ID	Name	Node	Transfer data	Interpretation	Error	Data
Rx	0	NMTZeroMsg			Reset all nodes		81 00

After Reset (NMT Protocol)

The drive answers message BOOT-UP message ID = 0x703 (defined 0x700 + Id node)

- Data Value "Index" = 0x00

ID	Name	Node	Transfer data	Interpretation	Error	Data
703			Boot-up			00
703			Boot-up			00

The procedure continues …

Then The Drive sends emergency messages (EMERGENCY PROTOCOL)

The drive sends message ID = 0x83 (defined 0x80 + Id node)

- Data Value "Error Code" = 0x0
- Data Value "Reg" = 0x0
- "Data" = 0

It means "ERROR RESET or NO ERROR"

ID	Name	Node	Transfer data	Interpretation	Error	Data
83	EMCY_003		00 00 00 00 00 00 00 00	Error reset or no error	E	00 00 00 00 00 00 00 00

11. | APPENDIX - EXAMPLE PROGRAMS

In this chapter the typical course of action is shown to launch a CANopen-drive.

In these examples the drive's Id Node is 1.

PROFILE VELOCITY PROCEDURE

The led status code shows that the drive is in "Switch on disabled" state (CANOpen Profile DS402) and led will blink alternatively [MACRO DRIVE STATE: INIT].

Set Mode of Operation

Caution

To configure the Profile Velocity Mode the drive isn't in "Operation enabled" state.

MASTER must send SDO "Mode of Operation" object, index 0x6060 and sub-index 0, with value 3:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2f 60 60 00 03 00 00 00	Mode of Operation Request Profile velocity
Tx	0x581	60 60 60 00 00 00 00 00	

Go to the State "Switched-On"

Change state in machine state of CANOpen Profile DS402 in "Switched On". MASTER must send twice time the SDO "control word" object, index 0x6040, with value 6 and with value 7

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 40 60 00 06 00 00 00	Change state in "Ready to Switch On"
Tx	0x581	60 40 60 00 00 00 00 00	
Rx	0x601	2b 40 60 00 07 00 00 00	
Tx	0x581	60 40 60 00 00 00 00 00	

Now the status of profile 402 is "Switched On"[MACRO DRIVE STATE: STAND-BY]. Read the "Status Word" Object, Index 0x6041 and sub-index 0: xxxx xxxx x01x 0011b

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read SDO Status Word
Tx	0x581	4b 41 60 00 23 00 00 00	"switched On": xxxx xxxx x01x 0011b

Set Acceleration e Deceleration

Configure the acceleration and the deceleration for profile velocity. (For example to configure acceleration 1000 rpm/sec). MASTER must send SDO "factor group" index 0x6083 and index 0x6084.

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	**0x601**	**23 83 60 00 e8 03 00 00**	**Acceleration 1000 rpm/s**
Tx	**0x581**	**60 83 60 00 00 00 00 00**	
Rx	**0x601**	**23 84 60 00 e8 03 00 00**	**Deceleration 1000 rpm/s**
Tx	**0x581**	**60 84 60 00 00 00 00 00**	

Go to the State "Operation Enabled".

Change state in machine state in "Operation Enabled". MASTER must send twice time the SDO "control word" object, index 0x6040, with value 15

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 40 60 00 0F 00 00 00	Change state in "Operation Enabled"
Tx	0x581	60 40 60 00 00 00 00 00	

Now the status of profile 402 is "Operation Enabled" [MACRO DRIVE STATE: RUN]. Read the "Status Word" Object, Index 0x6041 and sub-index 0: xxxx xxxx x01x 0111b

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read SDO Status Word
Tx	0x581	4b 41 60 00 27 00 00 00	"switched On": xxxx xxxx x01x 0111b

Set Target Velocity

Set the Set Point of Velocity. MASTER must send SDO "Target Velocity" index 0x60FF, i.e. 1000 RPM

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 ff 60 00 e8 03 00 00	Set Point Velocity
Tx	0x581	60 ff 60 00 00 00 00 00	

Read the "Velocity Actual Value" Object, Index 0x606C and sub-index 0. MASTER must send SDO 606Ch:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 6c 60 00 00 00 00 00	Read Actual Velocity
Tx	0x581	43 6c 60 00 e8 03 00 00	

Now MASTER can stop the drive in different ways. (NO EMERGENCY STOPS considered, in case of Emergency stops please refer to next paragraph).

Set the value of target velocity at 0. MASTER must send SDO "Target Velocity" 0x60FF with value 0:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 ff 60 00 00 00 00 00	Set Point Velocity
Tx	0x581	60 ff 60 00 00 00 00 00	

information

In this case drive will stop with proper ramp and will stay in Powered (with torque applied)

Change state in machine state in "Switched On". [MACRO DRIVE STATE: STAND-BY]. No torque applied to the motor. MASTER must send SDO "control word" object 0x6040 with value 7:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 40 60 00 07 00 00 00	Set State "Switched On"
Tx	0x581	60 40 60 00 00 00 00 00	

Now the status of profile 402 is the "Switched On". Read the "Status Word" Object, Index 0x6041 and sub-index 0: xxxx xxxx x01x 0011b

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read SDO Status Word
Tx	0x581	4b 41 60 00 23 00 00 00	

Change state in machine state in "Switched On Disabled". [MACRO DRIVE STATE: INIT]. No torque applied to the motor. MASTER must send SDO "control word" 0x6040 with value 0:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 40 60 00 00 00 00 00	Set State "Switched On Disabled"
Tx	0x581	60 40 60 00 00 00 00 00	

Now the status of profile 402 is the "Switch on disabled".Read the "Status Word" Object, Index 0x6041 and sub-index 0: xxxx xxxx x1xx 0000b.

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read Status Word
Tx	0x581	4b 41 60 00 40 00 00 00	"Switch On disabled": xxxx xxxx x1xx 0000b

Change state in machine state in "Quick Stop Active" [MACRO DRIVE STATE: STOP].Torque applied to the motor. MASTER sends SDO "control word" with value 2:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 40 60 00 02 00 00 00	Set State "Quick Stop Active"
Tx	0x581	60 40 60 00 00 00 00 00	

Now the status of profile 402 is the "Quick Stop Active".Read the "Status Word" Object, Index 0x6041 and sub-index 0: xxxx xxxx x00x 0111b

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 41 60 00 00 00 00 00	Read SDO Status Word
Tx	0x581	4b 41 60 00 07 00 00 00	"switched On" state: xxxx xxxx x00x 0111b

Trace Log Drive with SDO protocol (Target Velocity 1000 rpm)

ID	Name	Node	Transfer data	Interpretation	Error	Data	Counte
701	HBGuard_001	Node1	Boot-up			00	1
81	EMCY_001	Node1	00 00 00 00 00 00 00 00	Error reset or no error	E	00 00 00 00 00 00 00 00	1
81	EMCY_001	Node1	70 81 11 00 00 00 00 00	Communication - generic	E	70 81 11 00 00 00 00 00	2
601	CSDO_001	Node1	03	[6060,00] Initiate Download Rq. expedited		2F 60 60 00 03 00 00 00	1
581	SSDO_001	Node1		[6060,00] Initiate Download Rsp		60 60 60 00 00 00 00 00	1
601	CSDO_001	Node1	06 00	[6040,00] Initiate Download Rq. expedited		2B 40 60 00 06 00 00 00	2
581	SSDO_001	Node1		[6040,00] Initiate Download Rsp		60 40 60 00 00 00 00 00	2
601	CSDO_001	Node1		[6041,00] Initiate Upload Rq.		40 41 60 00 00 00 00 00	3
581	SSDO_001	Node1	31 10	[6041,00] Initiate Upload Rsp. expedited "1."		4B 41 60 00 31 10 00 00	3
601	CSDO_001	Node1	07 00	[6040,00] Initiate Download Rq. expedited		2B 40 60 00 07 00 00 00	4
581	SSDO_001	Node1		[6040,00] Initiate Download Rsp		60 40 60 00 00 00 00 00	4
601	CSDO_001	Node1		[6041,00] Initiate Upload Rq.		40 41 60 00 00 00 00 00	5
581	SSDO_001	Node1	33 10	[6041,00] Initiate Upload Rsp. expedited "3."		4B 41 60 00 33 10 00 00	5
601	CSDO_001	Node1	0f 00	[6040,00] Initiate Download Rq. expedited		2B 40 60 00 0F 00 00 00	6
581	SSDO_001	Node1		[6040,00] Initiate Download Rsp		60 40 60 00 00 00 00 00	6
601	CSDO_001	Node1	e8 03 00 00	[60ff,00] Initiate Download Rq. expedited		23 FF 60 00 E8 03 00 00	7
581	SSDO_001	Node1		[60ff,00] Initiate Download Rsp		60 FF 60 00 00 00 00 00	7
601	CSDO_001	Node1		[606c,00] Initiate Upload Rq. !UNUSED FIELDS USED	P	4B 6C 60 00 00 00 00 00	8
581	SSDO_001	Node1	e7 03 00 00	[606c,00] Initiate Upload Rsp. expedited		43 6C 60 00 E7 03 00 00	8
601	CSDO_001	Node1	07 00	[6040,00] Initiate Download Rq. expedited		2B 40 60 00 07 00 00 00	9
581	SSDO_001	Node1		[6040,00] Initiate Download Rsp		60 40 60 00 00 00 00 00	9
601	CSDO_001	Node1		[6041,00] Initiate Upload Rq.		40 41 60 00 00 00 00 00	10
581	SSDO_001	Node1	33 14	[6041,00] Initiate Upload Rsp. expedited "3."		4B 41 60 00 33 14 00 00	10

Trace Log Drive with PDO protocol (Target Velocity 1000 rpm)

ID	Name	Node	Transfer data	Interpretation	Error	Data	Counter
0	NMTZeroMsg			Reset all nodes		81 00	2
701	HBGuard_001	Node1	Boot-up			00	3
701	HBGuard_001	Node1	Boot-up			00	4
81	EMCY_001	Node1	00 00 00 00 00 00 00 00	Error reset or no error	E	00 00 00 00 00 00 00 00	3
81	EMCY_001	Node1	70 81 11 00 00 00 00 00	Communication - generic	E	70 81 11 00 00 00 00 00	4
201	ID1_RPDO1		06 00 03 00 00 0f 00			06 00 03 00 00 0F 00	273
381	ID1_TPDO3		RTR			Remote-Frame	7
381	ID1_TPDO3		31 10 00 00 00 00			31 10 00 00 00 00	8
201	ID1_RPDO1		07 00 03 00 00 0f 00			07 00 03 00 00 0F 00	274
381	ID1_TPDO3		RTR			Remote-Frame	9
381	ID1_TPDO3		33 10 00 00 00 00			33 10 00 00 00 00	10
201	ID1_RPDO1		0f 00 03 00 00 0f 00			0F 00 03 00 00 0F 00	275
381	ID1_TPDO3		RTR			Remote-Frame	11
381	ID1_TPDO3		37 10 00 00 00 00			37 10 00 00 00 00	12
401	ID1_RPDO3		0f 00 e8 03 00 00			0F 00 E8 03 00 00	4
381	ID1_TPDO3		RTR			Remote-Frame	13
381	ID1_TPDO3		37 14 e7 03 00 00			37 14 E7 03 00 00	14
201	ID1_RPDO1		07 00 03 00 00 0f 00			07 00 03 00 00 0F 00	276

READ VERSION RELEASE

Command to read the version release $100A_h$ object:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	40 0A 10 00 00 00 00 00	
Tx	0x581	43 0A 10 00 **31 30 38 00**	

In ASCII code = 0x31, 0x30, 0x38, 0x00 = 108 version firmware released

12. | APPENDIX – HEARTBEAT MECHANISM

One of the protective mechanisms available in CANopen is the heartbeat mechanism.

This mechanism allows the network master to detect a loss of communication from the network slaves, and it also allows the network slaves to react to a loss of communication from the master.

Lafert servo drives are compliant with the DS-301 and DS-402 versions of the CANopen protocol which define functions related to the heartbeat mechanism.

If the heartbeat is activated, when the drive detects a communication loss, it goes in "Fault" state automatically and the alarm is sent.

Heartbeat Sources and Message Structures

The standard DS301 describes that the CANopen nodes can be configured to transmit heartbeat messages and they can also be configured to monitor heartbeats from the host.
Nodes that generate heartbeats are called "producers", and nodes that monitor heartbeats are called "consumers".

The following picture is of DS301 standard document CiA301:

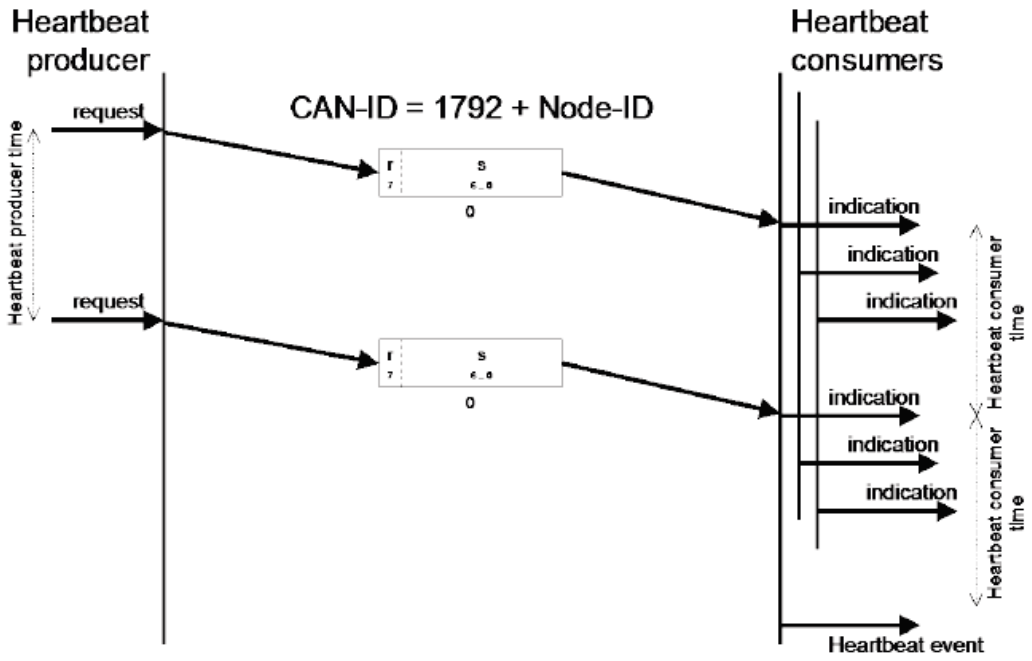


Figure 37 - Heartbeat Mechanism by DS301

Master Heartbeat:

The master heartbeat has the following characteristics:

- Produced by the CANopen master
- Consumed by CANopen slave nodes
- COB ID is 0x700
- The data frame can be empty

Master heartbeat message:

COB-ID	Rx/Tx	DLC	Byte							
			0	1	2	3	4	5	6	7
0x700	Rx	0								

Slave Heartbeat:

The slave heartbeat has the following characteristics:

- Produced by slave nodes on the network
- Consumed by the CANopen master
- The COB ID range is in the range 0x701 – 0x77F
- The data frame is 1 byte in length and contains a description of the slave node's communication state according to the table below:

Heartbeat Value	Description
0x0	Boot-up
0x1	Off bus
0x4	Stopped
0x5	Operational
0x7F	Pre-operational

Slave heartbeat message:

COB-ID	Rx/Tx	DLC	Byte							
			0	1	2	3	4	5	6	7
0x700 + Id Node	Tx	1	NMT State						-	

If the network is composed by 1 PLC and two drive than the drives must be configured as producer and the PLC as consumer.

Drive Configuration:

Lafert Servo Drive can be ONLY as a Heartbeat Producer by settings object 0x1017.

If the heartbeat producer time defined by 1017_h object is configured on the Lafert Servo Drive, the producer heartbeat protocol begins immediately and transmits producer heartbeat messages periodically.

Heartbeat monitoring starts as soon as the time interval of the producer is greater than zero. If the Heartbeat protocol is already active during the NMT state transition to "Pre-Operational", Heartbeat protocol starts with sending of the boot-up message. The boot-up message is Heartbeat message with one data byte 00_h.

To configure the 1017_h object (Producer Heartbeat Time) see the relative paragraph.

The time intervals are set in increments of 1ms steps; the values for the consumer must not be less than the values for the producer. Whenever the "Heartbeat" message is received, the time interval of the producer is restarted.

Example:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	2b 17 10 00 64 00 00 00	Drive Set Time HeratBeat 100ms (0x64)
Tx	0x581	60 17 10 00 00 00 00 00	
Tx	0x701	05	The drive send periodically the heartbeat message
Tx	0x701	05	with 0x5 value (State Drive is Operational)
Tx	0x701	05	
...	...	...	

Master Configuration:

The master controller must be configured as consumer. It has to define the 1016h object.

This object defines the period of time (ms) where a heartbeat from the master node is expected at the beginning of the period.

Monitoring begins on reception of the first heartbeat. A value of 0 disables heartbeat monitoring.

Object Description:

Index	Name	Object Code	Data Type	Category
1016_h	Consumer Heartbeat Time	ARRAY	U16	Optional

Entry Description:

Sub-Index	Description	Access	PDO mapping	Default Value
00_h	Highest sub-index supported	ro	no	0
01_h	Consumer Heartbeat Time 1st Node	rw	no	0
02_h	Consumer Heartbeat Time 2nd Node	rw	no	0

Value Definition:

31 24	23 16	15 0
reserved	Node Id	HeartBeat Time
MSB		LSB

Example:

Tx/Rx	ID	VALUE	DESCRIPTION
Rx	0x601	23 16 10 01 64 00 00 00	Set the consumer heartbeat time on node 1 to 100 ms (0x64)
Tx	0x581	60 16 10 01 00 00 00 00	

13. | APPENDIX – POSITION MONITORING

Between the motor shaft and the AGV wheel there is a gear box with a gear ratio. This value means that one revolution of AGV wheel corresponds to 21 motor shaft revolutions.

For example: diameter 21 mm.

When motor shaft makes one revolution the value of "Position Actual Value" 6064_h object increment or decrement (depending by direction of rotation) by resolution units.

Caution

The object 6064_h is initialized a 0 every time you reset or power up the drive, it is independent of position wheel.

Wheel Rotation:

Supposing the drive moves the motor with a clockwise rotation of motor shaft.

The "Position Actual Value" (6064_h) object was increased from 0.

If the value is 386007 increments then the motor shaft is rotated of

$$Motor\ Rotation = \frac{Actual\ Position\ (6064h:0)}{Feedback\ Resolution\ (3004h:2)} = \frac{386007}{16384} = 23{,}56\ rounds$$

The wheel will rotate of

$$\frac{Motor\ Rotation}{Gear\ Ratio\ \emptyset} = \frac{23{,}56}{21} = 1{,}122\ turns$$

The result of operation is equals 1 with a remainder of 0,122.

Angle Calculation:

The turn of the wheel is mapped in 360 degrees. When the "Position Actual Value" (6064_h) object is 386007 counts then the motor shaft will rotate of 23,56 rounds. The wheel will rotate 1,122 turns.

The angle of the motor wheel will be

$$Angle = \left(\left[\frac{Motor\ Rotation}{Feedback\ Resolution\ (3004h:2)}\right] div\ (Gear\ Ratio\ \emptyset)\right) x\ 360 = 0{,}122\ x\ 360 = 43{,}89\ degrees$$

Where 0,122 is the remainder of division $\left(\left(\frac{386007}{16384}\right)/21\ \right)$

REVISION HISTORY

Rel.	Date	Description
0.0	23/10/2019	• Draft of CANOpen User Guide for AGV
0.1	06/11/2019	• 2nd Draft of CANOpen User Guide for AGV
0.2	03/01/2020	• 3th Draft of CANOpen User Guide for AGV
0.3	07/01/2020	• Changes in table - I/O SIGNAL AGV
0.4	16/01/2020	• Added Object - 0x1008: Manufacturer Device Name - 0x1009: Manufacturer Hardware Version - 0x100A: Manufacturer Software Version - 0x3002: Brake Parameters - 0x3007: Dynamic Brake Parameters • Modified Object - 0x3020: Drive Digital Input - 0x60FD: Digital Inputs • Added Table Identifier • Added Example Programs in Appendix • Added Map Object Dictionary Memory • Added Mapping Default PDO • Added Store and Restore
0.5	17/01/2020	• Added Object - 0x3008: Emergency Enable Input Parameters
0.6	06/02/2020	• Added Object - 0x100C: Guard Time - 0x100D: Life Time Factor - 0x1017: Producer Heartbeat Time - 0x2003: Warning • Modified Object: - 0x6040: Control word – Bit Warning - 0x6041: Status Word – Bit Warning • Modified Node Guarding Protocol • Modified Heart-Beat Protocol • Update Error List CANopen - Add Communication Canopen: Error Code 0x8100
0.7	04/03/2020	• Added Object - 0x2041: Voltage Bus - 0x2050: Torque Current - 0x2053: Velocity Filtered - 0x1001: Error Register - 0x1003: Pre-defined Error Field • Object 0x3024 erased because It is in 0x3008 (Emergency Enable Parameters)

		• Modified Error List CANopen (Chapter Emergency messages) • Added Example Programs: • Emergency History
0.8	06/04/2020	• Added Object: - 0x1002:Manufacturer Status Register - 0x3050: Analog Output 1 - 0x3200: Current Pid - 0x3201: Speed Pid - 0x3202: Position Pid • Update Object: - 0x1010: Store parameters - 0x1011: Restore default parameters (page - 0x2003: Warning • Added SDO Abort Protocol • Update Error List CANopen - Data Set Param: add error code 0x6309, 0x630A, 0x630B - Resolver Fault Error: add error code 0x7370, 0x7373, 0x7374, 0x7375, 0x7376, 0x7377 - Warning: add error code 0x6001, 0x8B01, 0x8B06 • Update Store and Restore Chapter because changed e2prom
0.9	19/06/2020	• Update Object - 0x2003: Warning • Update Error List CANopen: - Golden Data Image: add error code 0x5A01, 0x5A02 - Can Protocol Communication: add error code 0x7530, 0x7531,0x7532 - Incremental Encoder Error: add error code 0x7390, 0x7391, 0x7392, 0x7393, 0x7394 - None Error Profile: add error code 0x8C04 - Hardware Error : 0x5501 • Added Graphic Velocity Profile Mode • Add Object 0x3001: Limits Parameters
1.0	13/07/2020	• Update Error List CANopen: - Hardware Error : 0x5501 - Error Parameters: 0x6321 • Update Object - 0x2002: Drive Mode change in Drive Control State - 0x3020: Drive Digital Input • Add Object - 0x3006: Max Motor Speed - 0x6402: Motor Type • Update Diagnostic Led • ERROR STATE MACHINE transition T12 • Add SAFETY Chapter

1.1	03/09/2020	• Update Error List CANopen: - Init Object CANopen from E^2prom: 0x8B06 - Data record no. 14: 0x630E - Data record no. 15: 0x630F • Update Object - 3002_h: Motor Brake Parameters - 3007_h: Dynamic Brake Parameters • Add Object - 3050_h : Analog Output 1 - $603F_h$: Error Code - 1018_h: Identity object • Add Abort Code for the following object: - 0X2000, 0X2001, 0X3001, 0x3002, 0x3008, 0x3050, 0x3200, 0x3201, 0x3202, 0x3202, 0x3300, 0x3004, 0x6060, 0x607E, 0x60FF, 0x607F, 0x6080, 0x6083, 0x6084, 0x60C5, 0x60C6, 0x606D, 0x606E, 0x6070 • Modify unit measure of objects: 0x2030, 0x2031, 0x2032, 0x2041 • Added Appendix : "FIRST CONFIGURATION" • Modified the appendix "Example Program" • Modified Chapter : "FUNCTIONS"
1.2	21/10/2020	• Modify unit measure of objects: 0x2041, 0x2050 • Modify Value Range Factory Group • Modify access of Object 6060h: rw (not ro) • Add digital Input 4 management • Add Appendix Heartbeat mechanism • Modify Network Management (NMT) Chapter • Add Graph Time frame Brake when the drive moves from "operation enabled" to "switched-on"
1.3	27/11/2020	• Add Chapter Torque Profile • SYNC Protocol • Update Error List CANopen: - Torque Profile: 0x9341, 0x8351 - Error EEprom Programming : 0x5563 … 0x556D • Modify PDO Protocol Chapter (add dynamic mapping features)
1.4	05/02/2021	• Update Error List CANopen: - Data Set Programming: 0x6401, 0x6402, 0x6403 - Load Level (i2t): 0x2352 - Warning Temperature : 0x4301, 0x4501, 0x4A01 - Init Object From EEprom : 0x8B10, 0x8B11, 0x8B12, 0x8B13, 0x8B14, 0x8B15, 0x8B16, 0x8B17, 0x8B18, 0x8B19, 0x8B1A, 0x8B1B, 0x8B1C, 0x8B1D, 0x8B1E, 0x8B1F, 0x8B20, 0x8B21, 0x8B22, 0x8B23, 0x8B24, 0x8B25, 0x8B26 • Modify object 0x3050 "Analog Output 1" : add function "Current Monitoring" • Add subindex 4 e 5 in "Motor Specific Settings" object 0x3006 • Modify object "Warning" 0x2003 - Add bit 8,9,10 warning temperature - Add bit 11 warning communication busoff/passive

		- Add bit 12, 13 Limitation Torque Limit • Modify Object 0x3020 "Digital Input Function" • Add Object for read the digital Input configuration: - Digital Input 1: 0x3021 - Digital Input 2: 0x3022 - Digital Input 3: 0x3023 - Digital Input 4: 0x3024 - Positive torque limit value: 0x60E0 - Negative torque limit value: 0x60E0 • Modify chapter digital Input in Function • Add Chapter "APPENDIX – POSITION MONITORING" • Add Led Code Status in "Diagnostic" : "Communication Error" • Modify Chapter "CAN Error Communication"