CANopen
Communication Profile

Servo DriveTTA-CAN
CANopen Drive

WARNING

This is a general manual describing the CANopen Communication Profile of the TRANSTECHNIK **TTA-CAN** drive (hardware release version 2.0 and firmware release from version 11.E3). For the installation and commissioning of the drive, please refer to the appropriate manuals (**TTA-CAN Installation Guide** and **TTA-CAN User Guide**).

Instructions for storage, use after storage, commissioning as well as all technical details require the MANDATORY reading of the manual before getting the amplifiers operational.

Maintenance procedures should be attempted only by highly skilled technicians having good knowledge of electronics and servo systems with variable speed (EN 60204-1 standard) and using proper test equipment.

The compliance with the standards and the "CE" approval is only valid if the items are installed according to the recommendations of the amplifier manuals. Connections are the user's responsibility if recommendations and drawings requirements are not met.

Any contact with electrical parts, even after power down, may involve severe physical damage.
Wait for at least 5 minutes after power down before handling the amplifiers (a residual voltage of several hundreds of volts may remain during a few minutes).

ESD INFORMATION (ElectroStatic Discharge)
TRANSTECHNIK amplifiers are conceived to be best protected against electrostatic discharges. However, some components are particularly sensitive and may be damaged if the amplifiers are not properly stored and handled.

STORAGE

- The amplifiers must be stored in their original package.
- When taken out of their package, they must be stored positioned on one of their flat metal surfaces and on a dissipating or electrostatically neutral support.
- Avoid any contact between the amplifier connectors and material with electrostatic potential (plastic film, polyester, carpet…).

HANDLING

- If no protection equipment is available (dissipating shoes or bracelets), the amplifiers must be handled via their metal housing.
- Never get in contact with the connectors.

ELIMINATION
In order to comply with the 2002/96/EC directive of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE), all TRANSTECHNIK devices have got a sticker symbolizing a crossed-out wheel dustbin as shown in Appendix IV of the 2002/96/EC Directive.
This symbol indicates that TRANSTECHNIK devices must be eliminated by selective disposal and not with standard waste.

TRANSTECHNIK does not assume any responsibility for physical or material damage due to improper handling or wrong descriptions of the ordered items.
Any intervention on the items, which is not specified in the manual, will immediately cancel the warranty.

TRANSTECHNIK reserves the right to change any information contained in this manual without notice.

Issue: **2.7**

Content

Chapter 1 – Overview

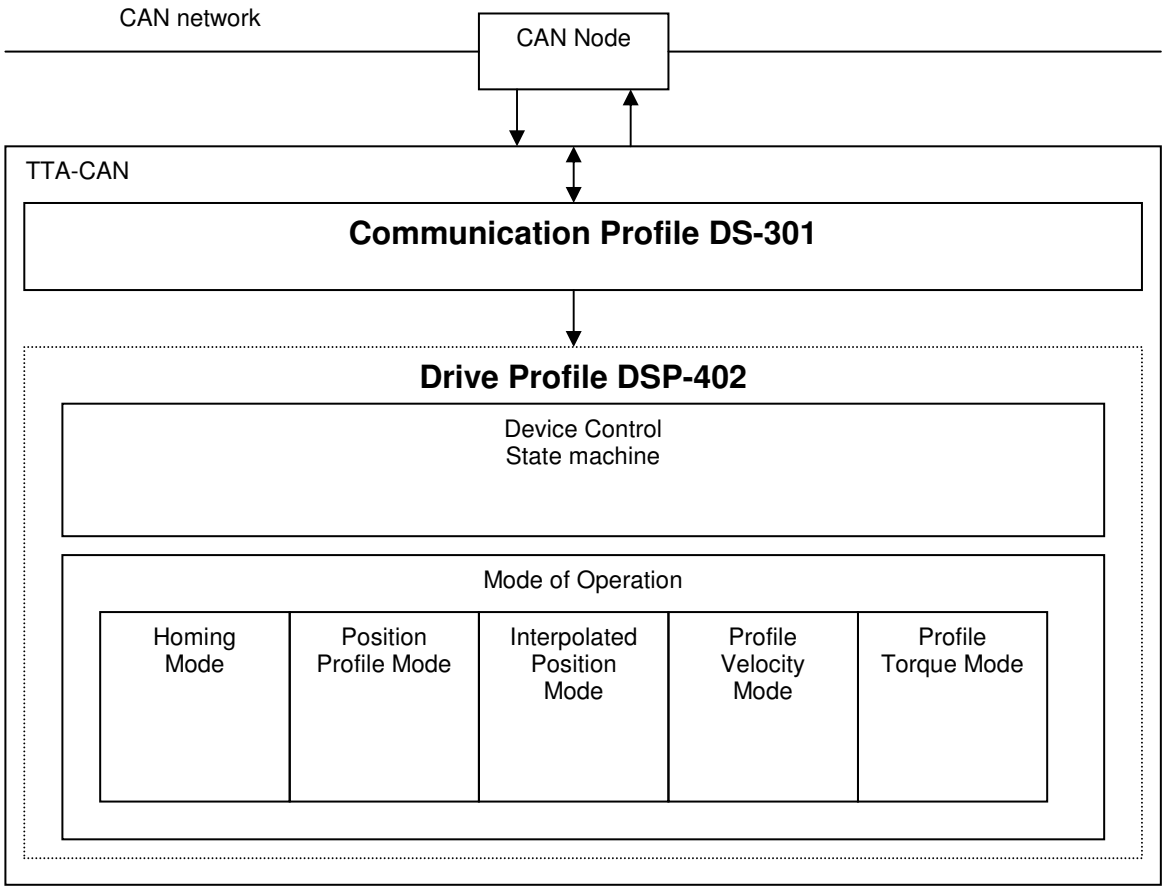
1.1 - REFERENCE

CiA DS-201..207	CAN Application Layer for Industrial Applications Version 1.1
CiA DS-301	Application Layer and Communication Profile Version 4.01
CiA DSP-402	Device Profile: Drive and Motion Control Version 1.1

1.2 - DEFINITIONS

CAN	Controller Area Network
CiA	CAN in Automation e. V. CAN-Bus international manufacturer and user organisation.
CAL	CAN Application Layer. The Application layer for CAN as specified by CiA.
COB	Communication Object is a CAN message. Data must be sent accross a CAN network inside a COB.
COB-ID	COB-Identifier. Each CAN message has a single identifier. There are 2032 different identifiers in a CAN network.
NMT	Network Management. One of the services of the application layer. It performs initialisation, configuration and error handling in a CAN network.
PDO	Process Data Object.
SDO	Service Data Object.
pp	Profile Position Mode
pv	Profile Velocity Mode
hm	Homing Mode
ip	Interpolated Position Mode
tq	Profile Torque Mode
pc	Position Control Function
TTA-CAN	TRANSTECHNIK TTA-CAN range servo drive with resolver and encoder feedback input

1.3 - ARCHITECTURE



Device Control: The drive starting and stopping are executed by the state machine.

The state machine - handled with a control word and a status word via CAN bus - gives the possibility to control the state of the drive: servo on/off, start/stop movement, fault reset...

Operation modes: The operation mode defines the behaviour of the drive :

Homing Mode: In this mode, various methods can be used to find a home position.

Profile Position Mode: In this mode, a trapezoidal trajectory generator gives the drive the possibility to execute a positioning with pre-set parameters such as target position, profile speed and acceleration.

Interpolated Position Mode: In this mode, the drive interpolates the continuous position reference from a coordinator controller. This mode provides the possibility for multiaxis synchronisation.

Profile Velocity Mode: This is the simple speed mode with speed reference from a host controller.

Profile Torque Mode: Only current loops are closed in this mode.

By default, **TTA-CAN** drives are working in Interpolated Position Mode.

Chapter 2 – Communication Profile

2.1 - COMMUNICATION OBJECTS

2.1.1 - CAN TELEGRAM

SOM	COB-ID	RTR	CTRL	Data segment	CRC	ACK	EOM

SOM : Start Of Message

COB-ID : COB-Identifier of 11 bits

RTR : Remote Transmission Request

CTRL: Control field

Data : up to 8 bytes

CRC : Cyclic Redundancy Check

ACK : Acknowledge

EOM : End Of Message

2.1.2 - DEFAULT COB-ID

The COB-ID is of 11 bits. Node-ID (bits 0 - 6) is the drive address from 1 to 127.

10	9	8	7	6	5	4	3	2	1	0
Function Code				NODE-ID						

Default COB-ID for the TTA-CAN:

Broadcast objects of the pre-defined connection set:

Object	Function Code	Resulting COB-ID	Communication Parameter at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h

Peer-to-peer objects of the pre-defined connection set:

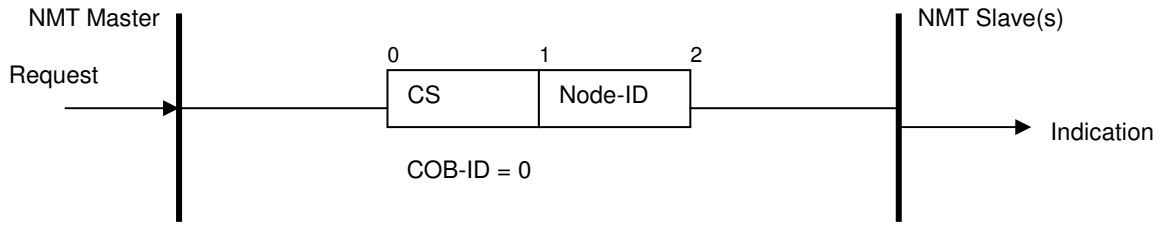
Object	Function Code	Resulting COB-ID	Communication Parameter at Index
EMERGENCY	0001	129 (81h) - 255 (FFh)	1014h
PDO1 (TX)	0011	385 (181h) - 511 (1FFh)	1800h
PDO1 (RX)	0100	513 (201h) - 639 (27Fh)	1400h
PDO2 (TX)	0101	641 (281h) - 767 (2FFh)	1801h
PDO2 (RX)	0110	769 (301h) - 895 (37Fh)	1401h
PDO3 (TX)	0111	897 (381h) - 1023 (3FFh)	1802h
PDO3 (RX)	1000	1025 (401h) - 1151 (47Fh)	1402h
SDO (TX)	1011	1409 (581h) - 1535 (5FFh)	1200h
SDO (RX)	1100	1537 (601h) - 1663 (67Fh)	1200h

TX = Transmit from drive to master

RX = Receive by drive from master

2.1.3 - NETWORK MANAGEMENT OBJECTS (NMT)

NMT Protocols

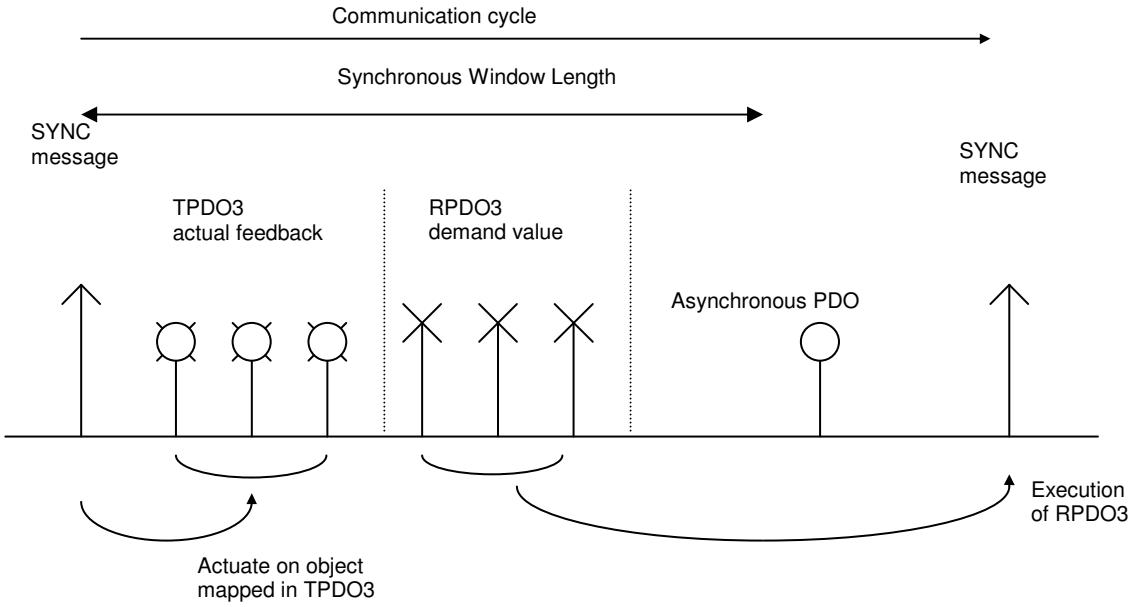


NMT Protocol	Command Specifier CS	Remarks
Start Remote Node	1	Change to NMT Operational state
Stop Remote Node	2	Change to NMT Stop state
Enter Pre-Operational	128	
Reset Node	129	
Reset Communication	130	

Node-ID: The Node-ID indicates the address of the **TTA-CAN** (defined by switches). If Node_ID = 0, the protocol addresses all NMT slaves.

2.1.4 - SYNCHRONISATION OBJECT (SYNC)

The SYNC object is a broadcast message sent by the master. This message provides a network clock. The period is specified by the communication cycle period (object 1006h). The TRANSTECHNIK servo-drives use this SYNC message to synchronize their clock.
At least 180 ms are necessary for the servo-drive to start the synchronisation.



2.1.5 - PROCESS DATA OBJECT (PDO)

PDOs are unconfirmed messages used for real-time data exchange.
PDOs sent by the master are RPDOs and PDOs sent by the drive are TPDOs.
Data in each PDO are defined by a list of objects (PDO mapping).

TRANSTECHNIK servo drives have TPDO1, RPDO1, TPDO2, RPDO2, TPDO3 and RPDO3.

Each PDO is defined by:

- PDO communication parameters at object 1400h + RPDO_number-1 (for RPDO) or 1800h + TPDO_number-1 (for TPDO)
- PDO mapping at object 1600h + RPDO_number-1 (for RPDO) or 1A00h + TPDO_number-1 (for TPDO)

2.1.5.1 - Communication parameters

Communication parameters are:
- PDO COB-ID
- Transmission type

The distribution of COB-ID is defined by default (see section 2.1.2). The modification of COB-ID of PDO can be made in *NMT Pre-Operational State*; the new COB-ID will take effect when the NMT state machine changes to *Operation State*. The modification must not be taken in *NMT Operational State*, otherwise a Reset_Communication will be necessary before the new COB-ID takes effect.

Transmission type supported by TRANSTECHNIK Servo Drives:

Transmission type	**PDO transmission**				
	cyclic	**acyclic**	**synchronous**	**asynchronous**	**RTR only**
1	TPDO1 TPDO2 TPDO3				
2-240	TPDO3				
253				TPDO1 TPDO2 TPDO3	TPDO1 TPDO2 TPDO3
254			TPDO3		
255				TPDO1	

- Transmission types 1 - 240 are synchronous transmissions with regard to the SYNC messages. A value between 1 and 240 means that the PDO is synchronously and cyclically transferred. The transmission type indicates the number of SYNC which is necessary to trigger PDO transmissions.
- Transmission type 253 means that the PDO is only transmitted on remote transmission request.
- Transmission type 255 is event triggered. The **TTA-CAN** supports 2 events: change of status word and change of Capture status.
- Transmission type 254 is specific to **TTA-CAN** (described in Chapter 3.13.8: PDO3 control).

PDO transmission modes of:
- *Synchronous*: the message is transmitted in synchronisation with the SYNC message. A synchronous message must be transmitted within a pre-defined time-window immediately after the SYNC message.
- *Asynchronous*: the message is sent independently of the SYNC message.

Triggering modes:
- *Event _Driven*:
Message transmission by reception of SYNC. In the TRANSTECHNIK Servo Drives, TPDO3 is used for this purpose.
Message transmission by specific event. In the TRANSTECHNIK Servo Drives, TPDO1 is used for this purpose.
- *Remotely requested*: the transmission of an asynchronous PDO is initiated on reception of a remote request by any other device.

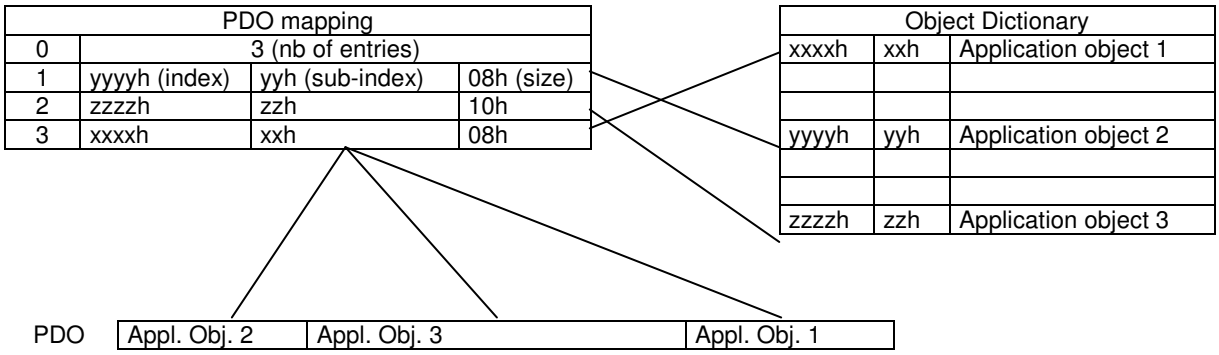
2.1.5.2 - PDO Mapping

The sub-index 0 of the mapping parameter contains the number of valid entries within the mapping record. This number of entries is also the number of application variables which shall be transmitted/received with the corresponding PDO. The sub-index 1 to number of entries contains the information about the mapped application variables. These entries describe the PDO contents by their index, sub-index and length (in bits).

Structure of PDO Mapping Entry:

Byte :	MSB		LSB
	index (16 bit)	sub-index (8 bit)	object length (8 bit)

Principle of PDO mapping:



2.1.5.3 - Multiplexed data

The multiplexed data is used to multiplex more than one axis demand value into one message RPDO3.
It is possible to send 4 axis demand values (16 bit absolute) with one RPDO3. Therefore, the controller must modify the COB-ID of RPDO3 of each axis to the same cob-ID. For example, for axis 1, object 60C1-1 is mapped into the first mapped object (object 1602-1), for axis 2, object 60C1-1 is mapped into the 2nd mapped object (object 1602-2) and so on... For each axis, the balance of the mapped objects must be mapped with a dummy object.

A dummy object mapped is realized with objects 0002h (integer8), 0003h (integer16), 0004h (integer32), 0005h (unsigned8), 0006h (unsigned16) or 0007h (unsigned32). These objects can be used to map a PDO as a dummy object but cannot be accessed via SDO (see DS-301, 9.5.3 Data type entry specification).

2.1.6 - SERVICE DATA OBJECT (SDO)

The SDO is a communication channel with 2 basic characteristics:
- Client/Server relationship,
- Object Dictionary.

Client/Server:
This is a relationship between a single client and a single server (TRANSTECHNIK Servo Drive). A client issues a request (upload/download) thus triggering the server to perform a certain task. After finishing the task, the server answers the request.

Object Dictionary:
All the objects (variables, constants, records...) of the server are defined as a list of objects where each element is appointed by an index and a sub-index. This list of objects is called object dictionary. This object dictionary allows the client the access to all objects of the server. The TRANSTECHNIK Servo Drive object dictionary consists of 2 parts: the communication profile (DS-301) for the objects related to the CAN communication and the device profile (DSP-402) for objects related to the drive functionality.

For more information about the SDO protocol, please report to the CiA DS-301 version 4.01 specification.

2.1.7 - EMERGENCY OBJECT (EMCY)

The Emergency telegram consists of 8 bytes:

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code		Error register (object 1001h)	Manufacturer Specific Error Field (object 3020h)				
				LSB			MSB	

2.1.7.1 – TTA-CAN Specific Error Field

Bit	Fault	treatment / Cycle	Stop	Remarks
0	Position Limits	Software / 500 µs	0, 1, 3	
1	I^2t	Software / 2 ms	0, 1, 3, warning	
2	RDC / Counting	Software / 500 µs	0	
3	Following error	Software / 500 µs	0, 3	
4	EEPROM	Software		
5	Synchro_CAN	Software / cycle	0, 1, 3	
6	LowSpeed	Software / 500 µs	3, 0, 1	
7	Procedure	Software	0	
8	Current_Offset	Software		Not erasable
13	Motor Temperature	Software / 500 µs	0, 1, 3, warning	
14	Init_400V	Software		Not erasable
15	Overcurrent	Software / 62.5 µs	0, 1, 3	
16	power	Hardware	0	see below
17	power	Hardware	0	see below
18	power	Hardware	0	see below
19	IGBT	Hardware	0	s.c. ASIPM / PWM Error / Alim ASIPM / T° ASIPM / Over I
20	Power Supply Undervoltage	Hardware	0	not stored
21	Resolver / Encoder Cable Interruption	Hardware	0	
24	Second sensor Cable Interruption	Hardware	0	
25	Hall Effect Sensors / Com. Channel (for absolute encoders)	Hardware	0	

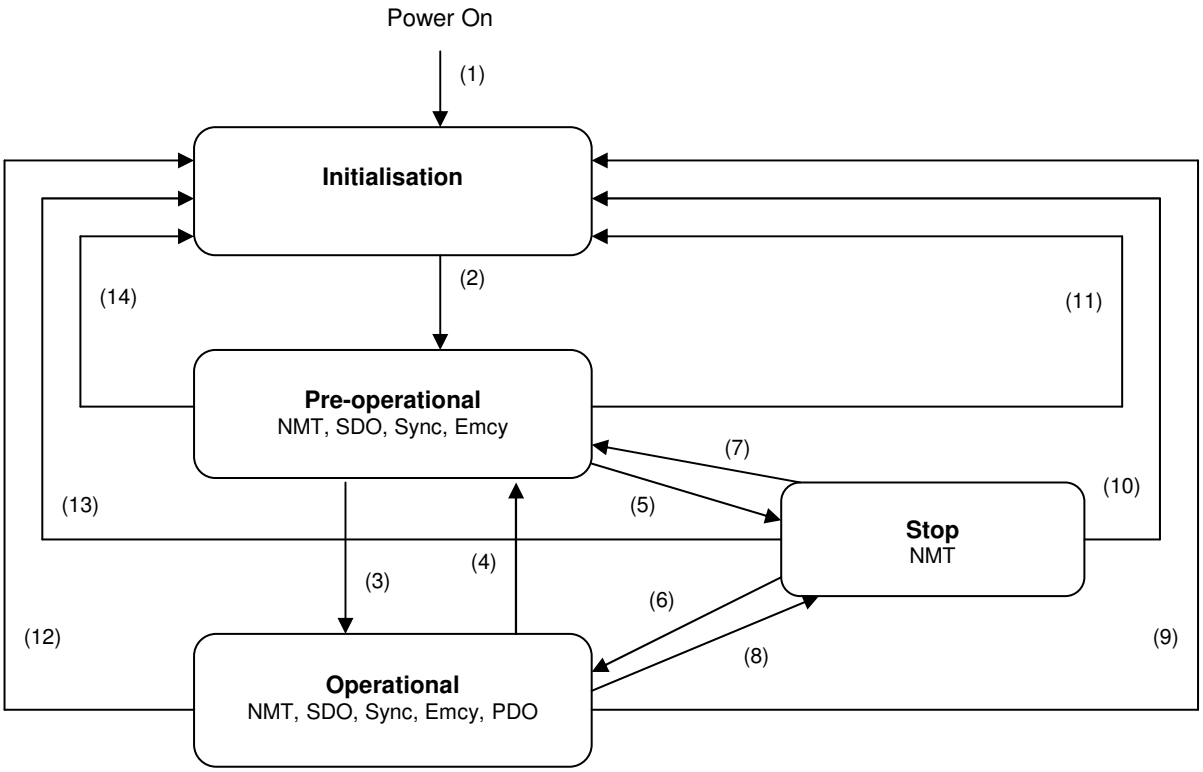
The power faults are coded with bits 16, 17 and 18. The meaning is:

Bit			Meaning
18	17	16	
0	1	0	Overvoltage error
0	1	1	24 V error (18 V to 29 V)
1	0	0	Short-circuit motor phases/ground
1	0	1	System braking resistor error (short-circuit or overload)
1	1	0	Fan system error
1	1	1	Motor brake error

2.2 - NETWORK INITIALISATION AND SYSTEM BOOT-UP

2.2.1 - NMT STATE MACHINE

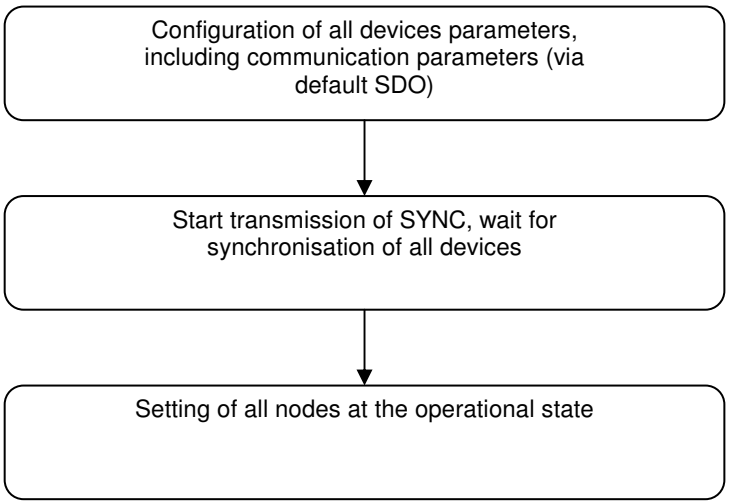
The NMT state machine defines the communication status.



(1)	At Power on, the initialisation state is automatically entered
(2)	Once the Initialisation over, Pre-Operational is automatically entered
(3), (6)	Start_Remote_Node indication
(4), (7)	Enter_Pre-Operational_State indication
(5), (8)	Stop_Remote_Node indication
(9), (10), (11)	Reset_Node indication
(12), (13), (14)	Reset_Communication indication

Minimum Boot-Up consists of one CAN telegram : a broadcast Start_Remote_Note message.

2.2.2 - INITIALISATION PROCEDURE



2.3 - OBJECT DICTIONARY

2.3.1 - GENERAL INFORMATION

2.3.1.1 - Object 1000h: Device Type

This object describes the type of a device.

Index	1000h
Name	Device Type
Object Code	VAR
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Default Value	00020192h

Device Profile Number : 402
Additional Information : 02 Servo Drive

2.3.1.2 - Object 1001h: Error Register

Index	1001h
Name	Error Register
Object Code	VAR
Data Type	Unsigned8
Access	ro
PDO Mapping	No
Default Value	0

Bit Number	Description
0	Generic Error

2.3.1.3 - Object 1008h: Manufacturer Device Name

Index	1008h
Name	Manufacturer Device Name
Object Code	VAR
Data Type	Visible String
Access	ro
PDO Mapping	No
Default Value	'TTA-CAN' for TTA-CAN drive.

2.3.1.4 - Object 1009h: Manufacturer hardware version

Index	1009h
Name	Manufacturer Hardware Version
Object Code	VAR
Data Type	Visible String
Access	ro
PDO Mapping	No

2.3.1.5 - Object 100Ah: Manufacturer software version

Index	100Ah
Name	Manufacturer Software Version
Object Code	VAR
Data Type	Visible String
Access	ro
PDO Mapping	No

2.3.1.6 - Object 100Ch: Guard Time

The guard time (in ms) multiplied by the life time factor gives the life time for Life Guarding Protocol. It is 0 if not used.

Index	100Ch
Name	Guard time
Object Code	VAR
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Default Value	0

2.3.1.7 - Object 100Dh: Life Time factor

Index	100Dh
Name	Life Time Factor
Object Code	VAR
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	0

2.3.1.8 - Object 1010h: Store Parameters

This object supports the saving of parameters in a non volatile memory. By read access, the device provides information about its saving capacities.
Sub-index 0: largest sub-index that is supported.
Sub-index 1: refers to all parameters that can be stored in the device.

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'.

Signature	MSB			LSB
ASCII	e	v	a	s
hex	65h	76h	61h	73h

On read access to the appropriate sub-index, the device provides information about its storage capabilities:

bit	value	meaning
31-2	0	reserved (=0)
1	0 1	Device does not save parameters autonomously Device saves parameters autonomously
0	0 1	Device does not save parameters on demand Device saves parameters on demand

Regarding the TRANSTECHNIK TTA-CAN servo drive, parameters are saved on demand, when the drive is disabled.

2.3.1.9 - Object 1018h: Identity Object

Index	1018h
Name	Identity Object
Object Code	RECORD
Number of Elements	1

Value Description

Sub Index	1
Description	Vendor ID
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Default Value	00000082h

2.3.2 - SYNC OBJECT

2.3.2.1 - Object 1005h: COB-ID Sync message

Index	1005h
Name	COB-ID Sync message
Object Code	VAR
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	00000080h

This object defines the COB-ID of the synchronisation object (SYNC).
The new COB-ID takes immediately effect if it is not changed in NMT Operation State. Otherwise, a *reset_communication* message is necessary.

2.3.2.2 - Object 1006h: Communication Cycle Period

Index	1006h
Name	Communication Cycle Period
Object Code	VAR
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Unit	µs
Value Range	1000..20000 (only the values multiples of 500 are correct for the TTA-CAN)
Default Value	eeprom

This object defines the communication cycle. This period is also used for the synchronisation in interpolated position mode. When the value of this object is reset at 0, the synchronisation is no more operative.

2.3.2.3 - Object 3006h: Communication Cycle Period adjustment

Index	**3006h**
Name	Communication cycle period adjustment
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	Cycle period adjustment limit
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Unit	us
Value Range	1.5us (0xF) to 20us (0xC8)
Default Value	1.5 us (0xF)
Conversion	0.1us
Write Condition	Drive disabled.

Sub Index	2
Description	Cycle period adjustment parameter 2
Data Type	Unsigned16
Default Value	0x3C
Remark	Reserved for future use. Not to be modified.

Communication cycle period adjustment :
- The SYNC cycle time T is set with object 1006 sub index 0.
- The maximum permissible jitter is T/2.
- The accuracy of the SYNC period must be better than 0.3%.

If the last condition cannot be fulfilled because the average value of the real communication cycle period given by the host controller differs too much from the theoretical value defined in the object 1006 sub index 0, the object 3006 sub index 1 allows to modify the "Cycle period adjustment limit ".

SYNC period error limit (%) = ABS(Treal - Ttheoretical) / Ttheoretical
Cycle period adjustment limit (us) = SYNC period error limit (%) x 500 (500us is the Tick of the TTA-CAN)
Object 3006 sub index 1 value = Cycle period adjustment limit (us) x 10 (conversion factor 0.1 us)

If, for example, the SYNC period error limit is 2 %, Cycle period adjustment limit (µs) = 2% x 500 µs = 10 µs and object 3006 sub index 1 value = 10 x 10 = 100 = 0x64

This object is not saved in the TTA-CAN EEPROM, so its value must be sent at each amplifier power up.

2.3.2.4 - Object 1007h: Synchronous Window Length

Index	1007h
Name	Synchronous Window Length
Object Code	VAR
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Unit	µs
Value Range	1500..20000
Default Value	

This object defines the synchronous window in which the drive can latch the value from synchronous PDO (RPDO3). Generally, it can be set at a value equal to (object 1006h) - 10000

2.3.3 - SDO OBJECTS

2.3.3.1 - Object 1200h: Server SDO Parameter

Index	1200h
Name	Server SDO parameter
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	COB-ID Client -> Server (rx)
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Default Value	600h + Node-ID

Sub Index	2
Description	COB-ID Client -> Server (rx)
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Default Value	580h + Node-ID

For the **TTA-CAN** it is not possible to modify the COB-ID of the SDO objects.

2.3.4 - EMERGENCY OBJECT

2.3.4.1 - Object 1014h: COB-ID Emergency message

Index	1014h
Name	COB-ID Emergency message
Object Code	VAR
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	80h + Node-ID

2.3.5 - PDO OBJECTS

2.3.5.1 - Object 1400h: 1st Receive PDO Communication Parameter

Index	**1400h**
Name	1st Receive PDO Communication Parameter (RPDO1)
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	COB-ID
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	200h + Node-ID

Sub Index	2
Description	Transmission Type
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	253

2.3.5.2 - Object 1401h: 2nd Receive PDO Communication Parameter

Index	**1401h**
Name	2nd Receive PDO Communication Parameter (RPDO2)
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	COB-ID
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	300h + Node-ID

Sub Index	2
Description	Transmission Type
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	253

2.3.5.3 - Object 1402h: 3rd Receive PDO Parameter

Index	**1402h**
Name	3rd Receive PDO Communication Parameter (RPDO3)
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	COB-ID
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	400h + Node-ID

Sub Index	2
Description	Transmission Type
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

2.3.5.4 - Object 1600h: 1st Receive PDO Mapping

Index	**1600h**
Name	1st Receive PDO Mapping
Object Code	RECORD
Number of Elements	0..4

Value Description

Sub Index	0
Description	number of mapped objects
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

Sub Index	1
Description	1st mapped object
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	60400010h (control word)

2.3.5.5 - Object 1601h: 2nd Receive PDO Mapping

Index	**1601h**
Name	2nd Receive PDO Mapping
Object Code	RECORD
Number of Elements	0..4

Value Description

Sub Index	0
Description	number of mapped objects
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

Sub Index	1
Description	1st mapped object
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	60FF0020h (target velocity)

2.3.5.6 - Object 1602h: 3rd Receive PDO Mapping

Index	**1602h**
Name	3rd Receive PDO Mapping
Object Code	RECORD
Number of Elements	0..4

Value Description

Sub Index	0
Description	number of mapped objects
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

Sub Index	1
Description	3rd mapped object
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	60C10120h (Interpolated data record)

2.3.5.7 - Object 1800h: 1st Transmit PDO Parameter

Index	**1800h**
Name	1st Transmit PDO Communication Parameter (TPDO1)
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	COB-ID
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	180h + Node-ID

Sub Index	2
Description	Transmission Type
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	253

2.3.5.8 - Object 1801h: 2nd Transmit PDO Parameter

Index	**1801h**
Name	2nd Transmit PDO Communication Parameter (TPDO2)
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	COB-ID
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	280h + Node-ID

Sub Index	2
Description	Transmission Type
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	253

2.3.5.9 - Object 1802h: 3rd Transmit PDO Parameter

Index	**1802h**
Name	3rd Transmit PDO Communication Parameter (TPDO3)
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	COB-ID
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	380h + Node-ID

Sub Index	2
Description	Transmission Type
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

2.3.5.10 - Object 1A00h: 1st Transmit PDO Mapping

Index	**1A00h**
Name	1st Transmit PDO Mapping
Object Code	RECORD
Number of Elements	0..4

Value Description

Sub Index	0
Description	number of mapped objects
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

Sub Index	1
Description	1st mapped object
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	60410010h (status word)

2.3.5.11 - Object 1A01h: 2nd Transmit PDO Mapping

Index	**1A01h**
Name	2nd Transmit PDO Mapping
Object Code	RECORD
Number of Elements	0..4

Value Description

Sub Index	0
Description	number of mapped objects
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

Sub Index	1
Description	1st mapped object
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	606C0020h (velocity value)

2.3.5.12 - Object 1A02h: 3rd Transmit PDO Mapping

Index	**1A02h**
Name	3rd Transmit PDO Mapping
Object Code	RECORD
Number of Elements	0..4

Value Description

Sub Index	0
Description	number of mapped objects
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Default Value	1

Sub Index	1
Description	1st mapped object
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	60640020h (Actual position value)

Chapter 3 – Device Profile

3.1 - PDO MAPPING

By default the TRANSTECHNIK Servo Drive runs in interpolated position mode and the default PDO mapping is for this mode.

3.1.1 - RECEIVE PDO

PDO No.	Mapping Object Index	Mapping Object Name	Comment
1	6040h	Control Word	Controls the state machine
2	60FFh	Target velocity	Controls the speed
3	60C1h, 1	Interpolation Data Record	Controls the position

The RPDO1and RPDO2 are acyclic while RPDO3 is cyclic and synchronous.

In interpolated position mode, the position reference is given in Interpolated data record according to DSP-402.

3.1.2 - TRANSMIT PDO

PDO No.	Mapping Object Index	Mapping Object Name	Comment
1	6041h	Status Word	shows status
2	606Ch	Velocity value	actual velocity
3	6064h	Actual Position	actual position

The TPDO1 and TPDO2 are remotely request while the TPDO3 is cyclic and synchronous.

3.2 - COMMON ENTRIES IN THE OBJECT DICTIONARY

3.2.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
6402	VAR	Motor Type	Unsigned16	ro
6410	RECORD	Motor Data	-	rw
6502	VAR	Supported Drive Modes	Unsigned32	ro
6504	VAR	Drive Manufacturer	String	ro
6510	RECORD	Drive Data	-	rw
60FD	VAR	Digital Inputs	Unsigned32	ro
60FE	RECORD	Digital Outputs	-	rw

3.2.2 - OBJECT DESCRIPTION

3.2.2.1 - Object 6402h: Motor Type

The motor type is PM Synchronous motor.

Index	6402h
Name	Motor Type
Object Code	VAR
Data Type	Unsigned16
Access	ro
PDO Mapping	No
Default Value	0003

3.2.2.2 - Object 6410h: Motor Data

This object contains information about the connected motor.
The parameters "Number of Motor Pole Pairs", "Motor Phase" and "Sensor Offset" are necessary for the current commutation. These parameters can be calculated by the auto-phasing procedure.

TTA-CAN Motor Data:

Index	**6410h**
Name	Motor Data
Object Code	RECORD
Number of Elements	6

Value Description

Sub Index	1
Description	Motor Id
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Value Range	0..(2^{14}-1)
Default Value	eeprom
Write Condition	Drive disabled.

Sub Index	2
Description	Number of motor poles pairs This parameter defines the number of motor pole pairs per revolution of a rotating motor.
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Unit	-
Value Range	1 to 32
Default Value	eeprom
Write Condition	Drive disabled.

Sub Index	3
Description	Phase order corresponds to the phase order (U, V, W) of the motor connection.
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Value Range	0..1
Default Value	eeprom
Write Condition	Drive disabled.

Sub Index	4
Description	Position sensor offset phase shift between position sensor and motor rotor.
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Default Value	eeprom
Write Condition	Drive disabled.

Sensor Offset (resolver feedback):

Bit number	**Description**
0-15	Value of the offset 0 to FFFFh (0 to 360° mechanical degree).

The electrical degree of the offset will be:

electrical offset = mechanical value / Number of pole pairs

Sensor Offset (encoder feedback):

Bit number	Description
0-14	Absolute value of the offset 0 to 7FFFh (0 to 360° electrical degree).
15	Sign of the offset (encoder wiring direct or reverse)

This parameter is only valid for "Absolute single-turn or multi-turn SIN/COS" encoder feedback configuration (see object 3080h).
This parameter value is depending on the encoder adjustment. There is only one value for a motor with a number of pole pairs > 1.

Sub Index	5
Description	Current Phase Lead Coefficient This parameter is not used in the TTA-CAN
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Default Value	0

Sub Index	6
Description	Hall Effect Sensors Offset
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Unit	-
Default Value	eeprom

HES Offset (encoder feedback):

Bit number	Description
0-14	Absolute value of the offset 0 to 7FFFh (0 to 360° electrical degree).
15	Sign of the offset (encoder wiring direct or reverse)

This parameter is only valid for "60° HES" or "120° HES" encoder feedback configuration (see object 3080h).
This parameter value is depending on the Hall effect sensors adjustment. There is only one value for a motor with a number of pole pairs > 1.

3.2.2.3 - Object 6502h: Supported Drive Modes

Index	6502h
Name	Supported Drive Modes
Object Code	VAR
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Value	0000006D

The TRANSTECHNIK Servo Drive can be used in following modes:

Bit Number	Description
0	Profile Position Mode
2	Profile Velocity Mode
3	Profile Torque Mode
5	Homing Mode
6	Interpolated Position Mode

3.2.2.4 - Object 6504h: Drive Manufacturer

Index	6504h
Name	Drive Manufacturer
Object Code	VAR
Data Type	Visible String
Access	ro
PDO Mapping	No
Value	'Infranor'

3.2.2.5 - Object 6510h: Drive Data

Index	6510h
Name	Drive Data
Object Code	RECORD
Number of Elements	7

Value Description

Sub Index	1
Description	Drive Family Code 02
Data Type	Unsigned16
Access	ro
PDO Mapping	No
Default Value	0002

Sub Index	2
Description	Drive Option Code
Data Type	Unsigned16
Access	ro
PDO Mapping	No
Default Value	0000

Sub Index	3
Description	Drive Voltage
Data Type	Unsigned16
Access	ro
PDO Mapping	No
Unit	V

Sub Index	4
Description	Drive Current Rating
Data Type	Unsigned16
Access	ro
PDO Mapping	No
Unit	0.01 A (the read value must be multiplied by 0.01 for getting the Drive Current Rating in A)

Sub Index	5
Description	Max Current
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Unit	Percentage of amplifier rating (3FFFh = 100%)
Value Range	0000h..3FFFh
Default Value	eeprom

Sub Index	6
Description	Rated Current
Data Type	Unsigned16
Access	rw
PDO Mapping	No
Unit	Percentage of amplifier rating (3FFFh = 100%)
Value Range	0000h..2000h
Default Value	eeprom

Sub Index	7
Description	I^2t Mode
Data Type	Unsigned8
Access	rw
PDO Mapping	No
Unit	-
Value Range	0=limiting, 1=fusing
Default Value	eeprom

3.2.2.6 - Object 60FDh: Digital Inputs

Index	**60FDh**
Name	Digital Inputs
Object Code	VAR
Data Type	Unsigned32
Access	ro
PDO Mapping	Possible
Value	-

Bit Number	**Description**
0	Negative Limit Switch
1	Positive Limit Switch
2	Home Switch
3	Inhibit
16	Positive Limit Switch Input / Digital Input 0
17	Negative Limit Switch Input / Digital Input 1
18	INHIBIT
19	INDEX / Digital Input 3
20	CAPTURE / Digital Input 4
21	LOW SPEED / Digital Input 5

Please see **TTA-CAN Installation** manual for the pin out of the physical inputs.

3.2.2.7 - Object 60FEh: Digital Outputs

Index	**60FEh**
Name	Digital Outputs
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	Physical Outputs
Data Type	Unsigned32
Access	rw
PDO Mapping	Possible
Default Value	00000000h

Sub Index	2
Description	Bitmask
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	00000000h

Bit Number	Description
0	Set brake
16	Digital Output 0
17	Digital Output 1
18	Digital Output 2
19	Digital Output 3

Refer to the TTA-CAN manual for the pin out of the physical outputs.

For enabling a digital output control, the corresponding bit of object 60FE sub-index 2 must be set at 1 (digital output bitmask). The digital output can then be controlled by the corresponding bit of object 60FE sub-index 1.

3.2.2.8 - Object 30FFh: Specific Outputs Configuration

Index	30FFh
Name	Specific Outputs Configuration
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0

Bit	Meaning
0	Amplifier error flag on Digital Output 0
1	Amplifier error flag on Digital Output 1
2	Amplifier error flag on Digital Output 2
3	Amplifier error flag on Digital Output 3
4	Digital cam 1 status on Digital Output 0
5	Digital cam 2 status on Digital Output 1
6	Digital cam 3 status on Digital Output 2
7	Digital cam 4 status on Digital Output 3
8 to 15	reserved

The specific “Amplifier error flag” output is set at 0 when any amplifier fault is triggered.
The "Digital cam" status is controlled by object 2500 and object 2501.

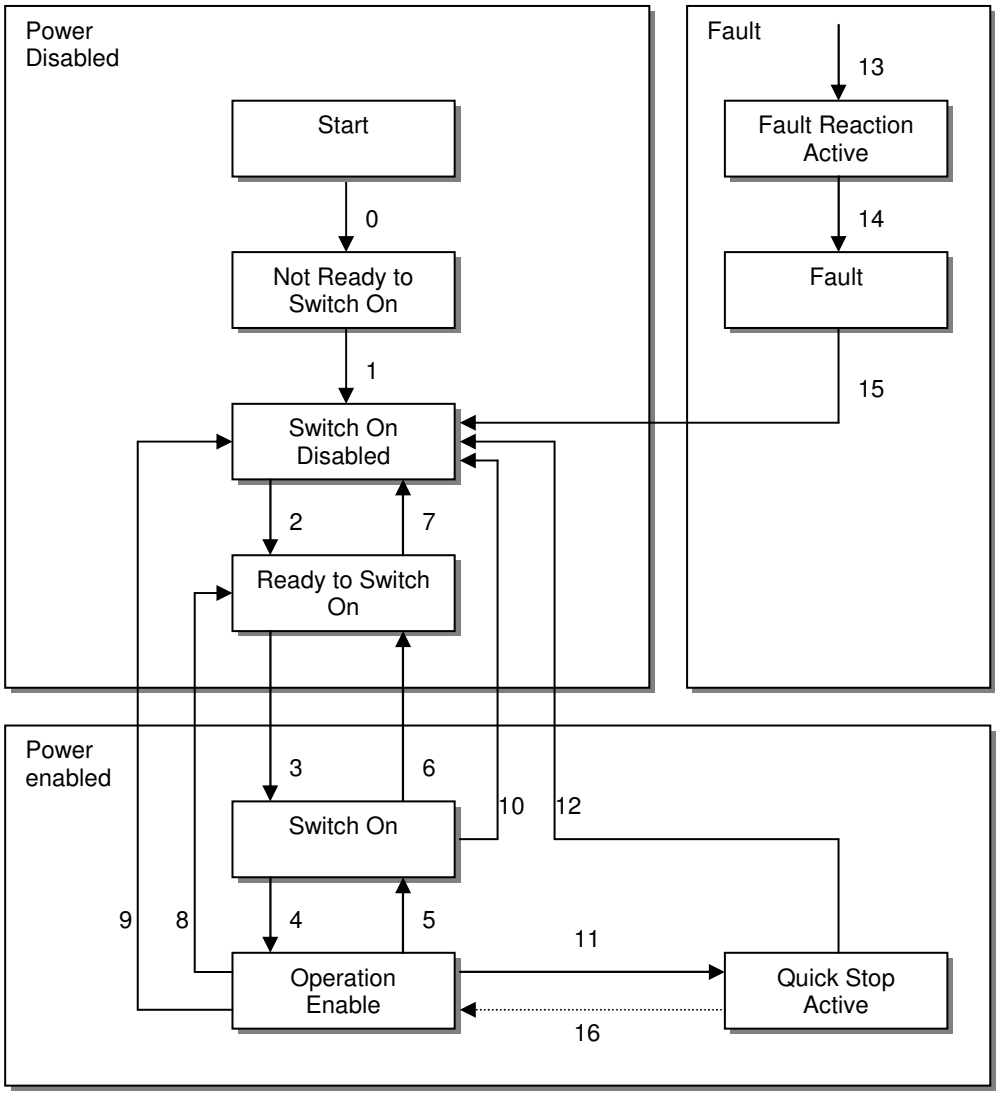
To enable a specific output on a given digital amplifier output, the corresponding bit of object 60FE sub-index 2 must be reset at 0 (digital output bitmask). The specific output can then be connected to this digital output by using the object 30FF (specific output configuration).

Example : Set the “Amplifier error flag” to the digital output 3.
- The bit 19 of object 60FE sub-index 2 must be reset at 0 (default value after power up)
- The bit 3 of object 30FF must be set at 1

3.3 - DEVICE CONTROL

3.3.1 - DRIVE STATE MACHINE

The state machine describes the status and the control sequence of the drive.



3.3.2 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
6040	VAR	Control Word	Unsigned16	rw
6041	VAR	Status Word	Unsigned16	ro
605A	VAR	Quick Stop Option Code	Integer16	rw
6060	VAR	Mode of Operation	Integer8	wo
6061	VAR	Mode of Operation Display	Integer8	ro

3.3.3 - OBJECT DESCRIPTION

3.3.3.1 - Object 6040h: Control Word

Index	6040h
Name	Control Word
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	Possible
Default Value	0000

Bit Number	Function
0	Switch On
1	Disable Voltage
2	Quick Stop
3	Enable Operation
4	Operation Mode Specific
5	Operation Mode Specific
6	Operation Mode Specific
7	Reset Fault (rising edge)
11	Out_Of_Limit (manufacturer specific)

Device control commands are triggered by the following bit patterns in the control word:

Command / Bit of the control_word	bit 7 Fault Reset	bit 3 Enable Operation	bit 2 Quick Stop	bit 1 Disable Voltage	bit 0 Switch On	Transition
Shutdown	X	X	1	1	0	2, 6, 8
Switch On	X	X	1	1	1	3
Disable Voltage	X	X	X	0	X	7, 9, 10, 12
Quick Stop	X	X	0	1	X	7, 10, 11
Disable Operation	X	0	1	1	1	5
Enable Operation	X	1	1	1	1	4, 16
Fault Reset	↑	X	X	X	X	15

Bit 4, 5, 6 are operation mode specific:

Bit	Profile Position Mode	Homing Mode	Interpolated Position Mode
4	new set point	Homing Operation Start	enable ip_mode
5	change_set_immediately	reserved	reserved
6	0: absolute 1: relative	reserved	reserved

Bit 11: Out_of_Limit. Manufacturer Specific.
When the motor is in limit position, this bit authorizes to move the motor.

3.3.3.2 - Object 6041h: Status Word

The status word indicates the current status of the drive. It is possible to define the TPDO1 to be transmitted at every change of status word (Device Event transmission type).

Index	6041h
Name	Status Word
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	ro
PDO Mapping	Possible
Default Value	-

Bit Number	Function
0	Ready to Switch On
1	Switch On
2	Operation Enabled
3	Fault
4	Voltage Disabled
5	Quick Stop
6	Switch On Disabled
7	Warning
9	Remote
10	Target Reached
12	Operation Mode Specific
13	Operation Mode Specific
14	Manufacturer Specific: Low Speed Mode
15	Manufacturer Specific: Drive Busy

Device Status Bit Meaning:

State	Bit 6 Switch On Disable	Bit 5 Quick Stop	Bit 3 Fault	Bit 2 Operation Enable	Bit 1 Switched On	Bit 0 Ready to Switch On
Not Ready to Switch On	0	X	0	0	0	0
Switch On Disabled	1	X	0	0	0	0
Ready to Switch On	0	1	0	0	0	1
Switched On	0	1	0	0	1	1
Operation Enable	0	1	0	1	1	1
Fault	0	X	1	1	1	1
Fault Reaction Active	0	X	1	1	1	1
Quick Stop Active	0	0	0	1	1	1

Bits 12, 13 are operation mode specific:

Bit	Profile Position Mode	Homing Mode	Interpolated Position Mode	Profile Velocity Mode
12	setpoint acknowledge	Homing attained	Ip-Mode active	Speed = 0
13	Following Error	Homing error	reserved	reserved

3.3.3.3 - Object 605Ah: Quick Stop Option Code

Index	605Ah
Name	Quick Stop Option Code
Object Code	VAR
Data Type	integer16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	3

Quick stop option code	Action
0	Disable drive
1	Slow down on speed ramp
3	Slow down on current limit

The slow down speed ramp is defined in object 3300h.
The slow down current limit is defined in object 3301h.

3.3.3.4. Object 6060h: Mode of Operation

This parameter changes the operation mode of the drive.

Index	6060h
Name	Mode of Operation
Object Code	VAR
Data Type	integer8
Object Class	all
Access	wo
PDO Mapping	Possible

Mode of Operation	Action
1	Profile Position Mode
3	Profile Velocity Mode
4	Torque Profile Mode
6	Homing Mode
7	Interpolated Position Mode

The actual mode is reflected in the operation mode display (object 6061h).

3.3.3.5. Object 6061h: Mode of Operation Display

Index	6061h
Name	Mode of Operation Display
Object Code	VAR
Data Type	integer8
Object Class	all
Access	ro
PDO Mapping	possible
Default Value	7

3.4 – POSITION SENSOR GROUP

3.4.1 - UNITS

The Units used in the TTA-CAN are the following:
Position: increment (inc)
Velocity: inc/s
Acceleration: inc/s²
Current: Percentage of amplifier rating (3FFFh = 100%)

3.4.2 - PRESENTATION

The TTA-CAN amplifier has got 2 position sensor inputs : one for a resolver and another one for an encoder. Various encoder types can be used (TTL, SinCos, incremental, absolute, single-turn, multi-turn). So, the TTA-CAN amplifiers can be configured for driving motors equipped with a resolver position feedback sensor or an encoder position feedback sensor. The Motor Feedback Selection object (3070) is set according to the motor position feedback sensor type (resolver or encoder). The Encoder Input Selection object (3080) is set according to the encoder sensor type (TTL, SIN/COS, incremental, absolute, single turn, multi turn).
The position sensor input which is not used for the motor position feedback (encoder or resolver) is called Second Sensor input. The Second Sensor input can be used for closing the drive position loop if a secondary position sensor is mounted on the motor load. The Second Sensor Feedback Selection object (306A) is set according to the position control loop structure: position feedback from the motor sensor or position feedback from the load sensor. The Second Position Sensor input can also be used for an electronic gearing application or for a stepper motor emulation application. The stepper motor emulation is only possible for a motor equipped with a resolver sensor.

3.4.3 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
607E	VAR	Polarity	Unsigned8	rw
608F	ARRAY	Position Resolution	Unsigned32	rw

Manufacturer specific object:

Index	Object	Name	Type	Attr.
3080	VAR	Encoder Input Selection	Unsigned16	rw
308F	VAR	Encoder Input Resolution	Unsigned32	rw
3090	VAR	Encoder Pulse Interpolation	Unsigned16	rw
3091	VAR	Encoder Zero Mark Pitch	Unsigned16	rw
3070	VAR	Motor Feedback Selection	Unsigned16	rw
3098	VAR	Motor Feedback Programming	Unsigned32	wo
306A	VAR	Second Sensor Feedback	Unsigned16	rw
306C	VAR	Second Sensor Scaling factor	Unsigned16	rw

3.4.4 - OBJECT DESCRIPTION

3.4.4.1 - Object 607Eh: Polarity

Index	607Eh
Name	Polarity
Object Code	VAR
Data Type	Unsigned8
Object Class	pc hm pp pv tq ip
Access	rw
PDO Mapping	No
Default Value	00

Bit Number	Description
7	Motor Position polarity 0 => multiply by 1 (default) 1 => multiply by -1 resolver or encoder
6	Motor Velocity polarity 0 => multiply by 1 (default) 1 => multiply by -1 resolver or encoder
5	2nd Position sensor polarity 0 => multiply by 1 (default) 1 => multiply by -1 encoder only

Position and velocity polarity bits are set or reset at the same time.

Remark: When "motor position polarity" is reversed, the limit switches wiring **must not** be modified. So, in this case, the negative limit switch is stopping the forward movement (increasing position value) and the positive limit switch is stopping the backward movement (decreasing position value).

3.4.4.2 - Object 608Fh: Position Resolution

Index	608Fh
Name	Position Resolution
Object Code	ARRAY
Number of Elements	2

Value Description

Sub Index	1
Description	Revolution Increments
Data Type	Unsigned32
Object Class	all
Access	rw
PDO Mapping	No
Unit	inc
Value Range	190h .. 3D0900h
Default Value	eeprom

This parameter defines the number of increments for one motor revolution. This parameter affects only the scaling of the position demand and the position feedback value, but neither the position resolution nor the position accuracy of the position sensors (resolver or encoder) .

Sub Index	2
Description	Motor Revolutions
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Default Value	1

3.4.4.3 - Object 3080h: Encoder Input Selection

Index	3080h
Name	Encoder Input Selection
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

Value	Description
0	TTL incremental encoder configuration
1	TTL incremental encoder + 60° HES configuration
2	TTL incremental encoder + 120° HES configuration
3	SIN/COS incremental encoder configuration
4	SIN/COS incremental encoder + 60° HES configuration
5	SIN/COS incremental encoder + 120° HES configuration
6	Absolute single-turn SIN/COS encoder (ERN1085 compatible)
7	Absolute multi-turn / single-turn SIN/COS encoder with ENDAT protocol
8	Absolute multi-turn / single-turn SIN/COS encoder with HIPERFACE protocol
9	Absolute linear SIN/COS encoder with HIPERFACE protocol
F	Pulse and Direction input configuration for stepper emulation

3.4.4.4 - Object 308Fh: Encoder Input Resolution

Number of encoder pulses per motor revolution (rotating motor) or per motor pole pairs (linear motor) for a TTL type of encoder. Number of encoder signal periods per motor revolution (rotating motor) or per motor pole pairs (linear motor) for a SIN/COS encoder type.

Index	308Fh
Name	Encoder Input Resolution
Object Code	VAR
Data Type	Unsigned32
Object Class	all
Access	rw
PDO Mapping	No
Value Range	1F4h .. 0F4240h
Default Value	eeprom

3.4.4.5 - Object 3090h: Encoder Pulse Interpolation

Index	3090h
Name	Encoder Pulse Interpolation (for a SINCOS type of encoder)
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

Value	Description
0	No interpolation
1	SinCos interpolation (x1024)

3.4.4.6 - Object 3091h: Encoder Zero Mark Pitch

This parameter is equal to the number of encoder pulses between two successive Zero Mark signals divided by the Encoder Input Resolution value (see object 308Fh).
For a rotating motor, this parameter is generally equal to 1 because the encoder has got one Zero Mark signal per motor revolution. For a linear motor, this parameter is equal to the number of motor pole pairs between two successive Zero Mark signal.

Index	**3091h**
Name	Encoder Zero Mark Pitch (for motor feedback = encoder)
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

This parameter is used for the encoder counting protection when the encoder input is selected for the motor feedback.

Value	**Description**
0	No encoder counting protection
1 to 14	Encoder counting protection active: checks the number of encoder pulses between two successive Zero Mark signals. Used for rotating or linear motors.
15	Encoder counting protection active: controls the encoder position value for the Zero Mark signal activation. Used for applications with only one Zero Mark signal over the motor travel (linear motor)

3.4.4.7 - Object 3070h: Motor Feedback Selection

The motor feedback sensor is used to close the servo motor torque and speed control loops. The servo motor position loop can be closed by the motor feedback sensor or with the secondary sensor (see object 306Ah).

Index	**3070h**
Name	Motor Feedback Selection
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

Value	**Description**
0	Resolver input selection for the motor feedback
1	Encoder input selection for the motor feedback
2	SinCos tracks input selection for the motor feedback

3.4.4.8 - Object 3098h: Motor Feedback Programming

Index	3098h
Name	Motor Feedback Programming Procedure
Object Code	VAR
Data Type	Unsigned32
Object Class	all
Access	wo
PDO Mapping	No

The Motor Feedback Programming procedure modifies the encoder input selection and the motor feedback selection according to the parameters defined by objects 3070 and 3080.
In order to avoid running the motor feedback programming procedure by mistake, it is only executed when a specific signature is written to this object. The signature is 'fdbk'.

Signature	MSB			LSB
ASCII	k	b	d	f
hex	6Bh	62h	64h	66h

3.4.4.9 - Object 306Ah: Second Sensor Feedback

The selection of the second sensor feedback allows to close the position loop with the secondary sensor (encoder or resolver according to object 3070). Generally, this secondary sensor is on the load.

Index	306Ah
Name	Second sensor Feedback
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0

Bit Number	Description
0	Position feedback for position loop 0 => motor sensor feedback (resolver or encoder according to object 3070) 1 => secondary sensor feedback (encoder or resolver according to object 3070)

3.4.4.10 - Object 306Ch: Second Sensor Scaling factor

This parameter allows to increase the position resolution value on the second sensor.
-The “Revolution increments” value (object 608F sub-index 1) for an encoder secondary sensor feedback selection is calculated as described below : Revolution increments = number of secondary encoder pulses for one motor shaft revolution x 4 x second sensor scaling factor.
-The “Revolution increments” value (object 608F sub-index 1) for a resolver secondary sensor feedback selection is calculated as described below : Revolution increments = 65536 x number of secondary resolver shaft revolution for one motor shaft revolution x second sensor scaling factor.

Index	306Ch
Name	Second Sensor Scaling factor
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	1
Range	1 to 1024

Remark: for a SINCOS type of encoder, when encoder pulse interpolation is activated (object 3090), the scaling factor allows to adjust the interpolation factor (from 1 to 1024) for the second sensor position display.

3.5 - PROFILE POSITION MODE

In this mode, a trapezoidal trajectory generator gives the drive the possibility to execute a positioning with preset parameters as target position, profile speed and acceleration.

3.5.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
607A	VAR	Target Position	Integer32	rw
6080	VAR	Max Motor Speed	Unsigned16	rw
6081	VAR	Profile Velocity	Unsigned32	rw
6082	VAR	End Velocity	Unsigned32	rw
6083	VAR	Profile Acceleration	Unsigned32	rw
6084	VAR	Profile Deceleration	Unsigned32	rw
6086	VAR	Motion Profile Type	Integer16	rw
607D	ARRAY	Software Position Limit	Integer32	rw
6067	VAR	Position Window	Unsigned32	rw
307F	VAR	Position Modulo	Unsigned32	rw

3.5.2 - OBJECT DESCRIPTION

3.5.2.1 - Object 607Ah: Target Position

Target position is the final position where the motor will move to in profile position mode. The start position is the current position. The positioning begins with the rising edge of bit 4 of the control word (new set point). Bit 6 of control word indicates if the target position is absolute (=0) or relative (=1) movement.

Index	607Ah
Name	Target Position
Object Code	VAR
Data Type	Integer32
Object Class	pc pp eg
Access	rw
PDO Mapping	possible
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	(-2^{31}) to $(2^{31}-1)$ if Revolution increments (608F sub-index 1) is greather or equal to 10000h (-2^{15})xRevolution increments to $(2^{15}-1)$xRevolution increments if Revolution increments (608F sub-index 1) is lower than 10000h
Default Value	0

3.5.2.2 - Object 6080h: Max Motor Speed

The *max motor speed* defines the maximum speed the drive can reach. To avoid a saturation of the servo loop, the running speed must be less than *max. motor speed* (depends on the overshoot accepted for the servo loop response).

Index	6080h
Name	Max Motor Speed
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Unit	rpm
Value Range	100..25000 rpm
Default Value	eeprom

3.5.2.3 - Object 6081h: Profile Velocity

The *profile velocity* is the running velocity for a positioning. If the positioning is too short, the profile velocity may not be reached.

Index	**6081h**
Name	Profile Velocity
Object Code	VAR
Data Type	Unsigned32
Object Class	pp eg
Access	rw
PDO Mapping	Possible
Unit	inc/s
Value Range	-
Default Value	1000h

3.5.2.4 - Object 6082h: End Velocity

The *End velocity* is the final velocity value when the target position is reached. When the motor must stop at the target position, *End velocity=0.*

Index	**6082h**
Name	End Velocity
Object Code	VAR
Data Type	Unsigned32
Object Class	pp
Access	rw
PDO Mapping	Possible
Unit	inc/s
Value Range	-
Default Value	0

3.5.2.5 - Object 6083h: Profile Acceleration

Index	**6083h**
Name	Profile Acceleration
Object Code	VAR
Data Type	Unsigned32
Object Class	pp eg
Access	rw
PDO Mapping	No
Unit	inc/s^2
Value Range	-
Default Value	10000h

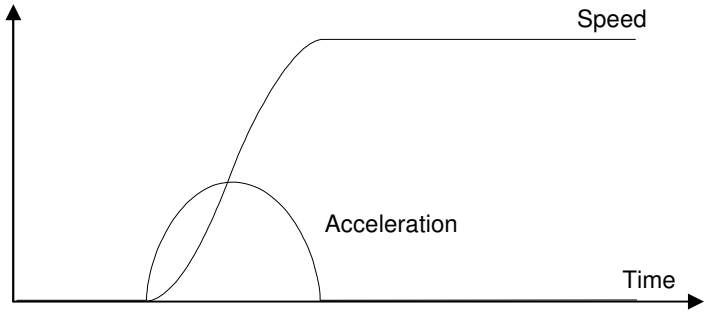
3.5.2.6 - Object 6084h : Profile Deceleration

Index	**6084h**
Name	Profile Deceleration
Object Code	VAR
Data Type	Unsigned32
Object Class	pp eg
Access	rw
PDO Mapping	No
Unit	inc/s^2
Value Range	-
Default Value	10000h

3.5.2.7 - Object 6086h: Motion Profile Type

Index	6086h
Name	Motion Profile Type
Object Code	VAR
Data Type	Integer16
Object Class	pp eg
Access	rw
PDO Mapping	No
Value Range	0 -> Trapezoidal profile -1 -> S-Curve
Default Value	0

The S-curve is defined by a polynomial. The acceleration profile is therefore parabolic.



3.5.2.8 - Object 607Dh: Software Position Limit

The software position limits are only active for a linear axis (see object 3360).
The operation mode of the software position limits is selected by object 3361.

Index	607Dh
Name	Software Position Limit
Object Code	ARRAY
Number of Elements	2

Value Description

Sub Index	1
Description	Min Position Limit
Data Type	Integer32
Object Class	all
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	$(-2^{31})..(2^{31}-1)$
Default Value	-2^{31}

Sub Index	2
Description	Max Position Limit
Data Type	Integer32
Object Class	all
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	$(-2^{31})..(2^{31}-1)$
Default Value	$2^{31}-1$

3.5.2.9 - Object 6067h: Position Window

The "Position Window" defines a symmetrical range of accepted positions relatively to the target position. If the current position of the motor is within the position window, this target position is considered as reached (bit 10 of status word - Target Reached - is set). If the position window value is 0, the position window control is not active.

Index	**6067h**
Name	Position Window
Object Code	VAR
Data Type	Unsigned32
Object Class	pp
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Default Value	0

3.5.2.10 - Object 307Fh: Position Modulo

When the position counter reaches the "Position Modulo" value, it is reset at 0. The motor runs only in the positive direction for an absolute displacement within the "Position Modulo" value (for example from position 90 to position 10 with a position modulo = 100 : the displacement of the motor is 20 in the forward direction). If the value of this object is set at 0, the modulo function is not activated.

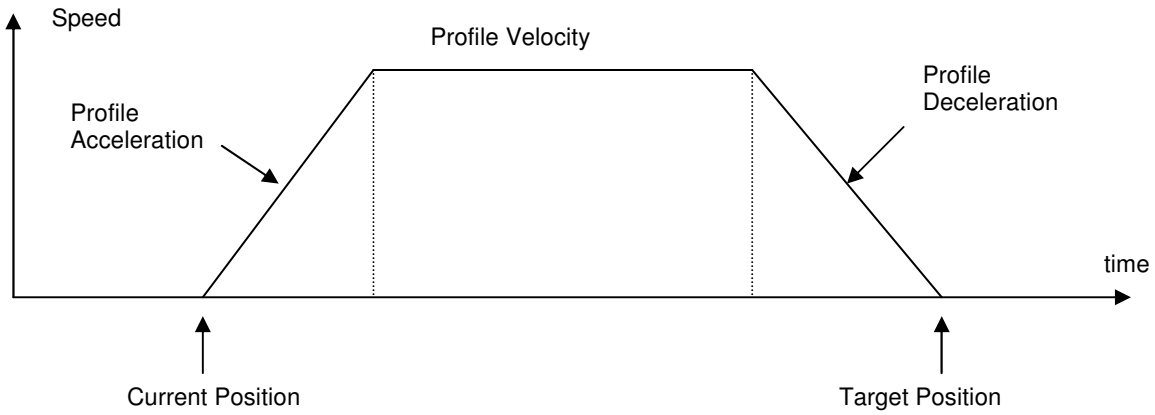
Index	**307Fh**
Name	Position Modulo
Object Code	VAR
Data Type	Unsigned32
Object Class	pp
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Default Value	0

3.5.3 - FUNCTIONAL DESCRIPTION

In profile position mode, these bits in the control word are relative to the control of the trajectory :

Bit Number	**Profile Position Mode**
4	new set point
5	change set immediately
6	0: absolute 1: relative

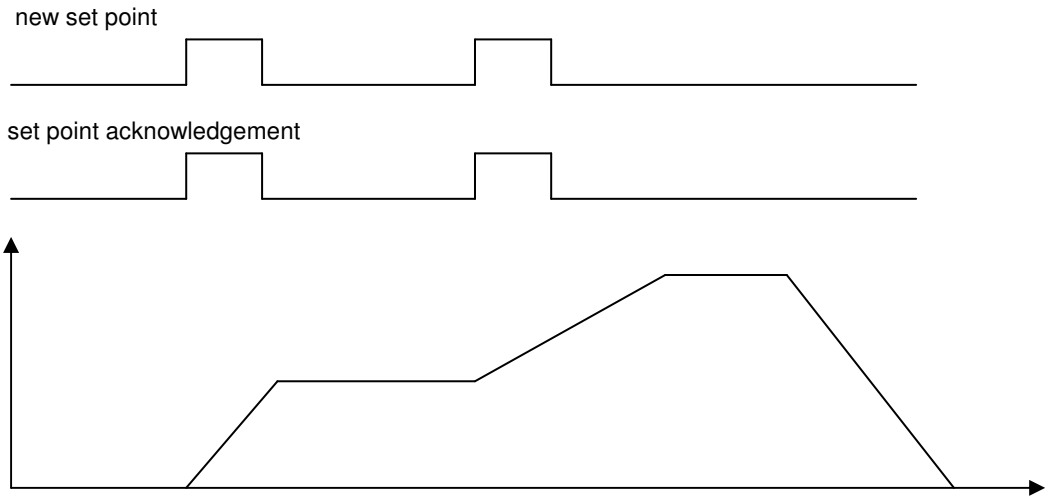
The movement will be triggered by a rising edge of bit 4 (new_set_point) of the control word. The acknowledgement of the new set point is confirmed by bit 12 (setpoint acknowledgement) of the status word. The target position will be taken as relative to the current position if bit 6 of control word = 1.



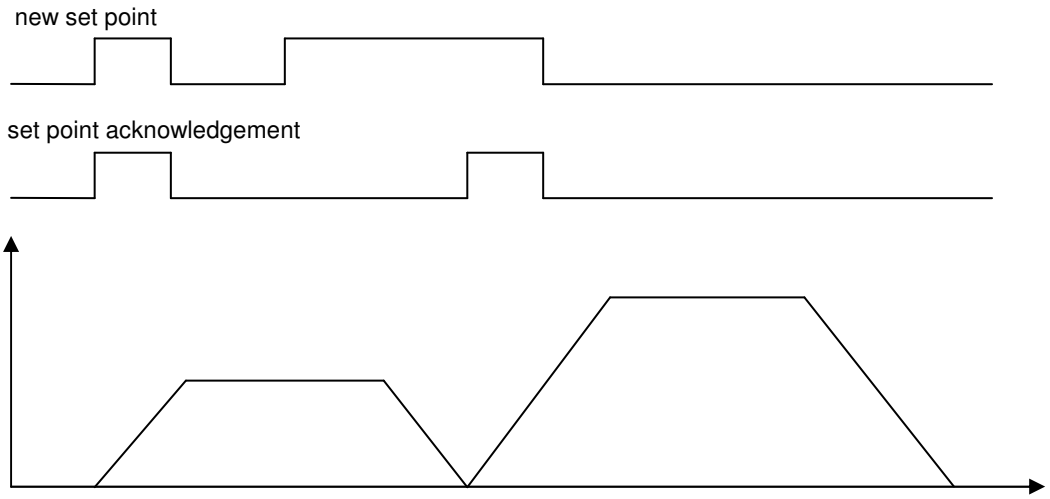
The speed profile is trapezoidal (motion profile type = 0) or S-curve (motion profile type = -1).

Change setpoint immediately

Bit change_set_immediately = 1 :



Bit change_set_immediately = 0 :



3.6 - HOMING MODE

3.6.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
607C	VAR	Home Offset	Integer32	rw
6098	VAR	Homing Method	Integer8	rw
6099	ARRAY	Homing Speeds	Unsigned32	rw
609A	VAR	Homing Acceleration	Unsigned32	rw

Manufacturer Specific Objects:

Index	Object	Name	Type	Attr.
309B	VAR	Zero Shift	Unsigned16	rw
309C	VAR	Homing Current Limit	Unsigned16	rw

3.6.2 - OBJECT DESCRIPTION

3.6.2.1 - Object 607Ch: Homing Offset

The *Home Offset* defines the position feedback value when the motor reaches the homing position.

Index	607Ch
Name	Home Offset
Object Code	VAR
Data Type	Integer32
Object Class	hm
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	$(-2^{31})..(2^{31}-1)$
Default Value	0

3.6.2.2 - Object 6098h: Homing Method

The *Homing Method* defines various ways of the drive to search the homing position.

Index	6098h
Name	Homing Method
Object Code	VAR
Data Type	Integer8
Object Class	hm
Access	rw
PDO Mapping	No
Default Value	23h

Value Description
Method supported: 1..14, 17..30, 33..35.
Methods specific: -1, -2, -3, -4.

Method	Search for Switch	Search for Index Pulse	Remarks
1	Negative Limit Switch	Exterior	
2	Positive Limit Switch	Exterior	
3	Positive Home Switch	Exterior	
4	Positive Home Switch	Interior	
5	Negative Home Switch	Exterior	
6	Negative Home Switch	Interior	
7	Home Switch, Negative Side	Exterior	Positive Initial Move. Reverse direction on Positive Limit Switch.
8	Home Switch, Negative Side	Interior	Positive Initial Move. Reverse direction on Positive Limit Switch.

9	Home Switch, Positive Side	Interior	Positive Initial Move. Reverse direction on Positive Limit Switch.
10	Home Switch, Positive Side	Exterior	Positive Initial Move. Reverse direction on Positive Limit Switch.
11	Home Switch, Positive Side	Exterior	Negative Initial Move. Reverse direction on Negative Limit Switch.
12	Home Switch, Positive Side	Interior	Negative Initial Move. Reverse direction on Negative Limit Switch.
13	Home Switch, Negative Side	Interior	Negative Initial Move. Reverse direction on Negative Limit Switch.
14	Home Switch, Negative Side	Exterior	Negative Initial Move. Reverse direction on Negative Limit Switch.
17	Negative Limit Switch	-	Methods 17 to 30 are similar to methods 1 to 14 without index pulse
18	Positive Limit Switch	-	
19	Positive Home Switch	-	
20	Positive Home Switch	-	
21	Negative Home Switch	-	
22	Negative Home Switch	-	
23	Home Switch, Negative Side	-	
24	Home Switch, Negative Side	-	
25	Home Switch, Positive Side	-	
26	Home Switch, Positive Side	-	
27	Home Switch, Positive Side	-	
28	Home Switch, Positive Side	-	
29	Home Switch, Negative Side	-	
30	Home Switch, Negative Side	-	
33		First Index Pulse	Negative Initial Move.
34		First Index Pulse	Positive Initial Move.
35		-	Homing On Current Position
-1	Mechanical Limit, Negative Move	First Index Pulse	
-2	Mechanical Limit, Positive Move	First Index Pulse	
-3	Mechanical Limit, Negative Move	-	
-4	Mechanical Limit, Positive Move	-	

Remark: When "motor position polarity" is reversed (see object 607E), the described methods are using the opposite limit switch,

3.6.2.3 - Object 6099h: Homing Speeds

Homing Speeds defines the motor speed when searching the homing position.

Index	**6099h**
Name	Homing Speeds
Object Code	ARRAY
Number of Elements	2
Data Type	Unsigned32

Value Description

Sub Index	1
Description	Speed during search of switch
Object Class	hm
Access	rw
PDO Mapping	No
Unit	inc/s
Default Value	00000019h

Sub Index	2
Description	Speed during search of zero
Object Class	hm
Access	rw
PDO Mapping	No
Unit	inc/s
Default Value	0000000Ah

3.6.2.4 - Object 609Ah: Homing Acceleration

Index	**609Ah**
Name	Homing Acceleration
Object Code	VAR
Data Type	Unsigned32
Object Class	hm
Access	rw
PDO Mapping	No
Unit	inc/s^2
Default Value	00010000h

3.6.2.5 - Object 309Bh: Zero Shift

The "Zero Shift" parameter defines the shift of the zero pulse for the homing search.

Index	**309Bh**
Name	Zero Shift
Object Code	VAR
Data Type	Unsigned16
Object Class	hm
Access	rw
PDO Mapping	No
Conversion	8000h -> 180 degree
Value Range	0..FFFFh
Default Value	eeprom

3.6.2.6 - Object 309Ch: Homing Current Limit

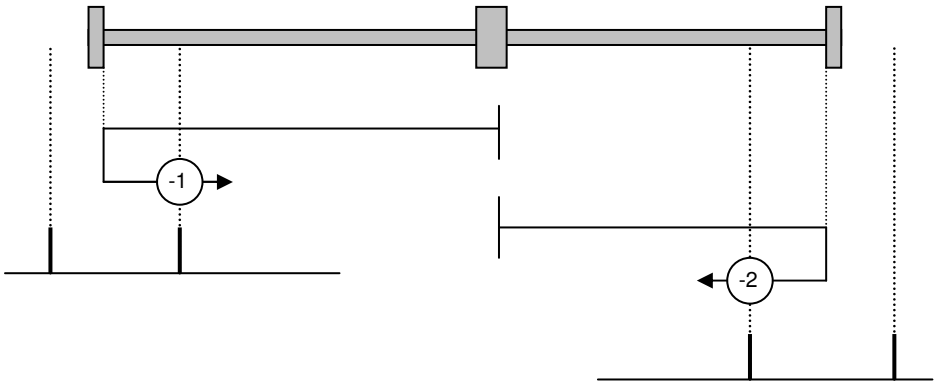
The "Homing current limit" defines the limit of current during homing on the mechanical limit. The value is defined as a percentage of the drive maximum current (defined by object 6510h sub-index 5).

Index	**309Ch**
Name	Homing Current Limit
Object Code	VAR
Data Type	Unsigned16
Object Class	hm
Access	rw
PDO Mapping	No
Unit	%
Conversion	0 to 3FFFh -> 0% to 100 %
Default Value	400h

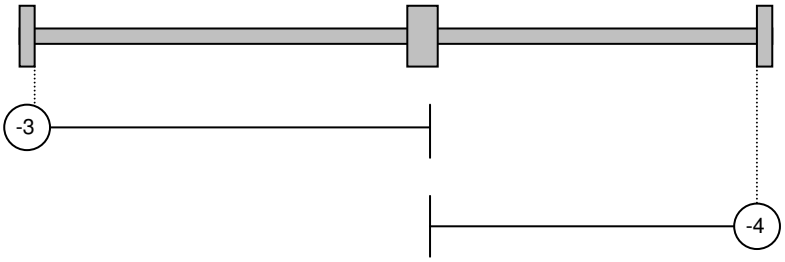
3.6.3 - FUNCTIONAL DESCRIPTION

The "Homing Current Limit" parameter defines the limit of current in the motor during the homing procedure. When the mechanical limit is reached, the current in the motor increases up to this limit and the motor speed is 0. This position will be taken as the homing position. An offset value (object 607Ch) can be used to preset the homing position value).

Method -1 and -2 define homing on mechanical limit with index pulse.



Method -3 and -4 define homing on mechanical limit.



3.7 - POSITION CONTROL FUNCTION

3.7.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
6063	VAR	Position Actual Value	Integer32	ro
6064	VAR	Position Actual Value	Integer32	ro
6065	VAR	Following Error Window	Unsigned 32	rw
3067	VAR	Position loop deadband	Unsigned32	rw
60FB	RECORD	Position Control Parameter Set		rw

3.7.2 - OBJECT DESCRIPTION

3.7.2.1 - Object 6063h: Position Actual Value*

Index	6063h
Name	Position Actual Value*
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	$(-2^{31})..(2^{31}-1)$

3.7.2.2 - Object 6064h: Position Actual Value

Index	6064h
Name	Position Actual Value
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	Possible
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	$(-2^{31})..(2^{31}-1)$

The **TTA-CAN** does not support different types of position unit, but only increments. Object 6064h *Position Actual Value* has thus the same unit as 6063h.

3.7.2.3 - Object 3067h: Position Loop Deadband

Index	3067h
Name	Position Loop Deadband
Object Code	VAR
Data Type	Unsigned32
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	0..7FFFh
Default Value	0

The "Position loop deadband" parameter introduces a deadband at standstill around the position loop setpoint. When the position error is lower than this parameter value, the position loop proportional gain is set at 0. This parameter is reserved for specific applications with load backlashes and a high level of dry frictions.

3.7.2.4 - Object 6065h: Following Error Window

Index	6065h
Name	Following Error Window
Object Code	VAR
Data Type	Unsigned 32
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	0..7FFF FFFFh (For the value 0 the following error protection is desactivated)
Default Value	eeprom

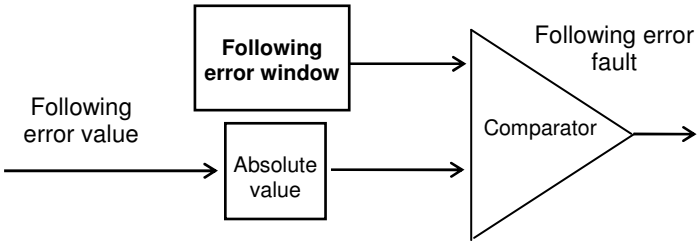
3.7.2.5 - Object 3031h: Position Following Error Dynamic Gain

Index	3031h
Name	Position Following Error Dynamic Gain
Object Code	VAR
Data Type	Unsigned16
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Value Range	0..7FFFh
Default Value	eeprom

3.7.2.6 - Object 3033h: Position Following Error Detection Mode

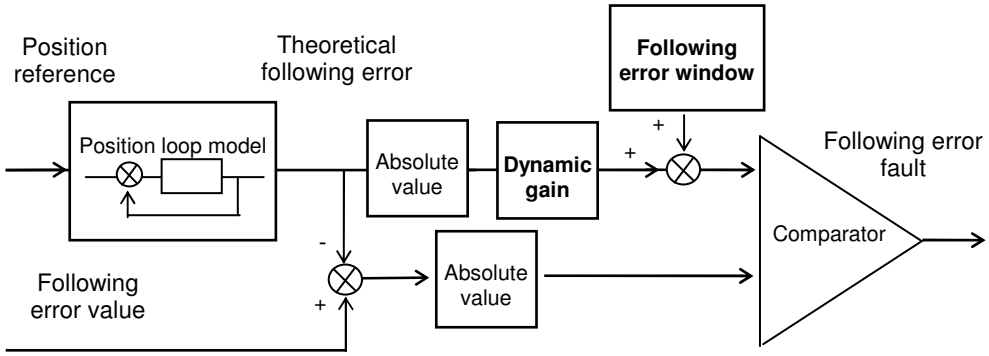
Index	3033h
Name	Position Following Error Detection Mode 1 --> Relative to the dynamic model 0 --> Absolute
Object Code	VAR
Data Type	Unsigned16
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Value Range	0..1
Default Value	eeprom

If the following error detection mode is set at 0, the position following error fault is generated as described below:



The measured position following error value is continuously compared with the **following error window** parameter value. When the measured position error is exceeding the **following error window**, the position following error fault is released.

If the following error detection mode is set at 1, the position following error fault is generated as described below:

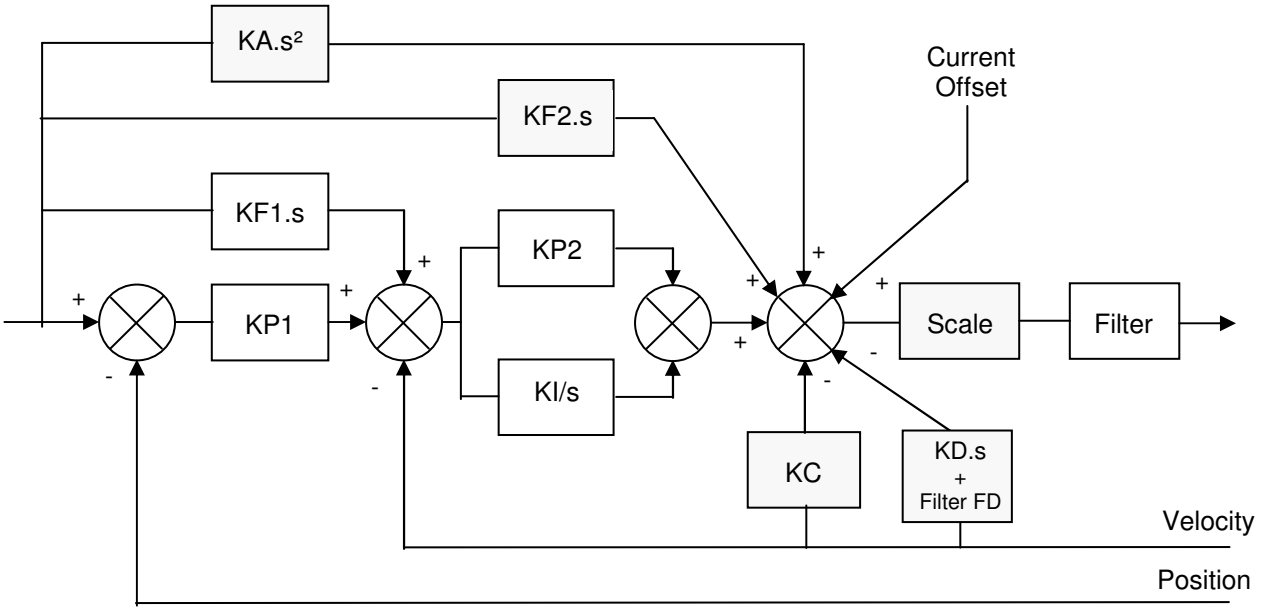


The measured position following error value is continuously compared with the theoretical position following error given by the position loop model. When the difference is exceeding the **following error window**, the position following error fault is released. In this configuration, when the position servo loop is adjusted to get the motor position continuously lagging the reference position (positioning applications without overshoot with a large position following error value), any small anomaly in the servo drive behaviour can be detected.

3.7.2.7 - Object 3214h: Position Error

Index	3214h
Name	Position Error Position Error is the difference between position reference and position feedback.
Object Code	VAR
Data Type	Integer32
Object Class	pp ip hm eg
Access	ro
PDO Mapping	possible

3.7.2.8 - Object 60FBh: Position Control Parameter Set



KA Feedforward Acceleration
KF1 Feedforward Speed 1
KF2 Feedforward Speed 2
KP1 Proportional Position Gain
KP2 Proportional Speed Gain
KI Integral Speed Gain
KC Damping Gain
KD Derivative Gain
FD Derivative Gain filter
Scale Regulator Gains Scaling

Index	**60FBh**
Name	Position Control Parameter Set
Object Code	RECORD
Number of Elements	4

Value Description

Sub Index	1
Description	Proportional Position Gain Defines the proportional gain that acts upon the position error (KP1).
Data Type	Unsigned16
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	2
Description	Feedforward Speed 1 Gain Defines the feedforward term amplitude (KF1) corresponding to the speed input command (derivation of the position input command). This feedforward term allows to reduce the following error during the motor acceleration and deceleration phases.
Data Type	Unsigned16
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	3
Description	Feedforward Acceleration Gain Defines the feedforward acceleration corresponding to the acceleration input command (second derivation of the position input command).This feedforward term allows to reduce the following error during the motor acceleration and deceleration phases.
Data Type	Unsigned16
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	4
Description	Feedforward Speed 2 Gain This gain value is equal to the damping speed gain value +Feedforward friction gain value. The feedforward friction gain allows to cancel the load viscous friction effect (load viscous friction torque is proportional to axis speed).This feedforward term allows to reduce the following error during the motor acceleration and deceleration phases.
Data Type	Unsigned16
Object Class	pp ip hm eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

3.8 - INTERPOLATED POSITION MODE

The interpolated position mode is used to control several axes in coordination. The trajectory must be generated by the host controller and the elementary set point is sent at a fixed cycle time (same as communication cycle time) to all axes.
The cycle time synchronisation of all axes is assumed by SYNC message. The flow of set point data must be sent in real-time.
The elementary set point could be only position if linear interpolation is chosen. The PV cubic interpolation mode requires position and velocity for each set point. The P3 cubic interpolation mode requires only position set point because the interpolator is using the 3 last position set points. However, the interpolation error is inherent when the acceleration is changing with the P3 cubic interpolation mode.
Both cubic interpolation modes require high position resolution when operating at low speed values. At very low speed, the linear interpolation mode is giving best results.

3.8.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
60C0	VAR	Interpolation Submode Select	Integer16	rw
60C1	RECORD	Interpolation Data Record		rw
60C4	RECORD	Interpolation Data Configuration		rw

3.8.2 - OBJECT DESCRIPTION

3.8.2.1 - Object 60C0h: Interpolation Submode Select

Index	**60C0h**
Name	Interpolation Submode Select
Object Code	VAR
Data Type	Integer16
Object Class	ip
Access	rw
PDO Mapping	No
Default Value	0

Interpolation Submode Select	**Description**
0	Linear interpolation
-1	PV cubic interpolation
-2	P3 cubic interpolation

3.8.2.2 - Object 60C1h: Interpolation Data Record

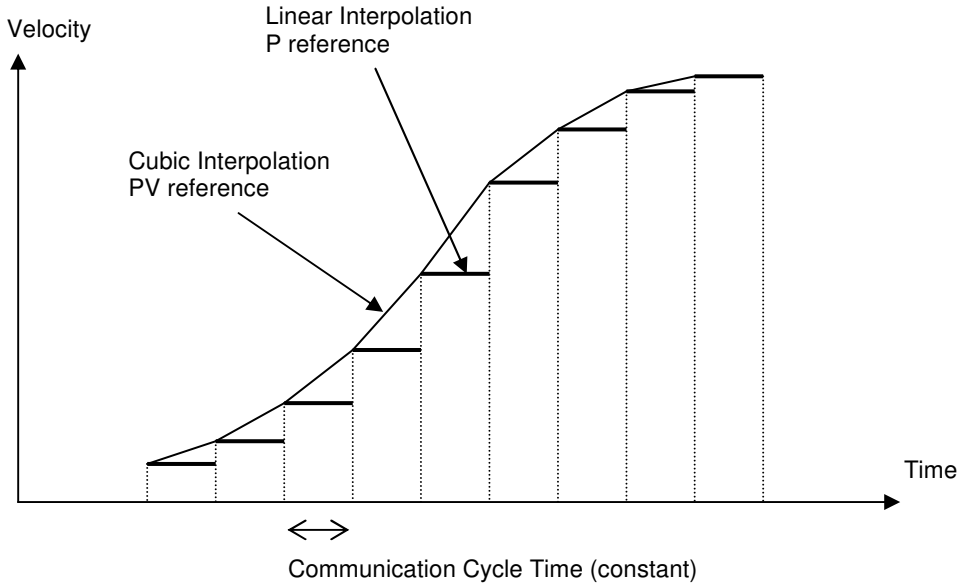
Index	**60C1h**
Name	Interpolation data record
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	First parameter of ip function
Data Type	Integer32
Object Class	ip
Access	rw
PDO Mapping	Possible

Sub Index	2
Description	Second parameter of ip function
Data Type	Integer32
Object Class	ip
Access	rw
PDO Mapping	Possible

When in linear interpolation mode, only the first parameter of interpolation data record is used. The data must be the position reference.
When in PV cubic interpolation mode, the first parameter of interpolation data record must contain position reference and the second parameter of interpolation data record contains velocity reference.



Note: The velocity reference for each set-point must be the instantaneous velocity at this point (not the average velocity).

3.8.2.3 - Object 60C4h: Interpolation Data Configuration

Index	**60C4h**
Name	Interpolation Data Configuration
Object Code	RECORD
Number of Elements	6

Value Description

Sub Index	1
Description	Max Buffer Size
Data Type	Unsigned32
Object Class	ip
Access	ro
PDO Mapping	No
Default Value	2

Sub Index	2
Description	Actual Size
Data Type	Unsigned32
Object Class	ip
Access	ro
PDO Mapping	No
Default Value	2

Sub Index	3
Description	Buffer Organisation
Data Type	Unsigned8
Object Class	ip
Access	ro
PDO Mapping	No
Default Value	0

Sub Index	4
Description	Buffer Position
Data Type	Unsigned16
Object Class	ip
Access	rw
PDO Mapping	No
Default Value	0

Sub Index	5
Description	Size of Data record
Data Type	Unsigned8
Object Class	ip
Access	wo
PDO Mapping	No

Sub Index	6
Description	Buffer Clear
Data Type	Unsigned8
Object Class	ip
Access	wo
PDO Mapping	No

3.9 - PROFILE VELOCITY MODE

The profile velocity mode authorizes the drive to operate with a velocity reference. Only speed loop and current loop are closed in this mode.

3.9.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
6069	VAR	Velocity Sensor Actual Value	Integer32	ro
606A	VAR	Sensor Selection Code	Integer16	ro
606B	VAR	Velocity Demand Value	Integer32	ro
606C	VAR	Velocity Actual Value	Integer32	ro
606D	VAR	Velocity Window	Unsigned 16	rw
606E	VAR	Velocity Window Time	Unsigned 16	rw
606F	VAR	Velocity Threshold	Unsigned 16	rw
6070	VAR	Velocity Threshold Time	Unsigned 16	rw
60FF	VAR	Target Velocity	Integer32	rw
604F	VAR	Ramp Function	Unsigned32	rw
60F9	RECORD	Velocity Control Parameter Set		rw
30F9	VAR	Integrator low frequency limit	Unsigned16	rw
30FB	VAR	Regulator Gain Scaling	Unsigned16	rw
3422	VAR	Time interval for speed measurement	Unsigned16	rw

3.9.2 - OBJECT DESCRIPTION

3.9.2.1 - Object 6069h: Velocity Sensor Actual Value

Index	6069h
Name	Velocity Sensor Actual Value
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	No
Unit	inc/s

3.9.2.2 - Object 606Ah: Sensor Selection Code

Index	606Ah
Name	Sensor Selection Code
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	ro
PDO Mapping	No
Default Value	0

Sensor Selection Code	Description
0	Velocity actual value from position sensor

3.9.2.3 - Object 606Bh: Velocity Demand Value

Index	606Bh
Name	Velocity Demand Value
Object Code	VAR
Data Type	Integer32
Object Class	pv
Access	ro
PDO Mapping	No
Unit	inc/s

3.9.2.4 - Object 606Ch: Velocity Actual Value

Index	606Ch
Name	Velocity Actual Value
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	Possible
Unit	inc/s

3.9.2.5 – Object 606Dh: Velocity window

Index	606Dh
Name	Velocity window: When the difference between the Target and the Actual velocity value is within the Velocity window longer than the Velocity window time, the target velocity is considered as reached (bit 10 of status word – Target Reached – is set). If the Velocity window value is 0, the velocity window control is not active.
Object Code	VAR
Data Type	Unsigned 16
Object Class	pv
Access	rw
PDO Mapping	No
Unit	inc/s
Default Value	0

3.9.2.6 – Object 606Eh: Velocity window time

Index	606Eh
Name	Velocity window time:
Object Code	VAR
Data Type	Unsigned 16
Object Class	pv
Access	rw
PDO Mapping	No
Unit	ms
Default Value	0

3.9.2.7 – Object 606Fh: Velocity threshold

Index	606Fh
Name	Velocity threshold: When the Actual velocity value is below the Velocity threshold longer than the Velocity threshold time, the motor is considered at standstill (bit 12 of status word – Velocity = 0 – is set). If the Velocity threshold value is 0, the velocity threshold control is not active.
Object Code	VAR
Data Type	Unsigned 16
Object Class	pv
Access	rw
PDO Mapping	No
Unit	inc/s
Default Value	0

3.9.2.8 – Object 6070h: Velocity threshold time

Index	6070h
Name	Velocity threshold time:
Object Code	VAR
Data Type	Unsigned 16
Object Class	pv
Access	rw
PDO Mapping	No
Unit	ms
Default Value	0

3.9.2.9 - Object 60FFh: Target Velocity

Index	60FFh
Name	Target Velocity
Object Code	VAR
Data Type	Integer32
Object Class	pv
Access	rw
PDO Mapping	Possible
Unit	inc/s
Default Value	0

3.9.2.10 - Object 604Fh: Ramp Function

Index	604Fh
Name	Ramp Function When different from 0, the ramp function is effective and this parameter defines the time for a speed variation between 0 and max. motor speed (6080h).
Object Code	VAR
Data Type	Unsigned32
Object Class	pv
Access	rw
PDO Mapping	No
Unit	ms
Value Range	0..16383
Default Value	eeprom

3.9.2.11 - Object 60F9h: Velocity Control Parameter Set

Index	60F9h
Name	Velocity Control Parameter Set
Object Code	RECORD
Number of Elements	8

See 3.7.2.7 for the controller diagram.

Value Description

Sub Index	1
Description	Proportional Speed Gain Defines the proportional regulator gain (KP2) that acts upon the speed error.
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	2
Description	Integral Speed Gain Defines the integral regulator gain (KI) that acts upon the speed error.
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	3
Description	Integral 2 Speed Gain This gain is used only in PI² speed regulator. It is the same one as the proportional position gain.
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	4
Description	Damping Gain This gain is used for getting the maximum servo loop stiffness.
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	5
Description	Speed Loop Low-pass filter Defines the cut-off frequency at -3 dB (Fev) of the first order filter that acts upon the current control. The value of this parameter is depending on the selected bandwidth.
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Conversion	Cut-off frequency (Hz) = 1000/pi*Ln(65536/[60F9-5])
Value Range	0B10h (1000 Hz) et F069h (20 Hz)
Default Value	eeprom

Sub Index	6
Description	Anti-resonance Filter Selection Enables or inhibits the anti-resonance filter.
Data Type	Unsigned8
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..1
Default Value	eeprom

Anti-resonance filter Selection	**Function**
0	standard filter
1	anti-resonance filter

Sub Index	7
Description	Derivative Gain
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	00..FFFFh
Default Value	eeprom

Sub Index	8
Description	Derivative Gain Filter Defines the time constant of the first order filter that acts upon the derivative gain.
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Conversion	1/1000 for getting the time constant in ms
Value Range	0..2710h
Default Value	eeprom

3.9.2.12 – Object 30F9h: Integrator low frequency limit

Index	30F9h
Description	Integrator low frequency limit Defines the low frequency value from which the controller integrator term is saturated. This parameter is used for reducing the motor heating in applications with large dry frictions due to the mechanical load.
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Conversion	Low frequency limit (Hz) = 1000/pi*Ln(65536/[30F9-0])
Value Range	FFFFh (0 Hz) to 8000h (220 Hz) 0 value = Integrator low frequency limit desactivated
Default Value	0

3.9.2.13 - Object 30FBh: Regulator Gains Scaling

Index	30FBh
Name	This term is acting like a multiplying factor for the regulator gains (KP2, KC, KI, KD, KA, KB) in order to avoid the gain values saturation when the load/motor inertia ratio is very high.
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Value Range	1..1000
Defaut Value	eeprom

3.9.2.14 - Object 3422h: Time Interval for speed measurement

Index	3422h
Name	Time Interval for Speed Measurement 0 -> Time interval for speed measurement = 0.5 ms 1 -> Time interval for speed measurement = 1 ms 2 -> Time interval for speed measurement = 2 ms The higher the time interval value, the better the speed measurement resolution but the speed loop gain values must be reduced to maintain the servo loop stability because of the larger speed measurement delay. For a TTL encoder feedback : The speed measurement resolution value can be calculated according to the following formula : Speed resolution (rpm) = 15000 / Encoder feedback resolution (object 308Fh) / Time interval (ms)
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Range	0 to 2
Defaut Value	eeprom

3.9.2.15 - Object 30F5h: Speed Following Error Limit

Index	30F5h
Name	Speed Following Error Limit
Object Code	VAR
Data Type	Integer16
Object Class	pv
Access	rw
PDO Mapping	No
Unit	rpm
Range	0 to 25000
Default Value	eeprom

3.10 - PROFILE TORQUE MODE

In this mode, the drive operates only with current loops and there is no speed or position control.

3.10.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
6071	VAR	Target Torque	Integer16	rw
6087	VAR	Torque Slope	Unsigned32	rw
6088	VAR	Torque Profile Type	Integer16	ro
60F6	RECORD	Torque Control Parameter Set		rw
6078	VAR	Filtered current monitor	Integer16	ro
3078	VAR	Current monitor filter	Unsigned16	rw

3.10.2 - OBJECT DESCRIPTION

3.10.2.1 - Object 6071h: Target Torque

Index	6071h
Name	Target Torque
Object Code	VAR
Data Type	Integer16
Object Class	tq
Access	rw
PDO Mapping	Possible
Unit	per thousand of the drive max. current

3.10.2.2. Object 6087h: Torque Slope

Index	6087h
Name	Torque Slope
Object Code	VAR
Data Type	Unsigned32
Object Class	tq
Access	rw
PDO Mapping	No
Default Value	1000h

3.10.2.3 - Object 6088h: Torque profile type

This parameter is not used by TRANSTECHNIK Servo Drives, but only for compliance.

Index	6088h
Name	Torque profile type
Object Code	VAR
Data Type	Integer16
Object Class	tq
Access	ro
PDO Mapping	No
Default Value	0

3.10.2.4 - Object 60F6h: Torque Control Parameter Set

This object defines the parameters of the current loops.

Index	60F6h
Name	Torque Control Parameter Set
Object Code	RECORD
Number of Elements	4

Value Description

Sub Index	1
Description	q-Loop Proportional Gain
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	2
Description	q-Loop Integral Gain
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	3
Description	d-Loop Proportional Gain
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

Sub Index	4
Description	d-Loop Integral Gain
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Value Range	0..FFFFh
Default Value	eeprom

3.10.2.5 - Object 6078h: Filtered current monitor

This object is the filtered motor current monitor. This monitor is calculated with vector transformation. This means that the current monitor is correct only if the motor position is correct (no motor sensor error...).

Index	**6078h**
Name	Filtered Current Monitor
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	ro
PDO Mapping	Possible
Unit	per thousand of amplifier Rating

3.10.2.6 - Object 3078h: Current monitor filter

This object defines the time constant value for the first order filter acting on the motor current monitor (see object 6078h) .

Index	3078h
Name	Current Monitor filter
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	no
Unit	micro-seconds
Range	500 to 10 000
Default value	4000

3.11 – STEPPER EMULATION MODE

The stepper motor emulation application is only possible for motors equipped with a resolver such as a position feedback sensor (see object 3070: motor feedback selection). The Pulse and Direction input configuration must be selected in the Encoder input selection (see object 3080). The Pulse train and the Direction signal are both generated by the host controller via its stepper motor control interface.
The Stepper emulation mode of operation is selected by setting the value –1 in the object 6060. When Pulse following control is enabled in the control word, the servo motor position setpoint is recieived via the Pulse and Direction input pins of the TTA-CAN amplifier.

3.11.1 - FUNCTIONAL DESCRIPTION

The specific bits of the control word (object 6040) used in stepper emulation mode are described below:

Bit	Function
4	Enable pulse following
5	reserved
6	reserved

The specific bits of the status word (object 6041) used in stepper emulation mode are described below:

Bit	Function
12	Pulse following OK
13	reserved

When the amplifier is switched on with the stepper emulation mode selected, Pulse following control is disabled. In this case, the input pulses are not counted and the motor is enabled at standstill.

When bit 4 of the control word (Enable pulse following) is set at 1, the motor starts following the input pulses.
If a pulse train is received by the amplifier when bit 4 of the control word (Enable pulse following) is set at 1, the motor speed is ramping according to the ramp function (object 604F). When the motor speed value corresponds to the pulse train frequency, bit 12 of the status word (Pulse following OK) is set at 1. At this moment, the motor position is locked in phase and frequency with the pulse train. The polarity of the motor speed is given by the logic state of the Direction input.
When bit 4 of the control word (Enable pulse following) is set at 0 while the motor is running, the motor speed is ramping to 0 according to the ramp function (object 604F). Bit 12 of the status word (Pulse following OK) is set at 0 as soon has the motor starts braking.

3.11.2 – PARAMETER SETTING

The motor displacement direction with regard to the Direction input logic state can be configured by using bit 5 of the Polarity parameter (object 607E).

The motor Maximum speed value is calculated according to the host controller pulse frequency limit as follow :
Maximum speed (rpm) = 120 x pulse frequency limit (Hz) / Revolution increments (object 608F sub-index 1)
The Max Motor Speed parameter (object 6080) is set to the previously calculated maximum speed value + 10% to avoid amplifier speed saturation.

The Revolution increments value (object 608F sub-index 1) for a stepper emulation application is calculated as follow : Revolution increments = 2 x number of controller pulses for one motor shaft revolution x second sensor scaling factor (object 306C). The second sensor scaling factor is set at 1 for stepper emulation application.

3.12 – ELECTRONIC GEARING MODE

Electronic gearing allows to synchronize the position of the servo motor with the position of a master axis according to a given gearing ratio value. The master axis is equipped with a position sensor connected to the second sensor input of the TTA-CAN amplifier.
The Electronic gearing mode of operation is selected by setting the value –2 in the object 6060. When Motor Gearing control is enabled in the control word, the servo motor position setpoint is received via the second sensor input (encoder or resolver according to the object 3070).

3.12.1 - FUNCTIONAL DESCRIPTION

The specific bits of the control word (object 6040) used in electronic gearing mode are described below:

Bit	Function
4	Enable motor gearing
5	Start offset profile
6	reserved

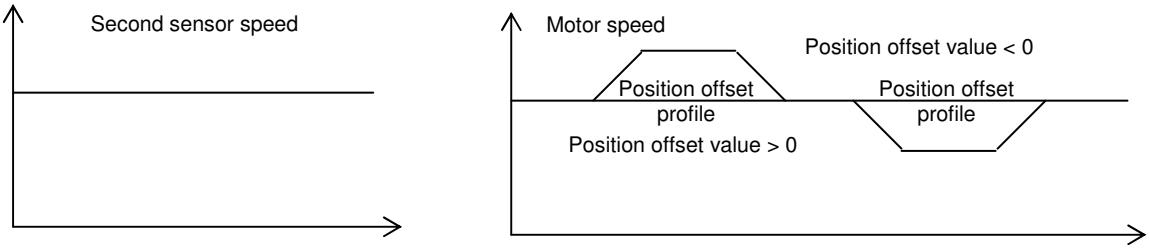
The specific bits of the status word (object 6041) used in electronic gearing mode are described below:

Bit	Function
12	Motor gearing OK
13	reserved

When the amplifier is switched on with the electronic gearing mode selected, Motor gearing control is disabled. In this case, the second sensor displacement is not measured and the motor is enabled at standstill.

When bit 4 of the control word (Enable motor gearing) is set at 1, the motor starts following the second sensor displacement.
If the second sensor is moving when bit 4 of the control word (Enable pulse following) is set at 1, the motor speed is ramping according to the ramp function (object 604F). When the motor speed value corresponds to the second sensor velocity, bit 12 of the status word (Motor gearing OK) is set at 1. At this moment, the motor position is locked in phase and frequency with the second sensor position.
When the bit 4 of the control word (Enable motor gearing) is set at 0 while the motor is running, the motor speed is ramping to 0 according to the ramp function (object 604F). Bit 12 of the status word (Motor gearing OK) is set at 0 as soon has the motor starts braking.

When the motor gearing is enabled, a position offset value can be added to the second sensor position value. In this case, the motor displacement of the position offset value will be triggered by a rising edge of bit 5 of the control word (Start offset profile). The total motor displacement during the position offset profile execution is equal to the second sensor displacement plus the position offset value as shown below.



3.12.2 – PARAMETER SETTING

The motor displacement direction with regard to the second sensor direction can be configured by using bit 5 of the Polarity parameter (object 607E).

For an electronic gearing application with an encoder type of sensor on the master axis, the Revolution increments value (object 608F sub-index 1) is calculated as follows:
Revolution increments = 4 x number of encoder pulses for one master encoder shaft revolution x second sensor scaling factor (object 306C) / Gearing ratio
Gearing ratio = Motor shaft speed / Master encoder shaft speed

For an electronic gearing application with a resolver sensor type on the master axis, the Position resolution parameter value is calculated as follows :
Position resolution = 65536 x second sensor scaling factor (object 306C) / Gearing ratio
Gearing ratio = Motor shaft speed / Master resolver shaft speed

The position offset value is entered in the object 607A. The profile velocity is defined by the object 6081, the profile acceleration is defined by the object 6083 and the profile deceleration is defined by the object 6084. The profile type can also be selected by the object 6086.

3.13 - MANUFACTURER DEVICE SPECIFIC

3.13.1 - OBJECT DICTIONARY ENTRIES

Index	Object	Name	Type	Attr.
Error and Stop Function				
3020	VAR	Drive Error Code	Unsigned32	ro
3021	VAR	Warning Code	Unsigned16	ro
3330	VAR	Fault Reaction Mask 1	Unsigned16	rw
3331	VAR	Fault Reaction Mask 3	Unsigned16	rw
3300	ARRAY	Stop 1 Parameter	Unsigned32	rw
3301	VAR	Stop 3 Parameter	Unsigned16	rw
305A	VAR	Inhibit Stop option code	Integer16	rw
3302	VAR	Inhibit Current limit	Unsigned16	rw
3310	VAR	Low Speed Threshold	Unsigned32	rw
3320	ARRAY	Motor Temperature Sensor Threshold	Unsigned16	rw
3321	VAR	Motor Temperature Monitor	Unsigned16	ro
Current Function				
30D0	VAR	Current Offset	Integer16	rw
30D1	VAR	Current Limitation	Integer16	rw
30D4	VAR	Current Monitor	Integer16	ro
30D5	VAR	I^2t Monitor	Integer16	ro
30E0	VAR	Voltage Monitor	Integer16	ro
Inputs/Outputs				
30FD	VAR	Digital Input Bitmask	Unsigned16	rw
30C0	VAR	Analog input 1 set point selection	Unsigned16	rw
30C1	VAR	Analog Input 1 voltage	Integer16	ro
30C2	VAR	Analog Input 1 low pass filter	Unsigned16	rw
30C3	VAR	Analog Input 2 voltage	Integer16	ro
30C4	VAR	Analog input 1 offset compensation	Unsigned32	wo
Position Control Function				
3210	VAR	Raw Position	Integer32	ro
3211	VAR	Internal Position Offset	Integer32	rw
3212	VAR	Motor Position	Integer32	ro
3213	VAR	Second Sensor Position	Integer32	ro
3360	VAR	Axis Type (Rotative/Linear)	Unsigned8	rw
3361	VAR	Software position limit operation	Unsigned8	rw
3350	VAR	Absolute 16 bit mode	Unsigned8	rw
3217	VAR	Position reference error flag	Unsigned16	rw
Absolute encoder feedback				
3215	VAR	Absolute position Reset procedure	Unsigned32	wo
3216	ARRAY	Encoder Position	Integer 32	ro
3218	VAR	Absolute position range	Unsigned32	ro
3219	VAR	Absolute encoder position offset	Unsigned32	rw
Servo On/Off Timing Function				
3304	VAR	Amplifier Reaction Delay	Unsigned16	rw
3305	VAR	Brake Reaction Delay	Unsigned16	rw
Position Capture				
3389	VAR	Capture Input Selection	Unsigned16	rw
3380	VAR	Capture Configuration	Unsigned16	rw
3381	VAR	Capture Status	Unsigned16	ro
3382	VAR	Capture PositionFilter	Unsigned32	rw
338A	VAR	Capture Time Filter	Unsigned16	rw
3383	VAR	Capture 1 Position	Integer32	ro
3384	VAR	Capture 2 Position	Integer32	ro
3385	VAR	Single shot Capture Command (trigger)	Unsigned8	rw
3386	VAR	Single shot Capture Status	Unsigned16	ro
3387	VAR	Single shot Capture 1 Position	Integer32	ro
3388	VAR	Single shot Capture 2 Position	Integer32	ro
TPDO3 Control				
33A0	VAR	TPDO3 Count	Unsigned8	rw
33A1	VAR	TPDO3 Control	Unsigned8	rw

Auto-Phasing				
3410	VAR	Auto-phasing	Unsigned32	wo
3414	VAR	Motor phasing	Unsigned32	wo
Auto-Tuning				
3421	VAR	Auto-tuning Time Interval Select	Unsigned16	rw
3430	VAR	Auto-tuning Application Requirements	Unsigned16	rw
3420	RECORD	Auto-tuning		rw
Encoder Output				
3487	RECORD	Encoder Emulation Output		rw
Cogging torque compensation				
3491	VAR	Enable/Disable cogging torque compensation	Unsigned16	rw
3492	VAR	Cogging torque acquisition procedure	Unsigned32	wo
SinCos inputs calibration				
3496	VAR	SinCos inputs error compensation procedure	Unsigned32	wo

3.13.2 - ERROR AND STOP FUNCTION

3.13.2.1 - Object 3020h: Drive Error Code

Index	**3020h**
Name	Drive Error Code
Object Code	VAR
Data Type	Unsigned32
Object Class	all
Access	ro

See *section 2.1.7* for the meaning of this object.

3.13.2.2 - Object 3021h: Warning Code

Index	**3021h**
Name	Warning Code
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	ro

Bit	**Meaning**
1	I^2t warning
3	Position Following
9	Limit Switch -
10	Limit Switch +
11	Drive over-temperature warning
13	Motor over-temperature warning

3.13.2.3 - Object 3330h: Fault Reaction Mask 1

This object defines which errors (when the pertaining bit is set) will trigger a stop 1.

Index	**3330h**
Name	Fault Reaction Mask 1
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0000h

Bit	Fault
0	Position Limits
1	I^2t
2	RDC
3	Following Error
4	EEPROM
5	Synchro_CAN
6	LowSpeed
7	Procedure
8	Current_Offset
13	Temp. Motor
14	Init_400V
15	Over_Current

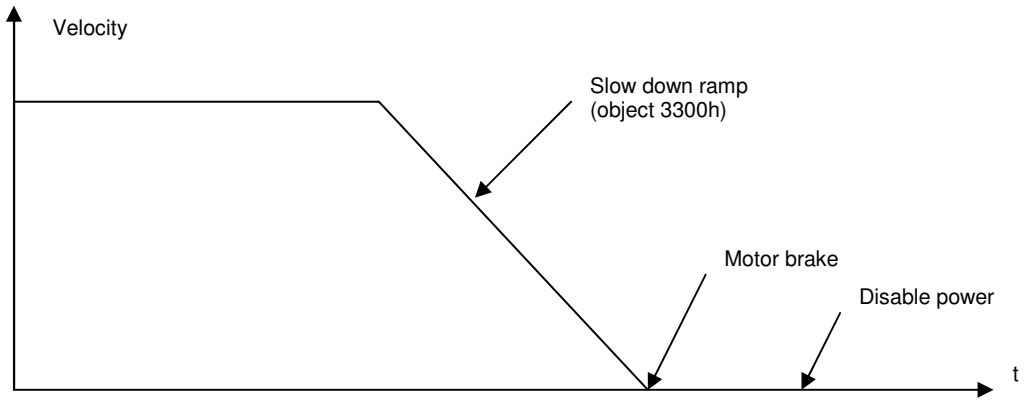
3.13.2.4 - Object 3331h: Fault Reaction Mask 3

This object defines which errors (when the pertaining bit is set) will trigger a stop 3.

Index	3331h
Name	Fault Reaction Mask 3
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0041h

3.13.2.5 - Object 3300h: Stop 1 Parameter

With stop 1, the motor is slowed down in position mode with a slow down ramp.
There are 2 values for the stop 1 ramp: one value when the motor is moving in the positive direction and one value when the motor is moving in the negative direction. This gives the possibility to optimise the stopping time for a vertical axis. For a horizontal axis, these ramps should be the same.



Index	3300h
Name	Stop 1 Parameter
Object Code	ARRAY
Number of Elements	2

Value Description

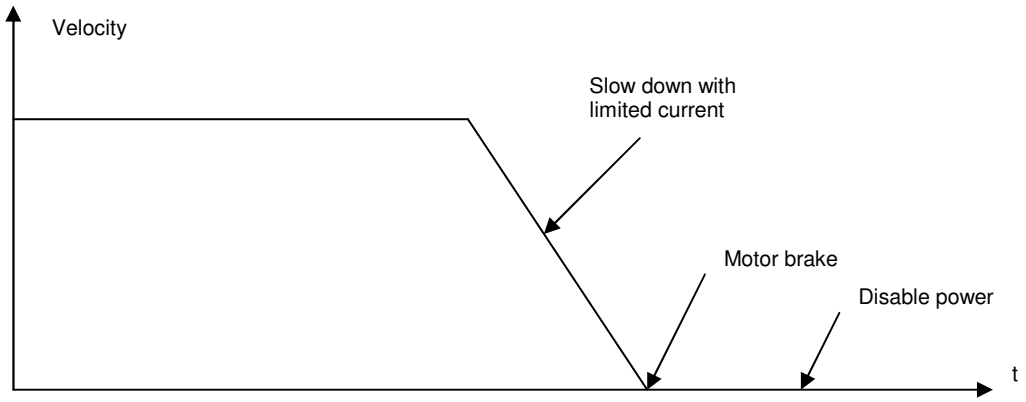
Sub Index	1
Description	Stop 1 Ramp (positive direction)
Data Type	Unsigned32
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Unit	inc/s^2
Default Value	0003D090h

Sub Index	2
Description	Stop 1 Ramp (negative direction)
Data Type	Unsigned32
Object Class	pp ip hm pv eg
Access	rw
PDO Mapping	No
Unit	inc/s^2
Default Value	0003D090h

3.13.2.6 - Object 3301h: Stop 3 Parameter

Index	**3301h**
Name	Stop 3 Parameter
Object Code	VAR
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
Unit	Percentage of amplifier rating (3FFFh = 100%)
Default Value	3FFFh

With stop 3, the motor is slowed down in velocity loop with a current limitation.



3.13.2.7 - Object 305Ah: Inhibit Stop Option Code

Index	**305Ah**
Name	Inhibit Stop Option Code
Object Code	VAR
Data Type	integer16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	3

Quick stop option code	Action
0	Disable drive function
1	Slow down on slow down ramp
3	Slow down on current limit

The slow down ramp is defined in object 3300h.
The slow down current limit is defined in object 3302h.

3.13.2.8 - Object 3302h: Inhibit Current limit

Index	3302h
Name	Inhibit Current limit
Object Code	VAR
Data Type	Unsigned16
Object Class	pp ip hm pv eg
Access	rw
Unit	Percentage of amplifier rating (3FFFh = 100%)
Default Value	3FFFh

If "Inhibit Stop Option Code" = 3 (see object 305A), when the INHIBIT input is activated, the motor is slowed down in velocity loop with this current limitation.

3.13.2.9 - Object 3310h: Low Speed Threshold

The Low Speed Threshold defines the maximum speed authorized in low speed mode (LOW SPEED Input is activated). If the motor speed is higher, a LowSpeed error will occur and the motor will be slowed down with the defined stop.

Index	3310h
Name	Low Speed Threshold
Object Code	VAR
Data Type	Unsigned32
Object Class	all
Access	rw
PDO Mapping	No
Unit	inc/s

3.13.2.10 - Object 3320h: Motor Temperature Sensor

This object defines the parameters for temperature error detection by a motor temperature sensor.

Index	3320h
Name	Motor Temperature Sensor
Object Code	ARRAY
Number of Elements	3

Value Description

Sub Index	1
Description	Sensor Type (CTN/CTP)
Data Type	Unsigned8
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

This parameter defines the sensor type for error detection: 0 means CTN and 1 means CTP.

Sub Index	2
Description	Motor Temperature Sensor Error Threshold
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

This parameter defines the threshold of the equivalent resistor corresponding to the temperature at which an error will be triggered.
The threshold value is defined below:

Threshold = 40958.5 x Rst / (1.2 + Rst)

Rst is the equivalent resistance, in kΩ, of this temperature value.

Sub Index	3
Description	Motor Temperature Sensor Warning Threshold
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

This parameter defines the threshold of the equivalent resistor corresponding to the temperature at which a warning will be notified.
The threshold value is defined below:

Threshold = 40958.5 x Rst / (1.2 + Rst)

Rst is the equivalent resistance, in kΩ, of this temperature value.

3.13.2.11 - Object 3321h: Motor Temperature Sensor Monitor

Index	**3321h**
Name	Motor Temperature Sensor Monitor
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	ro
PDO Mapping	possible

The returned value gives an image of the equivalent resistance with

R = 1.2 x [3321h] / (40958.5 - [3321h])

The temperature corresponding to this resistance depends on the temperature sensor specifications.

3.13.3 - CURRENT FUNCTIONS

3.13.3.1 - Object 30D0h: Current Offset

This parameter defines an offset for the current output. This is useful for a vertical axis.

Index	**30D0h**
Name	Current Offset
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	rw
PDO Mapping	Possible
Unit	Percentage of amplifier rating (3FFFh = 100%)
Value Range	0-2000h
Default Value	0

3.13.3.2 - Object 30D1h: Current Limitation

This parameter defines the dynamic limitation of the current output.

Index	30D1h
Name	Current Limitation
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	rw
PDO Mapping	Possible
Unit	Percentage of amplifier Max Current (3FFFh = 100%)
Default Value	3FFFh

3.13.3.3 - Object 30D4h: Current Monitor

This object is the motor current monitor. This monitor is calculated with vector transformation. This means that the current monitor is correct only if the motor position is correct (no motor sensor error...).

Index	30D4h
Name	Current Monitor
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	ro
PDO Mapping	Possible
Unit	Percentage of amplifier rating (3FFFh = 100%)

3.13.3.4 - Object 30D5h: I²t Monitor

This object is used for the calculation of the motor RMS current value integrated during the last 4 seconds .

Index	30D5h
Name	I²t Monitor
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	ro
PDO Mapping	Possible
Unit	Percentage of amplifier rating (3FFFh = 100%)

The motor RMS current value in % of the amplifier rating is calculated according to the following formula:
RMS motor current (%) = $[(I^2t \text{ monitor value } (\%)) \times 50]^{1/2}$ = $[\text{value(30D5h)} \times 5000 / 16384]^{1/2}$

3.13.3.5 - Object 30E0h: Voltage Monitor

This object is the motor voltage monitor.
Data conversion: 1000h -> 100V for TTA-400/I-CAN amplifier
2000h -> 100V for TTA-230/I-CAN amplifier

Index	30E0h
Name	Voltage Monitor
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	ro
PDO Mapping	Possible

3.13.4 - INPUTS/OUTPUTS

3.13.4.1 - Object 30FDh: Digital Inputs Bitmask

Index	30FDh
Name	Digital Input Bitmask
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	003Fh

By default, the digital inputs of the drive are dedicated inputs with special function. All inputs can be dissociated from their function in order to be used as a simple digital input. The status of these inputs can be read by object 60FDh.

Bit	Meaning
0	INPUT0 / Limit Switch +
1	INPUT1 / Limit Switch -
2	INHIBIT
3	INPUT3 / INDEX
4	INPUT4 / CAPTURE
5	INPUT5 / LOW SPEED

3.13.4.2 - Object 30C1h: Analog input 1 voltage

Index	30C1h
Name	Analog input 1 voltage
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	ro
PDO Mapping	possible
Conversion	+/- 7FFF corresponds to +/- 10V

Note: The analog inputs 1 and 2 must be selected by the 3 jumpers OUT/IN-SEL located in the amplifier connector board (see TTA-CAN manual).

3.13.4.3 - Object 30C2h: Analog input 1 low pass filter

Index	30C2h
Description	Analog input 1 Low-pass filter Defines the cut-off frequency at -3 dB (Fev) of the first order filter that acts upon the analog input 1.
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Conversion	Cut-off frequency (Hz) = 1000/pi*Ln(65536/[30C2-0])
Value Range	0B10h (1000 Hz) et F069h (20 Hz)
Default Value	eeprom

3.13.4.4 - Object 30C4h: Analog input 1 offset compensation

Writing to this object will execute the offset compensation procedure on analog input 1.

Index	30C4h
Name	Offset compensation procedure
Object Code	VAR
Data Type	Unsigned32
Object Class	all
Access	wo
PDO Mapping	No

In order to avoid running the offset compensation procedure by mistake, the offset compensation is only executed when a specific signature is written to the this object. This signature is 'offa'.

Signature	MSB			LSB
ASCII	a	f	f	o
hex	61h	66h	66h	6Fh

3.13.4.5 - Object 30C0h: Analog input 1 set point selection

The velocity set point in the Profile velocity mode can be selected either from the CAN bus (see object 60FFh) or from the Analog input 1 voltage value (10 V for maximum speed).
The torque set point in the Profile torque mode can be selected either from the CAN bus (see object 6071h) or from the Analog input 1 voltage value (10 V for maximum current).

Index	30C0h
Name	Analog input 1 set point selection
Object Code	VAR
Data Type	Unsigned16
Object Class	pv, pt
Access	rw
PDO Mapping	No
Default Value	0

Value	Description
0	CAN bus selection for the Profile velocity and Profile torque set point
1	Analog input 1 selection for the Profile velocity and Profile torque set point

3.13.4.6 - Object 30C3h: Analog input 2 voltage

Index	30C3h
Name	Analog input 2 voltage
Object Code	VAR
Data Type	Integer16
Object Class	all
Access	ro
PDO Mapping	possible
Conversion	+/- 7FFF correspond to +/- 10V

Note: The analog inputs 1 and 2 must be selected by the 3 jumpers OUT/IN-SEL located on the amplifier connector board (see TTA-CAN manual).

3.13.5 - POSITION CONTROL FUNCTION

The motor position (object 6064h) is calculated as shown below:

Pos = RawPos - Offset

Offset (object 3211h) is a value calculated by the homing procedure.

RawPos (object 3210h) is the position given by the internal counter of the drive. The lowest bits of RawPos allow to calculate the position over one motor revolution; this value is absolute over one revolution.

Rev_Pos = RawPos modulo [Revolution_increments]
Revolution_increments = object 608F sub-index 1

Example: Revolution_increments = 4096, the *Rev_Pos* position will be the lowest 12 bits.

When the homing is made, the backup of the motor position before the logic supply (24 V for the TTA-CAN) shuts down, allows, in some conditions, to get back the motor position without homing, as described below:

Example: *Pos1* (object 6064h) is the motor position and *Offset1* (object 3211h) is the offset value before the logic supply shut-down. *Pos1* value is absolute when the homing is made. *Pos1* and *Offset1* are saved.

When the logic supply is on again, *Pos2* (object 6064h) is the motor position. When the homing is not made, Offset2 is equal to 0.

So, the following calculation can be made:

Res1 = (Pos1 - Offset1) modulo [Revolution_increments]
Res2 = Pos2 modulo [Revolution_increments]

The motor shifts when the drive is out of logic supply:

Motor_Shift = (Res2 – Res1) modulo [Revolution_increments]

In the example above, where the *Revolution_increments* = 4096, this substraction must be done only with 12 bits and a sign extension in order to get the result on 32 bits. The result can be positive or negative.

The motor shift calculation when the drive is out of logic supply is possible as the *Res1* and *Res2* are physically absolute values. But *Res1* and *Res2* are only absolute over one revolution and the correct motor shift must not exceed 1/2 rev. or this calculation is not correct.
It is important to define a threshold for an error if the motor shift is large. But there is no way to know if the motor has moved an exact number of revolutions when the drive is out of logic supply.

The motor position we can find is:
Pos = Pos1 + Motor_Shift

To get this value from object 6064, it is necessary to define the position offset as follows:
Offset2 = Pos2 – (Pos1 + Motor_Shift)

Remarks:
- Object 3210h is only for information, it is not necessary to calculate the offset.
- To make the above calculation, it is necessary to back-up the internal offset (object 3211h) after the homing procedure and the motor position (object 6064h) before the logic supply shuts down.

3.13.5.1 - Object 3210h: Internal Raw Position

This object returns :
- the position value of the motor sensor (resolver or encoder) if the motor sensor is selected for position loop,
- the position value of the secondary sensor if the second sensor is selected for the position loop.

Index	**3210h**
Name	Internal Raw Position
Object Code	VAR
Data Type	Integer32
Object Class	All
Access	ro
PDO Mapping	No
Unit	Inc (Revolution increments : object 608F sub-index 1)

3.13.5.2 - Object 3211h: Internal Position Offset

Index	**3211h**
Name	Internal Position Offset
Object Code	VAR
Data Type	Integer32
Object Class	All
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Default Value	0

3.13.5.3 - Object 3212h: Motor Position

Index	**3212h**
Name	Motor Position This is the value of the position sensor on the motor (resolver or encoder).
Object Code	VAR
Data Type	Integer32
Object Class	All
Access	ro
PDO Mapping	Possible
Unit	Inc (Revolution increments: object 608F sub-index 1)

3.13.5.4 - Object 3213h: Second Sensor Position

Index	**3213h**
Name	Second Sensor Position This value is the position of the secondary sensor.
Object Code	VAR
Data Type	Integer32
Object Class	All
Access	ro
PDO Mapping	Possible
Unit	Inc 65536 increments by resolver shaft resolution x scaling factor (object 306C) for a resolver encoder input resolution (object 308F) x 4 x scaling factor (object 306C) for an encoder

Remark: for a SIN/COS encoder type the scaling factor allows to adjust the interpolation factor (from 1 to 1024) for the second sensor position display.

3.13.5.5 - Object 3360h: Axis Type

This parameter defines the axis type: linear or rotative.
A linear axis has the software position limit active. For a rotary axis, the software position limit is inactive.

Index	3360h
Name	Axis Type
Object Code	VAR
Data Type	Unsigned8
Object Class	ip pp hm
Access	rw
PDO Mapping	No
Value Range	0=rotative, 1=linear
Default Value	0

3.13.5.6 - Object 3361h: Software position limit operation

This parameter defines the operation mode of the software position limits when they are active (see object 3360).

Index	3361h
Name	Software position limit operation
Object Code	VAR
Data Type	Unsigned8
Object Class	ip pp hm
Access	rw
PDO Mapping	No
Value Range	0=error mode, 1=locked mode
Default Value	0

Software position limit operation in the "error mode" :
When the motor reaches the software position limit, an error occurs. To move out of the limit, the controller can: - set the bit "out of limit" in control word which authorizes to move the motor.
- or change the "axis type" to rotary.

Software position limit operation in the "locked mode":
When the motor reaches the software position limit, it starts braking. The slow down current limit is defined in object 3302h. When the 0 speed is reached, the motor remains enabled at standstill. To move out of the limit, the controller must generate a displacement in the opposite direction.

Remark : The hardware position limits (Limit Switch - and Limit Switch +) are operating in "locked mode".

3.13.5.7 - Object 3350h: Absolute 16-bit Mode

Index	3350h
Name	Absolute 16-bit mode
Object Code	VAR
Data Type	Unsigned8
Object Class	ip
Access	rw
PDO Mapping	No
Value Range	0..1
Default Value	0

The position reference in interpolated position mode can be defined as 16-bits only. This is to reduce the bus traffic.
When in 16-bit mode (object 3350h = 1), the position reference in object 60C1-1 via PDO is set at 16 bits and the drive calculates the upper word. At the beginning, it is necessary to set the upper word with object 60C1-1 via SDO (Integer32). The mapping of RPDO3 must be changed to object 60C1 sub-index 1 with 16-bit length.

3.13.5.8 - Object 3217h: Position reference error flag

Index	3217h
Name	Position reference error flag
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	1

The position reference error flag value is set at 1 at the logic board power up (24V). When the drive position reference for the position loop is lost, due to a position loop sensor error or a position loop couting error, the position reference error flag value is set as described below.

Bit	Meaning
0	Logic board power up
2	Position loop sensor counting error
5	Position loop sensor error

This object can be reset at 0 by the host controller when the homing procedure has been executed. Then, its value can be monitored in order to detect the loose of the position reference or the amplifier 24V power down.

3.13.6 - SERVO ON/OFF TIMING FUNCTION

The drive can be enabled or disabled by the control word (Switch On/Off). The timing sequence and the timing of the brake can be defined by objects 3304h and 3305h.

3.13.6.1 - Object 3304h: Amplifier Reaction Delay

Index	3304h
Name	Amplifier Reaction Delay
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Unit	ms
Default Value	0

This parameter defines the delay between the servo control by the control word and the effective servo loop output (PWM output). The **TTA-CAN** drive hardware requires about 45 ms to be active.

3.13.6.2 - Object 3305h: Motor Brake Reaction Delay

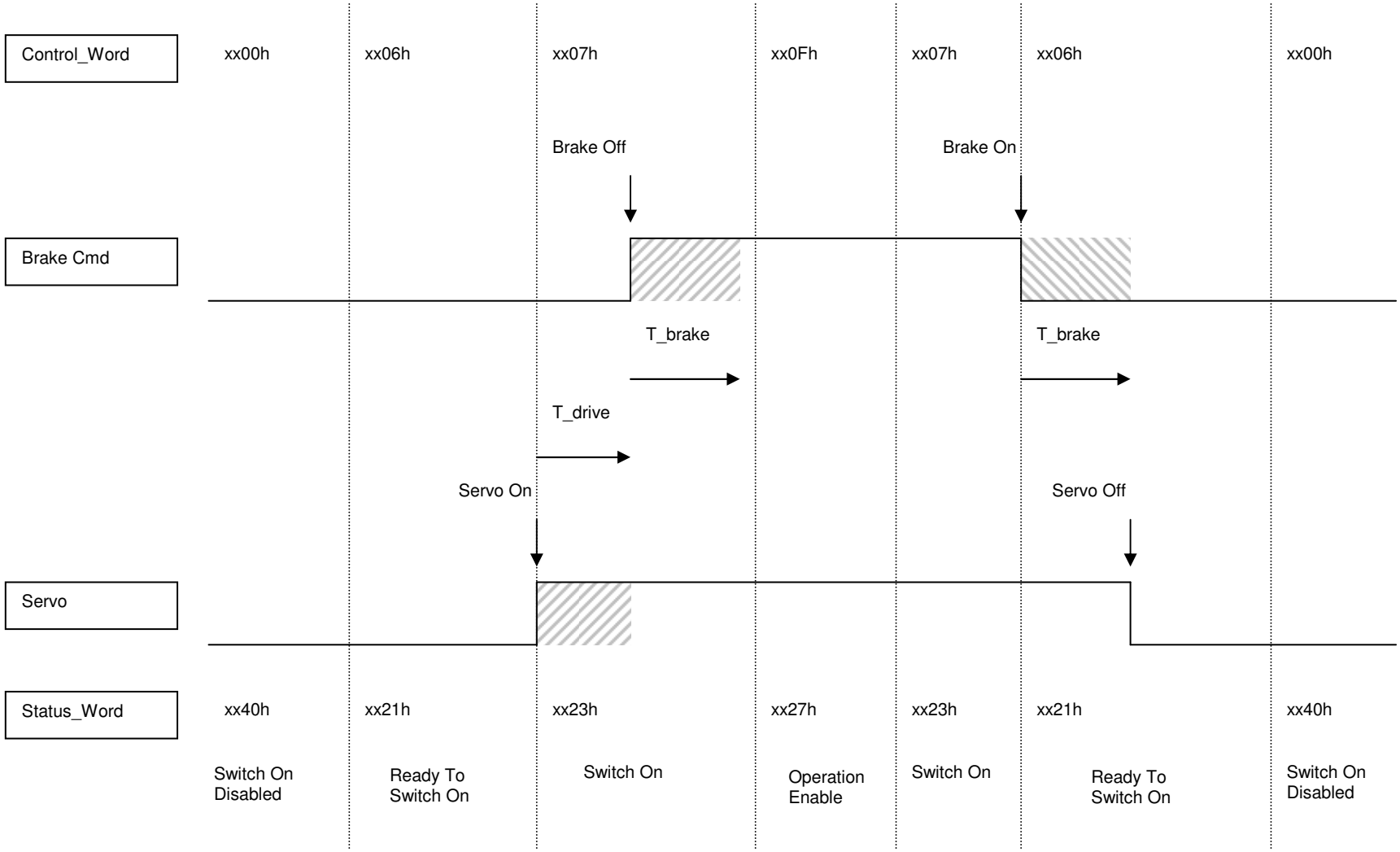
Index	3305h
Name	Motor Brake Reaction Delay
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Unit	ms
Default Value	0

This parameter defines the delay between the brake control and the effective motor brake opening or closing.

The control of the motor brake is automatic with Switch On/Off by the control_word. To disable the motor brake control, it is necessary to set at 1 bit 0 of object 60FE sub-index 2 (digital output bitmask). The motor brake is then manually controlled by bit 0 of object 60FE sub-index 1.

If there is no brake on the motor, it is necessary to disable the automatic control to avoid an error.

Servo On/Off Timing Diagram



3.13.7 - POSITION CAPTURE FUNCTION

The drive can capture the motor position and the secondary sensor position at a rate of 62,5 µs with 2 logic inputs : the CAPTURE input and the INDEX input .

The chart below defines the Main capture position value and the Secondary capture position value according to the sensor feedback configuration.

Configuration	Motor feedback	Position loop feedback	Main capture	Secondary capture
1	Resolver	Resolver	Resolver (object 6064)	Encoder (object 3213)
2	Resolver	Encoder	Encoder (object 6064)	Resolver (object 3212)
3	Encoder	Encoder	Encoder (object 6064)	Resolver (object 3213)
4	Encoder	Resolver	Resolver (object 6064)	Encoder (object 3212)

3.13.7.1 - Object 3389h: Capture Input Selection

Index	**3389h**
Name	Capture Input Selection
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	10h

Bit	**Meaning**
0-3	Capture 1 input selection: 0=CAPTURE input, 1=INDEX input
4-7	Capture 2 input selection: 0=CAPTURE input, 1=INDEX input
8-11	Capture 1 sensor selection: 0=Main capture, 1=Secondary capture
12-15	Capture 2 sensor selection: 0=Main capture, 1=Secondary capture

3.13.7.2 - Object 3380h: Capture Configuration

Index	**3380h**
Name	Capture Configuration
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0

Bit	**Meaning**
0	Capture 1 at rising edge
1	Capture 1 at falling edge
2	Capture 2 at rising edge
3	Capture 2 at falling edge

3.13.7.3 - Object 3381h: Capture Status

Index	3381h
Name	Capture Status
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	ro
PDO Mapping	Possible (16 bit length or 8 bit length)
Default Value	0

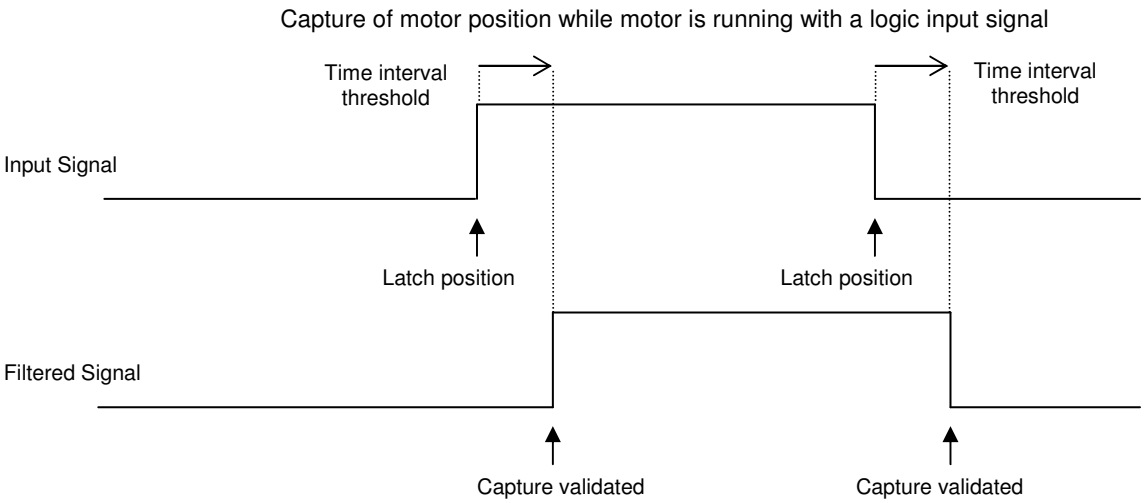
Bit	Meaning
0	Capture at rising edge of input 1 has occurred
1	Capture at falling edge of input 1 has occurred
2	Capture at rising edge of input 2 has occurred
3	Capture at falling edge of input 2 has occurred
4	Capture 1 signal
5	Capture 2 signal
8	Limit Switch +
9	Limit Switch -
10	INHIBIT
11	INDEX
12	CAPTURE
13	LOW SPEED

If object 3383h is mapped in a PDO3 (triggered by SYNC event) or PDO1 (triggered by changed value), then bits 0 and 1 are cleared as soon as the captured position value is transmitted by the PDO. Reading object 3383h by SDO does not clear bits 0 and 1.

If object 3384h is mapped in a PDO3 (triggered by SYNC event) or PDO1 (triggered by changed value), then bits 2 and 3 are cleared as soon as the captured position value is transmitted by the PDO. Reading object 3384h by SDO does not clear bits 2 and 3.
Bits 4 and 5 are status of the capture signal after filtering (see object 3382h below). These status before filtering are monitored in bits 11 and 12.

3.13.7.4 - Object 338Ah: Capture Time Filter

This parameter defines the time interval threshold of the capture time filter. After the rising or the falling edge of the input signal, the input signal level must be stable for a time interval value higher or equal to the time interval threshold defined by object 338Ah in order to get the position capture validated as described below.

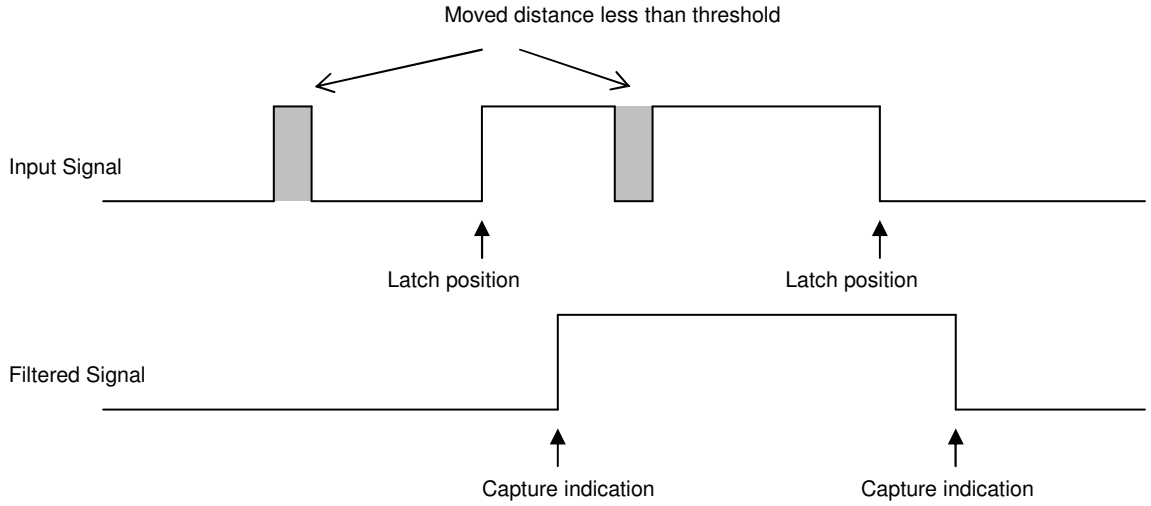


Index	338Ah
Name	Capture Time Filter
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Unit	Sampling periods (62,5 µs)
Value Range	0 to 32767 (2s)
Default Value	6 (375 µs)

3.13.7.5 - Object 3382h: Capture Position Filter

This parameter defines the value in distance threshold of the capture position filter. If the difference of position between rising and falling edges is less than the threshold, then the signal is the following:

Capture of motor position while motor is running with a logic input signal



Index	3382h
Name	Capture Position Filter
Object Code	VAR
Data Type	Unsigned32
Object Class	all
Access	rw
PDO Mapping	No
Unit	Inc (Revolution increments: object 608F sub-index 1)
Value Range	0..00007FFFh
Default Value	0

3.13.7.6 - Object 3383h: Capture 1 Position

This object returns the position latched by input 1 at rising or/and falling edge, as defined in the capture configuration (object 3380h).

Index	3383h
Name	Capture 1 Position
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	Possible (16-bit - LSW)
Unit	inc

3.13.7.7 - Object 3384h: Capture 2 Position

This object returns the position latched by input 2 at rising or/and falling edge as defined in the capture configuration (object 3380h).

Index	3384h
Name	Capture 2 Position
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	Possible (16-bit LSW)
Unit	inc

3.13.7.8 - Object 3385h: Single Shot Capture Command (trigger)

This parameter allows to enable a single shot position capture by setting the corresponding bit to 1. The single shot position capture to be enabled must be previously selected in the capture configuration (object 3380h). When a single shot position capture has been performed, the corresponding bit is automatically resetted.

Index	3385h
Name	Single Shot Capture Command (trigger)
Object Code	VAR
Data Type	Unsigned8
Object Class	all
Access	rw
PDO Mapping	Yes
Default Value	0

Bit	Meaning
0	Single shot capture 1 at rising edge (write 1 to enable)
1	Single shot capture 1 at falling edge (write 1 to enable)
2	Single shot capture 2 at rising edge (write 1 to enable)
3	Single shot capture 2 at falling edge (write 1 to enable)

3.13.7.9 - Object 3386h: Single Shot Capture Status

This parameter displays the single shot capture status. A status bit (0 to 3) is resetted when the corresponding command bit (object 3385h) is set to 1 (capture enabled). A status bit (0 to 3) is set to 1 when the capture has been performed.

Index	3386h
Name	Single Shot Capture Status
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	ro
PDO Mapping	Possible (16 bit length or 8 bit length)
Default Value	0

Bit	Meaning
0	Single shot capture at rising edge of input 1 has occurred
1	Single shot capture at falling edge of input 1 has occurred
2	Single shot capture at rising edge of input 2 has occurred
3	Single shot capture at falling edge of input 2 has occurred
4	Capture 1 signal
5	Capture 2 signal
8	Limit Switch +
9	Limit Switch -
10	INHIBIT
11	INDEX
12	CAPTURE
13	LOW SPEED

3.13.7.10 - Object 3387h: Single Shot Capture 1 Position

This object returns the position latched by input 1 at rising or/and falling edge, as defined in the single shot capture command (object 3385h).

Index	3387h
Name	Single Shot Capture 1 Position
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	Possible (16-bit - LSW)
Unit	inc

3.13.7.11 - Object 3388h: Single Shot Capture 2 Position

This object returns the position latched by input 2 at rising or/and falling edge as defined in the single shot capture command (object 3385h).

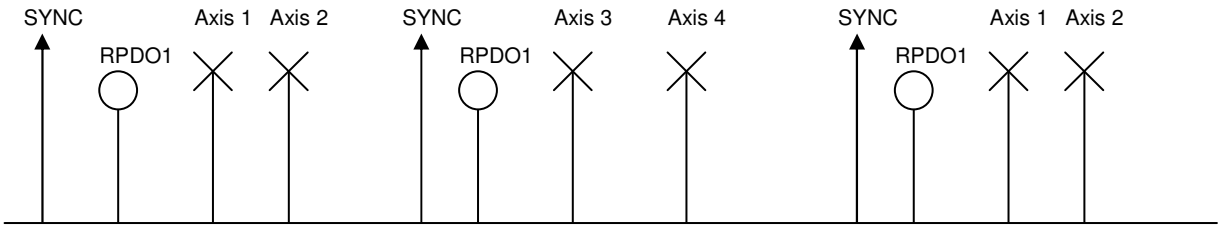
Index	3388h
Name	Single Shot Capture 2 Position
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	Possible (16-bit LSW)
Unit	inc

3.13.8 - TPDO3 CONTROL

The **TTA-CAN** has a special transmission mode for the TPDO3 defined by a TPDO3_Control (object 33A1h) and a TPDO3_Count (object 33A0h). The purpose of this mode is to control the number of cyclic TPDO3 for each axis.

TPDO3_Control is predefined for each axis. TPDO3_Count is counter value of the host. For each axis, when TPDO3_Count is equal to TPDO3_Control, it will transmit the TPDO3 in synchronisation with the SYNC message. The transmission type for the TPDO3 must be 254.

To be sure that all axes have got the same value of TPDO3_Count at the same synchronisation, the RPDO1 COB-ID must be redefined to be the same for all axes and mapped with TPDO3_Count object.



3.13.8.1 - Object 33A0h: TPDO3 Count

Index	**33A0h**
Name	TPDO3 Count
Object Code	VAR
Data Type	Unsigned8
Object Class	all
Access	rw
PDO Mapping	Possible
Default Value	00h

3.13.8.2 - Object 33A1h: TPDO3 Control

Index	**33A1h**
Name	TPDO3 Control
Object Code	VAR
Data Type	Unsigned8
Object Class	All
Access	rw
PDO Mapping	No
Default Value	00h

3.13.9 - PHASING FUNCTION

3.13.9.1 - Object 3410h: Auto-phasing procedure

Writing to this object will execute the auto-phasing procedure. For more information about the auto-phasing procedure, please refer to the TTA-CAN manual.

Index	**3410h**
Name	Auto-phasing procedure
Object Code	VAR
Data Type	Unsigned32
Object Class	-
Access	wo
PDO Mapping	No

In order to avoid running the auto-phasing procedure by mistake, the auto-phasing is only executed when a specific signature is written to the this object. This signature is 'apha'.

Signature	MSB			LSB
ASCII	a	h	p	a
hex	61h	68h	70h	61h

3.13.9.2 - Object 3414h: Motor phasing procedure

Writing to this object will execute the motor phasing procedure. This command starts the motor phasing procedure after the amplifier power up when incremental encoder types without Hall Effect Sensors are used (see TTA-CAN manual). INHIBIT input must be desactived.

Index	**3414h**
Name	Motor phasing procedure
Object Code	VAR
Data Type	Unsigned32
Object Class	-
Access	wo
PDO Mapping	No

In order to avoid running the motor phasing procedure by mistake, the motor phasing is only executed when a specific signature is written to the this object. This signature is 'mcal'.

Signature	MSB			LSB
ASCII	l	a	c	m
hex	6Ch	61h	63h	6Dh

3.13.10 - AUTO-TUNING FUNCTION

3.13.10.1 - Object 3421h: Auto-tuning Time Interval Selection Mode

Index	**3421h**
Name	Auto-tuning Time Interval Selection Mode
Object Code	VAR
Data Type	Unsigned16
Object Class	ip pp hm pv
Access	rw
PDO Mapping	No
Value Range	0..1 0 -> manual setting of "Time Interval Speed Measurement " parameter 1 -> automatic setting of "Time Interval Speed Measurement " parameter by auto-tuning. This parameter must be set before starting the auto-tuning.
Default Value	eeprom

3.13.10.2 - Object 3430h: Auto-tuning Application Requirements

Index	**3430h**
Name	Auto-tuning Application Requirements
Object Code	VAR
Data Type	Unsigned16
Object Class	ip pp hm
Access	rw
PDO Mapping	No
Value Range	0..1 0 -> Minimum tracking error 1 -> Minimum overshoot This parameter must be set before starting the auto-tuning.
Default Value	eeprom

3.13.10.3 - Object 3420h: Auto-tuning procedure

Index	3420h
Name	Auto-tuning procedure
Object Code	RECORD
Number of Elements	2

Value Description

Sub Index	1
Description	Bandwidth
Data Type	Unsigned8
Object Class	-
Access	rw
PDO Mapping	No
Value Range	0..8
Default Value	eeprom

This parameter defines the auto-tuning bandwidth:

Value	Filter	Bandwidth
0	Standard filter	Low Bandwidth
1	Standard filter	Medium Bandwidth
2	Standard filter	High Bandwidth
3	Anti-resonance filter	Low Bandwidth
4	Anti-resonance filter	Medium Bandwidth
5	Anti-resonance filter	High Bandwidth
6	High stiffness filter	Low Bandwidth
7	High stiffness filter	Medium Bandwidth
8	High stiffness filter	High Bandwidth

Sub Index	2
Description	Start auto-tuning procedure
Data Type	Unsigned32
Object Class	all
Access	wo
PDO Mapping	No

In order to avoid running the auto-tuning procedure by mistake, the auto-tuning is only executed when a specific signature is written to the this sub-index. The signature is 'atun'.

Signature	MSB			LSB
ASCII	n	u	t	a
hex	6Eh	75h	74h	61h

3.13.11 – ENCODER OUTPUT

3.13.11.1 - Object 3487h: Encoder Emulation Output

Index	**3487h**
Name	Encoder Emulation Output
Object Code	RECORD
Number of Elements	6

Value Description

Sub Index	1
Description	Encoder Output Resolution
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Value Range	
Default Value	eeprom

The **Encoder output resolution** parameter defines the encoder resolution on channels A and B of the encoder position output for one revolution of the motor shaft. Binary and decimal values are both accepted. The maximum encoder resolution per revolution is limited by the motor speed as shown in the table below:

Maximum motor speed (rpm)	up to 1600	up to 3200	up to 6400	up to 12800	up to 25000
Encoder Output Resolution (ppr)	512 to 16384	512 to 8192	512 to 4096	512 to 2048	512 to 1024

The resolution value defined in the **Encoder output resolution** parameter can be divided by 2, 4 or 8 by the **Encoder output division ratio** parameter.

Sub Index	2
Description	Encoder Output Division Ratio Encoder Division Ratio = $1 / 2^{[3480h]}$ Encoder Output Resolution = Encoder Feedback Resolution x Encoder Division Ratio $= [308Fh] / 2^{[3480h]}$
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Value Range	0..3
Default Value	eeprom
Write Condition	Drive disabled.

Sub Index	3
Description	Encoder Output Deadband
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Value Range	1..4095
Default Value	eeprom

The **Encoder Output deadband** parameter introduces a deadband at standstill around the current motor shaft position in order to avoid oscillations of +/- 1 encoder edge on channels A and B. The value 4095 corresponds to 1/16 of the motor shaft revolution.

Sub Index	4
Description	Encoder Output Zero Pulse Shift
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Value Range	0..32768
Default Value	eeprom

The **Encoder Output Zero pulse shift** parameter allows to shift the marker pulse position on channel Z with regard to the motor zero position given by the position sensor. The value 32767 corresponds to one motor shaft revolution. The marker pulse width is equal to 1/4 of the A and B channels period.

Sub Index	5
Name	Enable / Disable Encoder Output
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	0

Value	**Description**
0	Encoder Output disabled on the TTA-CAN X3 connector
1	Encoder Output enabled on the TTA-CAN X3 connector

At the TTA-CAN amplifier power up, the encoder output is always disabled.

Remarks: The TTL encoder input lines and the encoder output lines are located at the same pins number on the TTA-CAN X3 connector. So, when the encoder output is enabled, the TTL encoder input cannot be used.

Sub Index	6
Description	Encoder Output Programming Procedure
Data Type	Unsigned16
Object Class	all
Access	wo
PDO Mapping	No

The Programming procedure modifies the encoder emulation output according to the parameters defined by object 3086 subindex 1, 2, 3, 4. The procedure is only executed if a 1 is written to this object.

3.13.12 – COGGING TORQUE COMPENSATION (OPTION)

3.13.12.1 - Object 3491h: Enable / Disable Cogging Torque Compensation

The Cogging compensation option available in the TTA-CAN amplifier range allows to cancel the motor cogging torque effects for specific applications where torque accuracy higher than 1 % is required.
The TTA amplifier must be equipped by the factory for getting the cogging compensation option (reference TTA-CAN - U / I - CT). The cogging torque compensation is only valid if the cogging torque acquisition procedure has been performed before (see TTA-CAN manual).

Index	**3491h**
Name	Enable / Disable Cogging Torque Compensation
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Default Value	eeprom

Bit Number	**Description**
8	Cogging Torque Compensation 0 => Cogging Torque Compensation disabled 1 => Cogging Torque Compensation enabled

3.13.12.2 - Object 3492h: Cogging torque acquisition procedure

Writing to this object will execute the cogging torque acquisition procedure. For more information about the cogging torque acquisition procedure, please see TTA-CAN manual.

Index	**3492h**
Name	Cogging torque acquisition procedure
Object Code	VAR
Data Type	Unsigned32
Object Class	-
Access	wo
PDO Mapping	No

In order to avoid running the cogging torque acquisition procedure by mistake, the cogging torque acquisition is only executed when a specific signature is written to the this object. This signature is 'coga'.

Signature	MSB			LSB
ASCII	a	g	o	c
hex	61h	67h	6Fh	63h

3.13.13 - DIGITAL CAM

10 digital cams are available in the TTA-CAN amplifier.

3.13.13.1 - Object 2500h: Cam Configuration Register

Index	**2500h**
Name	Cam Configuration Register
Object Code	RECORD
Number of Elements	4

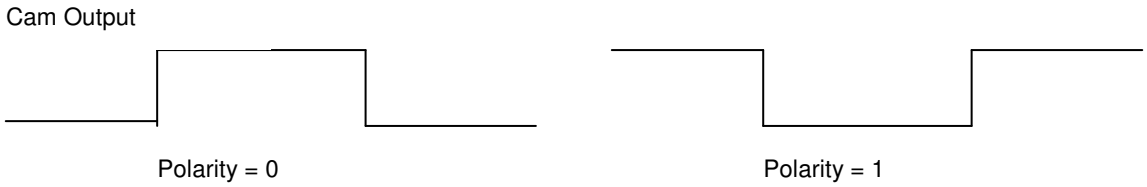
Value Description

Sub Index	1
Description	Cam Enable Register
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Defaut Value	0000h

Each bit (0 to 9) of the Cam Enable Register corresponds to a digital cam. A value 1 allows to enable the corresponding cam.

Sub Index	2
Description	Cam Polarity Register
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Defaut Value	0000h

Each bit (0 to 9) of the Cam Polarity Register allows to set the polarity of the cam output. Normal polarity (polarity bit = 0) sets the cam output with value 1 when the cam is active.

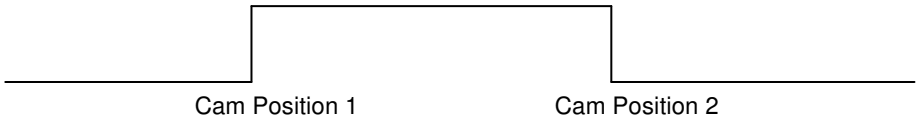


Sub Index	3
Description	Cam Type Register
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Defaut Value	03FFh

Each bit (0 to 9) of the Cam Type Register defines the cam type.

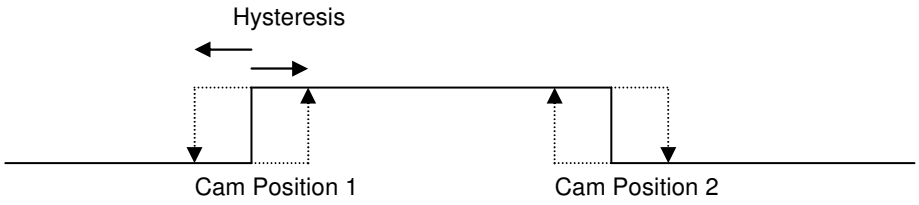
Cam Type = 0: Cam defined by 1 position.

Cam Type = 1: Cam defined by 2 positions.



Sub Index	4
Description	Cam Hysteresis Register
Data Type	Unsigned16
Object Class	all
Access	rw
PDO Mapping	No
Defaut Value	0000h

Cam Hysteresis Register defines a hysteresis of the cam position.



3. 13.13.2 - Object 2501h: Cam Positions

This object defines the 2 positions of each cam (Position 1 < Position 2). For the cam with one position, the second value has no effect.

Index	**2501h**
Name	Cam Configuration Register
Object Code	RECORD
Number of Elements	20

Value Description

Description	**Sub Index**	**Data Type**	**Object Class**	**Access**	**PDO Mapping**
Cam 1 Position 1	1	Integer32	all	rw	No
Cam 1 Position 2	2	Integer32	all	rw	No
Cam 2 Position 1	3	Integer32	all	rw	No
Cam 2 Position 2	4	Integer32	all	rw	No
Cam 3 Position 1	5	Integer32	all	rw	No
Cam 3 Position 2	6	Integer32	all	rw	No
Cam 4 Position 1	7	Integer32	all	rw	No
Cam 4 Position 2	8	Integer32	all	rw	No
Cam 5 Position 1	9	Integer32	all	rw	No
Cam 5 Position 2	10	Integer32	all	rw	No
Cam 6 Position 1	11	Integer32	all	rw	No
Cam 6 Position 2	12	Integer32	all	rw	No
Cam 7 Position 1	13	Integer32	all	rw	No
Cam 7 Position 2	14	Integer32	all	rw	No
Cam 8 Position 1	15	Integer32	all	rw	No
Cam 8 Position 2	16	Integer32	all	rw	No
Cam 9 Position 1	17	Integer32	all	rw	No
Cam 9 Position 2	18	Integer32	all	rw	No
Cam 10 Position 1	19	Integer32	all	rw	No
Cam 10 Position 2	20	Integer32	all	rw	No

3. 13.13.3 - Object 2502h: Cam Status

Index	**2502h**
Name	Cam Status
Object Code	VAR
Data Type	Unsigned16
Object Class	all
Access	ro
PDO Mapping	Possible

Each bit (0 to 9) of Cam Status Register is the output of the corresponding cam.

Important remark:
Cams are fully defined with object 2500h and 2501h. Due to its implementation in TTA-CAN, the *Cam Enable Register* (object 2500, sub-index 1) must be the last one to be written. This should be done each time any register of object 2500h or 2501h is modified.

3.13.14 – ABSOLUTE ENCODER FEEDBACK

3.13.14.1 - Object 3216h: Absolute encoder Position

Index	**3216h**
Name	Encoder Position
Object Code	ARRAY
Number of Elements	3

Value Description

Sub Index	1
Description	Encoder Increments per Revolution
Data Type	Unsigned32
Object Class	all
Access	ro
PDO Mapping	No
Unit	inc
Value Range	200h .. 1000 0000h
Default Value	0

This parameter defines the number of increments for one encoder revolution, in the absolute position channel for a rotary absolute encoder type. For a linear absolute encoder type this parameter is giving the encoder pitch value in nano-metres for the absolute position channel.

Sub Index	2
Description	Encoder Revolutions
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Default Value	0

This parameter defines the number of encoder revolutions for a rotary absolute multi-turn encoder. For a linear encoder or an absolute single-turn encoder this parameter value is equal to 0.

Sub Index	3
Name	Encoder Position Actual Value
Object Code	VAR
Data Type	Integer32
Object Class	all
Access	ro
PDO Mapping	No
Unit	Inc

This object returns the absolute position value, for an absolute encoder type, or the incremental position value, for an incremental encoder type. For an absolute encoder type, this position value is only valid when the encoder is at standstill. When the encoder is moving, this position value is wrong because of the acquisition delay via the encoder communication channel (up to 60 ms).

Remark : The polarity of the encoder position (object 3216 sub 3) depends on the encoder wiring and can be reversed with regard to the polarity of the position loop feedback (object 6064). In this case, if necessary, use the object 607E to reverse the position loop feedback polarity, in order to get the same polarity in both positions.

3.13.14.2 - Object 3215h: Absolute encoder position Reset procedure

Index	3215h
Name	Absolute position Reset procedure
Object Code	VAR
Data Type	Unsigned32
Object Class	-
Access	wo
PDO Mapping	No

Writing to this object will execute the reset procedure for the absolute position. In order to avoid running the absolute position reset procedure by mistake, the procedure is only executed when a specific signature is written to this object.

If the signature is 'razp', the procedure sets at 0 the encoder absolute position value. The absolute position offset value is calculated accordingly. For an absolute rotary encoder, this operation must be executed out of the operation travel range in order to avoid absolute position roll-over.

Signature	MSB			LSB
ASCII	p	z	a	r
hex	70h	7Ah	61h	72h

If the signature is 'setp', the procedure sets the encoder absolute position value at the center of the encoder absolute position range. The absolute position offset value is calculated accordingly. This operation is only valid for an absolute rotary encoder; it must be executed at the middle point of the axis travel range.This adjustment allows to overcome the roll-over phenomena.

Signature	MSB			LSB
ASCII	p	t	e	s
hex	70h	74h	65h	73h

3.13.14.3 - Object 3219h: Absolute encoder Position offset

Index	3219h
Description	Encoder absolute position offset
Data Type	Unsigned32
Access	rw
PDO Mapping	No
Default Value	0

This parameter defines the encoder absolute position offset value. This offset value is calculated when the absolute position reset procedure is executed. The max. value (modulo) for an absolute single-turn encoder is object 3216 sub 1 value. The max. value (modulo) for an absolute multi-turn encoder is object 3216 sub 1 value x object 3216 sub 2 value.The scaling is given by object 3216 sub 1 (number of increments per revolution). This parameter is saved in the amplifier EEPROM.

3.13.14.4 - Object 3218h: Absolute Position Range

Index	3218h
Description	Absolute position range
Data Type	Unsigned32
Access	ro
PDO Mapping	No
Default Value	0

This parameter defines the maximum value for the absolute position according to the user position resolution (object 608F sub index 1) and the encoder revolutions (object 3216 sub index 2) for a rotary absolute encoder. The minimum value for the absolute position is 0. Absolute position control is only possible between these limits. A position roll-over phenomenon occurs when moving beyond these limits. A linear absolute encoder is not concerned by the position roll-over, in this case, the returned value for this parameter is 0xFFFFFFFF (full encoder position range).

3.13.15 – SINCOS INPUTS CALIBRATION

Object 3496h: SinCos inputs error compensation

Index	**3496h**
Name	SinCos inputs error compensation procedure
Object Code	VAR
Data Type	Unsigned32
Object Class	-
Access	wo
PDO Mapping	No

Writing to this object will execute the SinCos inputs error compensation procedure (compensation of the offsets and amplitude gap between the Sin and Cos signals). This procedure is only necessary for low resolution encoders (number of Sin and Cos periods per revolution less than 128) or a SinCos tracks motor feedback sensor. This operation allows to reduce the motor speed ripples at the Sin and Cos signal frequency (number of Sin and Cos signal periods per revolution x motor speed in rpm / 60) and improve the position accuracy. This operation must be renewed if the motor encoder or the SinCos tracks position sensor is exchanged for maintenance. The compensation parameters are automatically saved in the amplifier internal memory at the end of the procedure.

In order to avoid running the SinCos encoder error compensation procedure by mistake, the procedure is only executed when a specific signature is written to this object. This signature is 'scmp'. The procedure must be started with amplifier disabled and **Inhibit** input desactivated.

Signature	MSB			LSB
ASCII	p	m	c	s
hex	70h	6Dh	63h	73h

Appendix 1 - Object Dictionary

COMMUNICATION

Index	Sub-index	Name	Type	Attr.	Default Value	PDO
1000	0	Device type	Unsigned32	ro	00020192h	No
1001	0	Error register	Unsigned8	ro	00h	No
1005	0	COB-ID SYNC	Unsigned32	rw	00000080h	No
1006	0	Communication cycle period	Unsigned32	rw	eeprom	No
3006		Cycle period adjustment				
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	Cycle period adjustment limit	Unsigned16	rw	000Fh	No
	2	Cycle period adjustment parameter 2	Unsigned16	rw	003Ch	No
1007	0	Synchronous Window Length	Unsigned32	rw		No
1008	0	Manufacturer device name	String	ro	TTA-CAN	No
1009	0	Manufacturer hardware version	String	ro		No
100A	0	Manufacturer software version	String	ro		No
100C	0	Guard Time	Unsigned16	rw	00h	No
100D	0	Life Time Factor	Unsigned8	rw	0000h	No
1010		Store Parameters	ARRAY			
	0	Number of supported entries	Unsigned8	ro	01h	No
	1	Save All Parameters	Unsigned32	rw	Signature = 65766173h	No
1014	0	COB-ID EMCY	Unsigned32	rw	00000080h +Node_Id	No
1018		Identity Object	RECORD			
	0	Number of supported entries	Unsigned8	ro	01h	No
	1	Vendor-ID	Unsigned32	ro	00000082h	No
1200		1st Server SDO Parameter	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	COB-ID client => server	Unsigned32	ro	00000600h +Node_Id	No
	2	COB-ID server => client	Unsigned32	ro	00000580h +Node_Id	No
1400		1st Receive PDO Parameter	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	COB-ID	Unsigned32	rw	00000200h +Node_Id	No
	2	transmission type	Unsigned8	rw	FDh	No
1401		2nd Receive PDO Parameter	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	COB-ID	Unsigned32	rw	00000300h +Node_Id	No
	2	transmission type	Unsigned8	rw	FDh	No
1402		3rd Receive PDO Parameter	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	COB-ID	Unsigned32	rw	00000400h +Node_Id	No
	2	transmission type	Unsigned8	rw	01h	No
1600		1st Receive PDO Mapping	RECORD			
	0	Number of supported entries	Unsigned8	rw	01h	No
	1	1st object to be mapped	Unsigned32	rw	60400010h	No
	2	2nd object to be mapped	Unsigned32	rw	00000000h	No
	3	3rd object to be mapped	Unsigned32	rw	00000000h	No
	4	4th object to be mapped	Unsigned32	rw	00000000h	No
1601		2nd Receive PDO Mapping	RECORD			
	0	Number of supported entries	Unsigned8	rw	01h	No
	1	1st object to be mapped	Unsigned32	rw	60FF0020h	No
	2	2nd object to be mapped	Unsigned32	rw	00000000h	No
	3	3rd object to be mapped	Unsigned32	rw	00000000h	No
	4	4th object to be mapped	Unsigned32	rw	00000000h	No
1602		3rd Receive PDO Mapping	RECORD			
	0	Number of supported entries	Unsigned8	rw	01h	No
	1	1st object to be mapped	Unsigned32	rw	60C10120h	No
	2	2nd object to be mapped	Unsigned32	rw	00000000h	No
	3	3rd object to be mapped	Unsigned32	rw	00000000h	No

	4	4th object to be mapped	Unsigned32	rw	00000000h	No
1800		1st Transmit PDO Parameter	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	COB-ID	Unsigned32	rw	00000180h +Node_Id	No
	2	transmission type	Unsigned8	rw	FDh	No
1801		2nd Transmit PDO Parameter	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	COB-ID	Unsigned32	rw	00000280h +Node_Id	No
	2	transmission type	Unsigned8	rw	FDh	No
1802		3rd Transmit PDO Parameter	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	COB-ID	Unsigned32	rw	00000380h +Node_Id	No
	2	transmission type	Unsigned8	rw	01h	No
1A00		1st Transmit PDO Mapping	RECORD			
	0	Number of supported entries	Unsigned8	rw	01h	No
	1	1st object to be mapped	Unsigned32	rw	60410010h	No
	2	2nd object to be mapped	Unsigned32	rw	00000000h	No
	3	3rd object to be mapped	Unsigned32	rw	00000000h	No
	4	4th object to be mapped	Unsigned32	rw	00000000h	No
1A01		2nd Transmit PDO Mapping	RECORD			
	0	Number of supported entries	Unsigned8	rw	01h	No
	1	1st object to be mapped	Unsigned32	rw	606C0020h	No
	2	2nd object to be mapped	Unsigned32	rw	00000000h	No
	3	3rd object to be mapped	Unsigned32	rw	00000000h	No
	4	4th object to be mapped	Unsigned32	rw	00000000h	No
1A02		3rd Transmit PDO Mapping	RECORD			
	0	Number of supported entries	Unsigned8	rw	01h	No
	1	1st object to be mapped	Unsigned32	rw	60640020h	No
	2	2nd object to be mapped	Unsigned32	rw	00000000h	No
	3	3rd object to be mapped	Unsigned32	rw	00000000h	No
	4	4th object to be mapped	Unsigned32	rw	00000000h	No

COMMON ENTRIES IN THE OBJECT DICTIONARY

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
6402	0	Motor Type	Unsigned16	ro	0003h	No
6410		Motor Data	RECORD			
	0	Number of supported entries	Unsigned8	ro	06	No
	1	Motor Id	Unsigned16	rw	eeprom	No
	2	Number of Pole Pairs	Unsigned16	rw	eeprom	No
	3	Motor Phase	Unsigned16	rw	eeprom	No
	4	Position Sensor Offset	Unsigned16	rw	eeprom	No
	5	Current Phase Lead	Unsigned16	rw	0	No
	6	Hall Effect Sensors Offset	Unsigned16	rw	eeprom	No
6502	0	Supported Drive Modes	Unsigned32	ro	0000006Dh	No
6504	0	Drive Manufacturer	String	ro	Infranor	No
6510		Drive Data	RECORD			
	0	Number of supported entries	Unsigned8	ro	07h	No
	1	Drive Family Code	Unsigned16	ro	02	No
	2	Drive Option Code	Unsigned16	ro	0000h	No
	3	Voltage	Unsigned16	ro		No
	4	Current Rating	Unsigned16	ro		No
	5	Max Current	Unsigned16	rw	eeprom	No
	6	Rated Current	Unsigned16	rw	eeprom	No
	7	I^2t Mode	Unsigned8	rw	eeprom	No
60FD	0	Digital Inputs	Unsigned32	ro		Possible
60FE		Digital Outputs	ARRAY			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	Physical Outputs	Unsigned32	rw	00000000h	Possible
	2	Bitmask	Unsigned32	rw	00000000h	No
30FF	0	Specific Outputs configuration	Unsigned16	rw	0000h	No

DEVICE CONTROL

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
6040	0	Control Word	Unsigned16	rw	0000h	Possible
6041	0	Status Word	Unsigned16	ro		Possible
6060	0	Mode of Operation	Integer8	wo		Possible
6061	0	Mode of Operation Display	Integer8	ro	07h	Possible
605A	0	Quick Stop Option Code	Integer16	rw	0003h	No

POSITION SENSOR GROUP

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
607E	0	Polarity	Unsigned8	rw	00h	No
608F		Position Resolution	ARRAY			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	Revolution Increments	Unsigned32	rw	eeprom	No
	2	Motor Revolutions	Unsigned32	ro	00000001h	No
3080	0	Encoder Input Selection	Unsigned16	rw	eeprom	No
308F	0	Encoder Input resolution	Unsigned32	rw	eeprom	No
3090	0	Encoder Pulse Interpolation	Unsigned16	rw	eeprom	No
3091	0	Encoder Zero mark pitch	Unsigned16	rw	eeprom	No
3070	0	Motor feedback selection	Unsigned16	rw	eeprom	No
3098	0	Motor Feedback Programming	Unsigned32	wo	Signature = 6B626466h	No
306A	0	Second Sensor Feedback	Unsigned16	rw	0000h	No
306C	0	Second Sensor Scaling factor	Unsigned16	rw	1	No

PROFILE POSITION MODE

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
607A	0	Target Position	Integer32	rw	00000000h	Possible
6080	0	Max Motor Speed	Unsigned16	rw	eeprom	No
6081	0	Profile Velocity	Unsigned32	rw	00001000h	Possible
6082	0	End Velocity	Unsigned32	rw	0	Possible
6083	0	Profile Acceleration	Unsigned32	rw	00010000h	No
6084	0	Profile Deceleration	Unsigned32	rw	00010000h	No
6086	0	Motion Profile Type	Integer16	rw	0000h	No
607D		Software Position Limit	ARRAY			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	Min Position Limit	Integer32	rw	80000000h	No
	2	Max Position Limit	Integer32	rw	7FFFFFFFh	No
6067	0	Position Window	Unsigned32	rw	00000000h	No
307F	0	Position Modulo	Unsigned32	rw	00000000h	No

HOMING MODE

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
607C	0	Home Offset	Integer32	rw	00000000h	No
6098	0	Homing Method	Integer8	rw	23h	No
6099		Homing Speeds	ARRAY			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	Speed during search for switch	Unsigned32	rw	00000019h	No
	2	Speed during search for zero	Unsigned32	rw	0000000Ah	No
609A	0	Homing Acceleration	Unsigned32	rw	00010000h	No
309B	0	Zero Shift	Unsigned16	rw	eeprom	No
309C	0	Homing Current Limit	Unsigned16	rw	0400h	No

POSITION CONTROL FUNCTION

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
6063	0	position actual value*	Integer32	ro		No
6064	0	position actual value	Integer32	ro		Possible Integer32 Integer16
6065	0	Following error window	Unsigned32	rw	eeprom	No
3031		Following Error Dynamic Gain	Unsigned16	rw	eeprom	No
3033		Following Error Detection Mode	Unsigned16	rw	eeprom	No
3214	0	Position Error	Integer32	ro		possible
3067	0	Position loop deadband	Unsigned32	rw	0	No
60FB		Position Control Parameter Set	RECORD			
	0	Number of supported entries	Unsigned8	ro	04	No
	1	Proportional Gain	Unsigned16	rw	eeprom	No
	2	Feed Forward Speed 1 Gain	Unsigned16	rw	eeprom	No
	3	Feed Forward Acceleration	Unsigned16	rw	eeprom	No
	4	Feed Forward Speed 2 Gain	Unsigned16	rw	eeprom	No

INTERPOLATED POSITION MODE

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
60C0	0	interpolation submode select	Integer16	rw	0000h	No
60C1		Interpolation data record	RECORD			
	0	Number of supported entries	unsigned8	ro	02h	No
	1	First parameter of ip function	integer32	rw		Possible integer32 integer16
	2	Second parameter of ip function	integer32	rw		Possible
60C4		Interpolation Data Configuration	RECORD			
	0	Number of supported entries	unsigned8	ro	06h	No
	1	Max Buffer Size	unsigned32	ro	2	No
	2	Actual Size	unsigned32	ro	2	No
	3	Buffer Organization	unsigned8	ro	00h	No
	4	Buffer Position	unsigned16	rw	00h	No
	5	Size of Data Record	unsigned8	wo		No
	6	Buffer Clear	unsigned8	wo		No

PROFILE VELOCITY MODE

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
6069	0	Velocity Sensor Actual Value	integer32	ro		No
606A	0	Sensor Selection Code	Integer16	ro	0000h	No
606B	0	Velocity Demand Value	Integer32	ro		No
606C	0	Velocity Actual Value	Integer32	ro		Possible
606D	0	Velocity Window	Unsigned 16	rw	0	No
606E	0	Velocity Window Time	Unsigned 16	rw	0	No
606F	0	Velocity Threshold	Unsigned 16	rw	0	No
6070	0	Velocity Threshold Time	Unsigned 16	rw	0	No
60FF	0	Target Velocity	Integer32	rw	00000000h	Possible
604F	0	Ramp Function	Unsigned32	rw	eeprom	No
60F9		Velocity Control Parameter Set	RECORD			
	0	Number of supported entries	Unsigned8	ro	08	No
	1	Proportional Gain	Unsigned16	rw	eeprom	No
	2	Integral 1 Gain	Unsigned16	rw	eeprom	No
	3	Integral 2 Gain	Unsigned16	rw	eeprom	No
	4	Damping Gain	Unsigned16	rw	eeprom	No
	5	Speed Error Low-pass Filter	Unsigned16	rw	eeprom	No
	6	Anti-resonance filter	Unsigned8	rw	eeprom	No
	7	Derivative Gain	Unsigned16	rw	eeprom	No
	8	Derivative Gain Filter	Unsigned16	rw	eeprom	No
30F9	0	Integrator low frequency limit	Unsigned16	rw	0	No
30FB	0	Regulator Gains Scaling	Unsigned16	rw	eeprom	No
3422	0	Time Interval for Speed Measurement	Unsigned16	rw	eeprom	No
30F5	0	Speed Following Error Limit	Integer16	rw	eeprom	No

PROFILE TORQUE MODE

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
6071	0	Target Torque	Integer16	rw		Possible
6087	0	Torque Slope	Unsigned32	rw		No
6088	0	Torque Profile Type	Integer16	ro	0000h	No
60F6		Torque Control Parameter Set	RECORD			
	0	Number of supported entries	Unsigned8	ro	04h	No
	1	q-Loop Proportional Gain	Unsigned16	rw	eeprom	No
	2	q-Loop Integral Gain	Unsigned16	rw	eeprom	No
	3	d-Loop Proportional Gain	Unsigned16	rw	eeprom	No
	4	d-Loop Integral Gain	Unsigned16	rw	eeprom	No
6078	0	Filtered current monitor	Integer16	ro		Possible
3078	0	Current monitor filter	Unsigned16	rw	4000	No

MANUFACTURER SPECIFIC

Index	Sub-Index	Name	Type	Attr.	Default Value	PDO
2500		Cam Configuration	RECORD			
	0	Number of supported entries	Unsigned8	ro	04	No
	1	Activation	Unsigned16	rw	00 00	No
	2	Polarity	Unsigned16	rw	00 00	No
	3	Type	Unsigned16	rw	00 00	No
	4	Hysteresis	Unsigned16	rw	00 00	No
2501		Cam Position	RECORD			
	0	Number of supported entries	Unsigned8	ro	14	No
	1	Cam 1 Position 1	Integer32	rw		No
	2	Cam 1 Position 2	Integer32	rw		No
	...	...	Integer32	rw		No
	20	Cam 10 Position 2	Integer32	rw		No
3020	0	Drive Error Code	Unsigned32	ro		No
3021	0	Warning Code	Unsigned16	ro		No
305A	0	Inhibit Stop Option Code	Integer16	rw	0003h	No
306A	0	Second Sensor Feedback	Unsigned16	rw	0000h	No
306C	0	Second Sensor Scaling factor	Unsigned16	rw	1	No
30D0	0	Current Offset	Integer16	rw	0000h	Possible
30D1	0	Current Limitation	Integer16	rw	3FFFh	Possible
30D4	0	Current Monitor	Integer16	ro		Possible
30D5	0	I^2t Monitor	Integer16	ro		Possible
30E0	0	Voltage Monitor	Integer16	ro		Possible
30FD	0	Digital Inputs Configuration	Unsigned16	rw	003Fh	No
30C0	0	Analog input 1 set point selection	Unsigned16	rw	0	No
30C1	0	Analog input 1 voltage	Integer16	ro		Possible
30C2	0	Analog input 1 low pass filter	Unsigned16	rw	eeprom	No
30C3	0	Analog input 2 voltage	Integer16	ro		Possible
30C4	0	Analog input 1 offset compensation	Unsigned32	wo	Signature = 6166666Fh	No
3210	0	Raw Position	Integer32	ro		No
3211	0	Internal Position Offset	Integer32	rw	00000000h	No
3212	0	Motor Position	Integer32	ro		Possible Integer32 Integer16
3213	0	Second Sensor Position	Integer32	ro		Possible Integer32 Integer16
3215	0	Absolute encoder reset procedure: -Encoder position reset -Encoder position centre range set	Unsigned32	wo	Signature = 707A6172h Signature = 70746573h	No
3216		Absolute encoder position				
	0	Number of supported entries	Unsigned8	ro	03h	No
	1	Increments per revolution	Unsigned32	ro	0h	No
	2	Revolutions	Unsigned32	ro	0h	No
	3	Position value	Integer32	ro	0h	No
3217	0	Position reference error flag	Unsigned16	rw	1	No
3218	0	Absolute position range	Unsigned32	ro	0h	No
3219	0	Absolute encoder position offset	Unsigned32	rw	0h	No

3300		Stop 1 Parameter	ARRAY			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	Stop 1 Ramp Pos	Unsigned32	rw	0003D090h	No
	2	Stop 1 Ramp Neg	Unsigned32	rw	0003D090h	No
3301	0	Stop 2 Curent Limit	Unsigned16	rw	3FFFh	No
3302	0	Inhibit Curent Limit	Unsigned16	rw	3FFFh	No
3304	0	Amplifier Reaction Delay	Unsigned16	rw	00h	No
3305	0	Brake Reaction Delay	Unsigned16	rw	00h	No
3310	0	Low speed threshold	Unsigned32	rw		No
3320		Motor Temperature Sensor	ARRAY			
	0	Number of supported entries	Unsigned8	ro	03h	No
	1	CTP/CTN	Unsigned8	rw	eeprom	No
	2	Motor Temperature Sensor Error Threshold	Unsigned16	rw	eeprom	No
	3	Motor Temperature Sensor Warning Threshold	Unsigned16	rw	eeprom	No
3321	0	Motor temperature sensor monitor	Unsigned16	ro		possible
3330	0	Fault Reaction Mask 1	Unsigned16	rw	0000h	No
3331	0	Fault Reaction Mask 2	Unsigned16	rw	0041h	No
3350	0	Absolute 16 bit mode	Unsigned8	rw	00h	No
3360	0	Axis Type (rotary/linear)	Unsigned8	rw	00h	No
3361	0	Software position limit operation	Unsigned8	rw	00h	No
3380	0	Capture Configuration	Unsigned16	rw		No
3381	0	Capture Status	Unsigned16	ro	0000h	Possible Unsigned8 Unsigned16
3382	0	Capture Position Filter	Unsigned32	rw	00000000h	No
3383	0	Capture 1 Position	Integer32	ro		Possible Integer16 Integer32
3384	0	Capture 2 Position	Integer32	ro		Possible Integer16 Integer32
3385	0	Single shot Capture Command	Unsigned8	rw	00h	Possible
3386	0	Single shot Capture Status	Unsigned16	ro	0000h	Possible Unsigned8 Unsigned16
3387	0	Single shot Capture 1 Position	Integer32	ro		Possible Integer16 Integer32
3388	0	Single shot Capture 2 Position	Integer32	ro		Possible Integer16 Integer32
3389	0	Capture Input Selection	Unsigned16	rw	10h	No
338A	0	Capture Time Filter	Unsigned16	rw	6	No
33A0	0	TPDO3 Count	Unsigned8	rw	00h	Possible
33A1	0	TPDO3 Control	Unsigned8	rw	00h	No
3410	0	Auto-phasing	Unsigned32	wo	Signature = 61687061h	No
3414	0	Motor Phasing	Unsigned32	wo	Signature = 6C61636Dh	No
3420		Auto-tuning	RECORD			
	0	Number of supported entries	Unsigned8	ro	02h	No
	1	Bandwidth	Unsigned8	rw	eeprom	No
	2	Start	Unsigned32	wo	Signature = 6E757461h	No
3421	0	Auto-tuning Time Interval Selection Mode	Unsigned16	rw	eeprom	No
3430	0	Auto-tuning Application Requirements	Unsigned16	rw	eeprom	No

3487		Encoder Emulation Output	RECORD			
	0	Number of supported entries	Unsigned8	ro	06h	No
	1	Encoder Output Resolution	Unsigned16	rw	eeprom	No
	2	Encoder Output Division Ratio	Unsigned16	rw	eeprom	No
	3	Encoder Output Deadband	Unsigned16	rw	eeprom	No
	4	Encoder Output Zero Pulse Shift	Unsigned16	rw	eeprom	No
	5	Enable/Disable Encoder Output	Unsigned16	rw	0000h	No
	6	Encoder Output Programming Procedure	Unsigned16	wo	1	No
3491	0	Enable/Disable cogging torque compensation	Unsigned16	rw	eeprom	No
3492	0	Cogging torque acquisition procedure	Unsigned32	wo	Signature = 61676F63h	No
3496	0	SinCos inputs error compensation procedure	Unsigned32	wo	Signature = 70606373h	No

Appendix 2 - Quick Start

A2.1 - OVERVIEW

The purpose of this paper is to give the user a quick overview of how to use TRANSTECHNIK Servo Drives.

Main steps :

1. Installation
- Wiring the drive
- Switching on power
- Commissioning

2. Operation
- Starting communication (NMT)
- Setting communication parameters (SDO)
- Starting Synchronisation (SYNC)
- Data Exchange (PDO)
- Servo on / Servo off
- Move on

A2.2 - INSTALLATION

A2.2.1 - WIRING

Please see the "TTA-CAN Installation Guide" for the drive connections.
The INHIBIT input and the LOW SPEED input must be connected to 24 V.

A2.2.2 - POWERING

Always switch on the 24 Volt supply (5 s) before the power supply, otherwise an error will be displayed which cannot be cleared. It is important to wait at least 30 s between switching off and on again the drive.

A2.2.3 - COMMISSIONING

The **TTA-CAN** drives commissioning is made by means of the PC software "VDSetup".

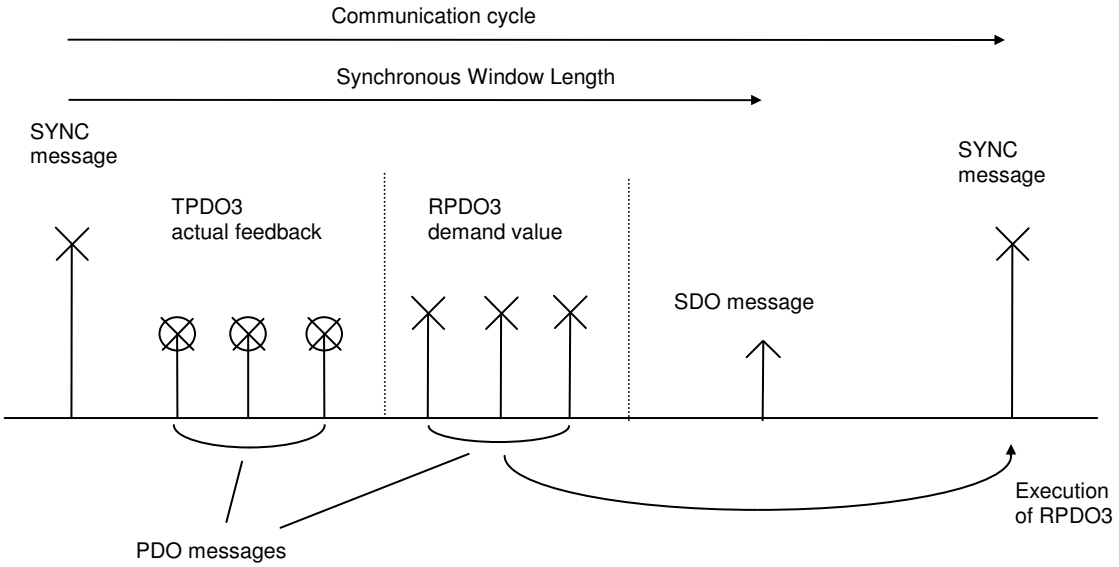
After launching VDSetup, the user must "disable CAN bus" in the "Set-up" menu. Otherwise, the drive cannot be enabled for auto-tuning or auto-phasing. Do not forget to "enable CAN bus" again before running with CAN bus.

After the commissioning, do not forget to store the parameters into the EEPROM.

Note: When executing the auto-tuning, the drive must be set into position mode in order to have the position gains calculated.

A2.3 - OPERATION

TRANSTECHNIK Servo Drives mainly runs in "Interpolated position mode" (see *section 3.8*). In this mode, the controller sends elementary position references to the drive at a regular cycle time.



A2.3.1 - START COMMUNICATION (NMT)

TRANSTECHNIK Servo Drives are able of a minimum boot-up. This means that after a reset at power-on, they will automatically switch to "Pre-Operational" state. In this state, it is possible to change parameters via the SDO communication.
To be able to exchange data via PDO, it is necessary to switch to "Operational" state by a "Start Remote Node" (see *section 2.1.3: NMT Protocols*).

A2.3.2 - SETTING COMMUNICATION PARAMETERS (SDO)

Except for the drive parameters which can be adjusted and stored in the drive, the communication parameters must be sent to the drive at every power on.

The main parameters to be set are:

- Communication cycle period (object 1006 sub-index 0): this object defines the duty cycle of the controller. This value is entered in µs and must be a multiple of 500. For example, 4000 for a cycle period of 4 ms.
- Synchronous window length (object 1007 sub-index 0): this object defines the synchronous window (µs) in which the drive can take the demand value via RPDO3. Generally, it can be set at a communication cycle value minus 1000.

Other communication parameters can be adjusted as COB-ID, PDO mapping...

For the interpolated position mode, the position resolution parameter (number of increments per motor revolution) also needs to be adjusted (object 608F sub-index 1).

A2.3.3 - START SYNCHRONISATION (SYNC)

After having started the communication by NMT, the controller must send a SYNC message and a RPDO3 message to the drive at every cycle defined in *section 3.2*. The drive is in "Operational" state.

A2.3.4 - EXCHANGE DATA (PDOS)

The exchange data in Interpolated position mode are mainly the position demand value sent by the controller (via RPDO3) and the current position value sent by the drive (via TPDO3).
It is possible to modify the mapping of these PDOs to exchange other data, but in interpolated position mode, the drive position demand is required for executing this function (object 60C1 sub-index 1: Interpolated data record).

A2.3.5. SERVO ON / SERVO OFF

Once the communication OK, the drive can be controlled via a control word (object 6040 sub-index 0) and a status word (object 6041 sub-index 0). The control and status words are written/read via SDO or RPDO1/TPDO1.

The state diagram is given in *section 3.3.1*.

Simplified sequence to enable the drive (servo on):

Control word sent by the controller	Status word replied by the drive	Drive state	Remarks
xx00h	x240h	Switch On Disabled	drive is disabled
xx06h	x221h	Ready To Switch On	drive is disabled
xx0Fh	x227h	Operation Enable	drive is enabled

The "Ready To Switch On" state is mandatory, as the drive cannot change from "Switch On Disabled" directly to "Operation Enable".
If there is an error, the drive will switch to "Fault Reaction Active" and then to "Fault" state. To exit the "Fault" state, the controller must send a rising edge of bit 7 in the control word. Once the fault reset, the drive will switch to "Switch On Disabled" state.

A2.3.6 - MOVE ON

Once in "operation enable", the controller must set bit 4 of the control word to enable the IP-Mode (see *section 3.3.3.1: Control_word*). The drive replies by setting bit 12 of the status word (see 3.3.3.2. Status_word). From this moment on, the drive will follow the position demand value via RPDO3.
Note: If the position demand is not set at the same value as the current position value when bit 4 of control word is set, a following error will occur.

A2.3.7 - OTHER CONSIDERATIONS

When an error occurs, an emergency message is sent to the controller (*see section 2.1.7*).
The "Fault" state can also be detected in the status word (xxxFh). The error meaning is in object 3020h (manufacturer specific).