UMC firmware

uniswarm
electronics for robots

FIRMWARE MANUAL

UMC motion control firmware

Description

The UniSwarm motion controllers (UMC) are specifically designed for use in robotics applications. These controllers provide precise and accurate control of motors to drive robotic movements. They are equipped with advanced motion control algorithms and a wide range of communication protocols, making them highly suitable for use in a variety of robotic systems. The functionality of the controllers may vary depending on the specific hardware board reference and revision, so it's important to consult the relevant hardware datasheet for additional information.

Features

- Advanced motion control algorithms for precise and accurate control of motors in robotics applications
- Support for CANOpen communication protocol for easy integration into robotics systems
- Configurable inputs and outputs for flexible control and monitoring of robotic movements
- Support for various types of motors, including brushless DC, brush DC, and stepper motors, commonly used in robotics

Interfaces

- CAN Fd bus compatible with CANOpen
- RS485 / RS422 interface for protocols like Modbus, Profibus or DMX512...

Motor

- DC brushed, brush-less and stepper
- Torque, speed and position limits set by software

Sensors

- Support for various types of sensors, including encoders, resolvers, and Hall effect sensors, commonly used in robotics
- Flexible input configuration options for precise monitoring of robotic movements
- Position, velocity, torque, limit switches

Mode

- TQ: Torque
- VL: Velocity
- PP: Profile Position
- IP: Interpolated Position

Please refer to the specific hardware datasheet for more detailed information on the features and capabilities.

Contents

List of Figures

List of Tables

Chapter 1

Configuration and status

1 Configuration process

2 Board

This section describes the specific status objects of the UMC.

2.1 Board status

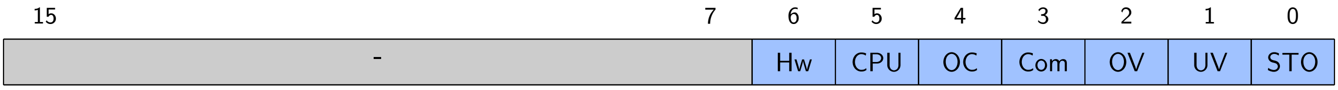
2.1.1 0x2800 Board_status

Index	Name	Object type
0x2800	Board_status	RECORD

Subindex	Name	Data type
1	Error	UINT16

0x2800.1 ***Error***

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-



15–7	6	5	4	3	2	1	0
-	Hw	CPU	OC	Com	OV	UV	STO

Figure 1.1: Board error bit-field

- b6 **Hw**: Error from board hardware
- b5 **CPU**: Error from CPU
- b4 **OC**: Board supply over-current detected
- b3 **Com**: Error from communication
- b2 **OV**: Over-voltage detected
- b1 **UV**: Under-voltage detected
- b0 **STO**: Error from STO input

2.1.2 0x2000 Board_voltage

Index	Name		Object type
0x2000	Board_voltage		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT16	RO,TPDO	-	0.01 V

Subindex	Name
1	Board_voltage_input

Input power board voltage in 0.01 V. This value is used for under / over-voltage detection. Useful to monitor battery voltage.

2.1.3 0x2001 Manufacture_date

Data type	Acces	Default	Range	Unit
USTRING	RO	YYYY/MM/DD hh:mm:ss	-	-

Date of board manufacturing.

2.1.4 0x2002 Calibration_date

Data type	Acces	Default	Range	Unit
VSTRING	RO	YYYY/MM/DD hh:mm:ss	-	-

Date of board calibration (if needed).

2.1.5 0x2003 Firmware_build_date

Data type	Acces	Default	Range	Unit
VSTRING	RO	-	-	-

Date and time of firmware build. In case of pre-release version, thanks to indicate this value to support team.

2.1.6 0x2004 Board_led

Data type	Acces	Default	Range	Unit
UINT8	RW	1	-	-

On board LEDs settings.

Value	Definition
0	Disables all status LED
1	Default LED display status

Table 1.1: Board LED options

2.1.7 0x2041 Board_voltage_config

Index	Name	Object type
0x2041	Board_voltage_config	RECORD

Subindex	Name	Data type
1	Undervoltage	UINT16
2	Overvoltage	UINT16

Over / under-voltage detection configuration. If the input board voltage fall below this under-voltage value, an under-voltage fault is generated. The same thing happens for over-voltage with dedicated parameter.

2.2 CPU status

2.2.1 0x2020 CPU_temperature

Index	Name		Object type
0x2020	CPU_temperature		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT16	RO,TPDO	-	0.1 °C

Subindex	Name
1	CPU1_temperature
2	CPU2_temperature

CPU temperature. Some boards can have multiple CPU with independent temperature sensor.

Note: The number of elements in the array may vary depending on the controller model.

2.2.2 0x2021 CPU_life_cycle

Index	Name		Object type
0x2021	CPU_life_cycle		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT16	RO,TPDO	-	-

Subindex	Name
1	CPU1_life_cycle
2	CPU2_life_cycle

A life time counter, incremented each CPU cycle. Can be used to monitor CPU speed or if a secondary CPU is frozen.

Note: The number of elements in the array may vary depending on the controller model.

2.2.3 0x2022 CPU_error

Index	Name		Object type
0x2022	CPU_error		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT16	RO	-	-

Subindex	Name
1	CPU1_error
2	CPU2_error

Future usage.

Note: The number of elements in the array may vary depending on the controller model.

2.2.4 0x2023 CPU_min_cycle_us

Index	Name		Object type
0x2023	CPU_min_cycle_us		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT16	RO	-	μs

Subindex	Name
1	CPU1_min_cycle_us
2	CPU2_min_cycle_us

Future statistics usage.

Note: The number of elements in the array may vary depending on the controller model.

2.2.5 0x2024 CPU_max_cycle_us

Index	Name		Object type
0x2024	CPU_max_cycle_us		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT16	RO	-	μs

Subindex	Name
1	CPU1_max_cycle_us
2	CPU2_max_cycle_us

Future statistics usage.

Note: The number of elements in the array may vary depending on the controller model.

2.2.6 0x2025 CPU_mean_cycle_us

Index	Name		Object type
0x2025	CPU_mean_cycle_us		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT16	RO	-	μs

Subindex	Name
1	CPU1_mean_cycle_us
2	CPU2_mean_cycle_us

Future statistics usage.

Note: The number of elements in the array may vary depending on the controller model.

2.3 Communication configuration

2.3.1 0x2040 Communication_config

Index	Name	Object type
0x2040	Communication_config	RECORD

Subindex	Name	Data type
1	Node_ID	UINT8
2	Bit_rate	UINT8

0x2040.1 *Node_ID*

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

CANOpen node id. To change the node id, set the new value in this object, store manufacturer parameters or all parameters and restart the node.

0x2040.2 *Bit_rate*

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

Value	Definition
0	Default speed, 1 Mbit/s
1	10 kbit/s
2	20 kbit/s
3	50 kbit/s
4	100 kbit/s
5	125 kbit/s
6	250 kbit/s
7	500 kbit/s
9	1 Mbit/s

Table 1.2: CAN bus bit rate options

To change the can speed, set the new value in this object, store manufacturer parameters or all parameters and restart the node.

800 kbit/s is not yet supported.

2.4 Power bridges status

Name	Description
0x2802 Currents_HL	
0x2803 Currents_LL	
0x2804 PWM	
0x2805 Back_EMF	

Table 1.3: Board bridges status objects

2.4.1 0x2802 Currents_HL

Index	Name		Object type
0x2802	Currents_HL		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT16	RO,TPDO	-	0.01 A

Subindex	Name
1	Current_HL_A
2	Current_HL_B
3	Current_HL_C
4	Current_HL_D

Value of current in bridge during the High to Low mosfet phase.

Warning: This value can evolve very quickly with working motors, only used for debug applications or low speed motor.

Note: The number of elements in the array may vary depending on the controller model.

2.4.2 0x2803 Currents_LL

Index	Name		Object type
0x2803	Currents_LL		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT16	RO,TPDO	-	0.01 A

Subindex	Name
1	Current_LL_A
2	Current_LL_B
3	Current_LL_C
4	Current_LL_D

Value of current in bridge during the Low to Low mosfet phase, circulating current.

Warning: This value can evolve very quickly with working motors, only used for debug applications or low speed motor.

Note: The number of elements in the array may vary depending on the controller model.

2.4.3 0x2804 PWM

Index	Name		Object type
0x2804	PWM		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT16	RO,TPDO	-	-

Subindex	Name
1	PWM_A
2	PWM_B
3	PWM_C
4	PWM_D

Duty cycle for each board bridge from 0 to 32768 (0% to 100%) or -1 for floating bridge.

Warning: This value can evolve very quickly with working motors, only used for debug applications or low speed motor.

Note: The number of elements in the array may vary depending on the controller model.

2.4.4 0x2805 Back_EMF

Index	Name		Object type
0x2805	Back_EMF		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT16	RO,TPDO	-	0.01 V

Subindex	Name
1	Back_EMF_A
2	Back_EMF_B
3	Back_EMF_C
4	Back_EMF_D

Back-EMF (Electro Motrice Force) voltage in tenth of volts for each bridge.

Warning: This value can evolve very quickly with working motors, only used for debug applications or low speed motor.

Note: The number of elements in the array may vary depending on the controller model.

2.5 Temperatures

Name	Description
0x2801 Driver_temperature	
0x2810 Driver_temperature_config	

Table 1.4: Temperatures objects

2.5.1 0x2801 Driver_temperature

Index	Name		Object type
0x2801	Driver_temperature		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT16	RO,TPDO	-	0.1 °C

Subindex	Name
1	Temperature_A
2	Temperature_B
3	Temperature_C
4	Temperature_D

Array of drivers bridge temperature in tenth of degree Celsius.

Note: The number of elements in the array may vary depending on the controller model.

2.5.2 0x2810 Driver_temperature_config

Index	Name	Object type
0x2810	Driver_temperature_config	RECORD

Subindex	Name	Data type
1	Protection_schmitt_triggers_low	INT16
2	Protection_schmitt_triggers_high	INT16

Schmitt trigger fault detection values for all driver temperature in tenth of degree Celsius. Default values may vary depending on the board.

To pass in over-temp state, bridge temperature needs to exceed the trigger high. A temperature that fall below trigger low will clear this fault.

`0x2810.1` ***Protection_schmitt_triggers_low***

Data type	Acces	Default	Range	Unit
INT16	RW	750	-	0.1 °C

`0x2810.2` ***Protection_schmitt_triggers_high***

Data type	Acces	Default	Range	Unit
INT16	RW	800	-	0.1 °C

3 Axis

This chapter describes the specific status objects of the UMC with the index range `0x4000` to `0x4FFF`.

3.1 Multi-axis

In the index range of the manufacturer-specific profile area (`0x4000` to `0x4FFF`), the device can manage up to 8 axis. Please check the hardware datasheet for the exact amount of axis supported. In multi-axis devices, objects of different axis can be accessed with the following addresses:

Axis	Sub-function	Objects index range
Axis 1	Motion	0x4000 to 0x4005
	Motor	0x4006 to 0x401F
	Torque	0x4020 to 0x403F
	Velocity	0x4040 to 0x405F
	Position	0x4060 to 0x407F
	Brake	0x4080 to 0x408F
Axis 2	...	0x4200 to 0x43FF
Axis 3	...	0x4400 to 0x45FF
Axis 4	...	0x4600 to 0x47FF
Axis 5	...	0x4800 to 0x49FF
Axis 6	...	0x4A00 to 0x4BFF
Axis 7	...	0x4C00 to 0x4DFF
Axis 8	...	0x4E00 to 0x4FFF

Table 1.5: Index range of specific board object

Note: All the objects described below correspond to axis 1

3.2 Motor

This area indicates different parameter statuses.

3.2.1 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

3.2.2 Input objects

Name	Description
a1_Motor_config	Generic motor configuration
a1_BLDC_config	BLDC specific configuration
a1_Brake_config	Brake configuration
a1_Limit_switches_config	Limit switches configuration

Table 1.6: Motor configuration objects

3.2.3 Output objects

Name	Description
a1_Motion_status	Motion status
a1_Motor_status	Generic motor status
a1_BLDC_status	BLDC specific status

Table 1.7: Motor status objects

3.2.4 0x4000 a1_Motion_status

Index	Name	Object type
0x4000	a1_Motion_status	RECORD
Subindex	**Name**	**Data type**
1	Error	UINT16

0x4000.1 ***Error***

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

Error bit-field that give the source of error with the following bits :

15 9	8	7	6	5	4	3	2	1	0
-	Sync	PosC	PosS	VIC	VIS	TqC	TqS	Mot	Brd

Figure 1.2: Motion error bit-field

- b8 **Sync**: Error from sync controller
- b7 **PosC**: Error from position controller
- b6 **PosS**: Error from position sensor
- b5 **VIC**: Error from velocity controller
- b4 **VIS**: Error from velocity sensor
- b3 **TqC**: Error from torque controller
- b2 **TqS**: Error from torque sensor
- b1 **Mot**: Error from motor
- b0 **Brd**: Error from board

3.2.5 0x4006 a1_Motor_status

Index	Name	Object type
0x4006	a1_Motor_status	RECORD

Subindex	Name	Data type
1	Command	INT16
2	Current	INT16
3	Torque	Q15.16
4	Velocity	Q15.16
5	Position	Q15.16
6	Error	UINT16
7	Temperature	INT16

0x4006.1 *Command*

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

Duty cycle sent to the power stage from -32768 to 32768 (-100% to 100%).

0x4006.2 *Current*

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	0.01 A

Average phase current used to compute the torque.

0x4006.3 *Torque*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Torque in N.m from motor current multiply by the current constant.

0x4006.4 *Velocity*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Motor velocity from software motor control.

0x4006.5 *Position*

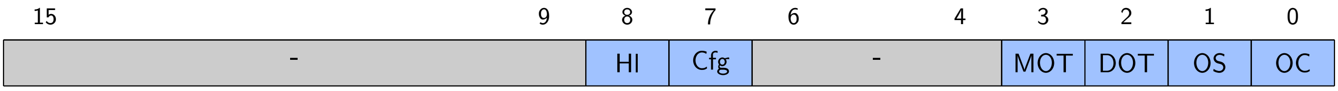
Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Motor position from software motor control.

0x4006.6 *Error*

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

Motor error bit-field, 0 if no error is present.



15 9	8	7	6 4	3	2	1	0
-	HI	Cfg	-	MOT	DOT	OS	OC

Figure 1.3: Motor error bit-field

- b8 **HI**: Hall sensor forbidden value (for hall based motor only)
- b7 **Cfg**: Configuration error

- b3 **MOT**: Motor over-temperature
- b2 **DOT**: Driver over-temperature
- b1 **OS**: Over-speed
- b0 **OC**: Over-current

0x4006.7 *Temperature*

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	0.1 °C

Motor temperature if a sensor is configured.

3.2.6 0x4008 a1_BLDC_status

Index	Name	Object type
0x4008	a1_BLDC_status	RECORD

Subindex	Name	Data type
1	Hall_raw_value	INT8
2	Hall_phase	INT8
3	Electrical_angle	INT16

0x4008.1 *Hall_raw_value*

Data type	Acces	Default	Range	Unit
INT8	RO,TPDO	-	-	-

Direct value of 3 bits from hall sensor connector.

0x4008.2 *Hall_phase*

Data type	Acces	Default	Range	Unit
INT8	RO,TPDO	-	-	-

Decoded phase value from hall sensor connector.

0x4008.3 *Electrical_angle*

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	°

Current powered electrical angle in degrees from BLDC software control.

3.2.7 0x4007 a1_Motor_config

Index	Name	Object type
0x4007	a1_Motor_config	RECORD

Subindex	Name	Data type
1	Type	UINT16
2	Peak_current	UINT16
3	Burst_current	UINT16
4	Burst_duration	UINT16
5	Sustained_current	UINT16
6	Current_constant	Q15.16
7	Max_velocity	UINT16
8	Velocity_constant	Q15.16
9	Flags	UINT16

0x4007.1 *Type*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	-

Type of motor to use on this axis and software method.

Value	Motor type
0x0000	No motor
0x0101	DC motor
0x0201	BLDC trapezoidal with hall
0x0202	BLDC sinusoidal with hall

Table 1.8: Motor types

0x4007.2 *Peak_current*

Data type	Acces	Default	Range	Unit
UINT16	RW	3200	-	0.01 A

Maximum peak current in 0.01A. This value is set to hardware current comparator. When this value is exceeded, every power bridge is open to avoid hardware damage.

0x4007.3 *Burst_current*

Data type	Acces	Default	Range	Unit
UINT16	RW	2000	-	0.01 A

Software current limit value in 0.01A. If this value is exceeded during more than the duration defined in Burst_duration.

0x4007.4 *Burst_duration*

Data type	Acces	Default	Range	Unit
UINT16	RW	1000	-	ms

Software current limit duration in ms. See Burst_current description.

0x4007.5 *Sustained_current*

Data type	Acces	Default	Range	Unit
UINT16	RW	1500	-	0.01 A

Continuous phase current in 0.01A.

0x4007.6 *Current_constant*

Data type	Acces	Default	Range	Unit
Q15.16	RW	65536	-	Nm/A

Current constant to compute torque from on board current measures. Given in Nm / A.

0x4007.7 *Max_velocity*

Data type	Acces	Default	Range	Unit
UINT16	RW	25000	-	rpm

Maximal motor velocity. If this velocity is exceeded, a motor fault is generated.

0x4007.8 *Velocity_constant*

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	rpm/V

Velocity constant used to compute velocity from on board back EMF measures for specific motors.

0x4007.9 ***Flags***

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	-

Various motor configuration flags.

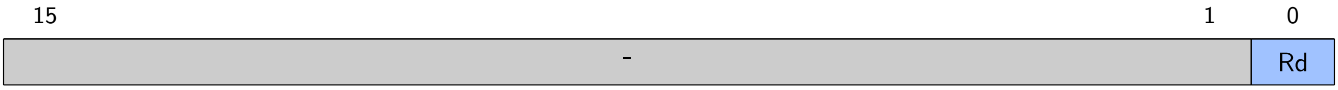


Figure 1.4: Motor config bit-field

- b0 **Rd**: Reverse motor direction

3.2.8 0x4009 a1_BLDC_config

Index	Name	Object type
0x4009	a1_BLDC_config	RECORD

Subindex	Name	Data type
1	Pole_pair	UINT8

0x4009.1 ***Pole_pair***

Data type	Acces	Default	Range	Unit
UINT8	RW	1	-	-

Pole pair count for BLDC motors only.

3.3 Brake

3.3.1 0x4082 a1_Brake_config

Index	Name	Object type
0x4082	a1_Brake_config	RECORD

Subindex	Name	Data type
1	Brake_mode	UINT8
2	Brake_excitation_time_ms	UINT16
3	Brake_excitation_duty	UINT16
4	Brake_activated_duty	UINT16
5	Brake_release_delay	UINT16
6	Brake_release_to_OE	UINT16
7	Brake_closing_delay	UINT16
8	Brake_closing_to_idle	UINT16

0x4082.1 ***Brake_mode***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

Value	Definition
0	Brake disabled
2	Open drain brake, normally closed
3	Open drain brake, normally open
4	Open source brake, normally closed
5	Open source brake, normally open
6	Push-pull brake, normally closed
7	Push-pull brake, normally open

Table 1.9: Brake modes enum

Warning: Supported modes may vary depending on the board model.

0x4082.2 *Brake_excitation_time_ms*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	ms

Brake excitation duration. During this period, the used duty cycle is the brake excitation one. To disable excitation, set time to 0.

0x4082.3 *Brake_excitation_duty*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	0.00152588 %

Brake duty cycle during excitation period from 0 to 65535 (0% to 100%).

0x4082.4 *Brake_activated_duty*

Data type	Acces	Default	Range	Unit
UINT16	RW	32768	-	0.00152588 %

Continuous brake duty cycle after excitation period from 0 to 65535 (0% to 100%).

0x4082.5 *Brake_release_delay*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	ms

When activating Operation Enabled, delay in ms of control loop stabilization before releasing brake.

0x4082.6 *Brake_release_to_OE*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	ms

With control loop activated with neutral target, delay of brake release before starting Operation Enabled.

0x4082.7 *Brake_closing_delay*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	ms

When dis-activating Operation Enabled, delay of control loop stabilization before closing brake.

0x4082.8 *Brake_closing_to_idle*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	ms

With control loop still activated with neutral target, delay of brake close before stopping control loop and power.

3.3.2 0x4083 a1_Limit_switches_config

Index	Name	Object type
0x4083	a1_Limit_switches_config	RECORD

Subindex	Name	Data type
1	Limit_low_input_select	UINT8
2	Limit_low_config_flags	UINT8
3	Limit_high_input_select	UINT8
4	Limit_high_config_flags	UINT8
5	Home_input_select	UINT8
6	Home_config_flags	UINT8

`0x4083.1` ***Limit_low_input_select***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

Digital input selection for low limit input. For example, set 2 to select DI2 as low limit input.

`0x4083.2` ***Limit_low_config_flags***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

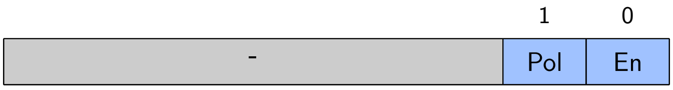


Figure 1.5: Limit_low_input_select bit-field

- `b1` **Pol**: Polarity selection
- `b0` **En**: Limit enabled

Flags to enable limit and choose the polarity of the input. If polarity is set to 0, the limit will be active when the selected digital input is high. Sets polarity to 1 to invert polarity.

If the low limit input is active with enabled set to 1, the axis refuse to move with a negative target.

`0x4083.3` ***Limit_high_input_select***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

Digital input selection for high limit input. For example, set 3 to select DI3 as high limit input.

`0x4083.4` ***Limit_high_config_flags***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

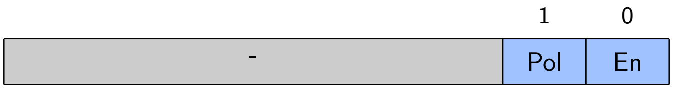


Figure 1.6: Limit_high_config_flags bit-field

- `b1` **Pol**: Polarity selection
- `b0` **En**: Limit enabled

Flags to enable limit and choose the polarity of the input. If polarity is set to 0, the limit will be active when the selected digital input is high. Sets polarity to 1 to invert polarity.

If the high limit input is active with enabled set to 1, the axis refuse to move with a positive target.

`0x4083.5` ***Home_input_select***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

Digital input selection for home input. For example, set 4 to select DI4 as home input.

`0x4083.6` ***Home_config_flags***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

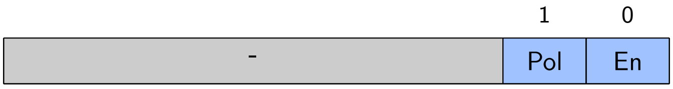


Figure 1.7: Home_config_flags bit-field

- `b1` **Pol**: Polarity selection
- `b0` **En**: Limit enabled

3.4 Sensors

3.4.1 `0x4022` *a1_Torque_sensor_status* `0x4042` *a1_Velocity_sensor_status* `0x4062` *a1_Position_sensor_status*

Index	Name	Object type
0x4022	a1_Torque_sensor_status	RECORD

Subindex	Name	Data type
1	Raw_data	Q15.16
2	Flags	INT16
3	Value	Q15.16

`0x4022.1` ***Raw_data***

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Current raw data value from the sensor.

`0x4022.2` ***Flags***

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

Current sensor status flag, 0 mean valid value.

Value	Sensor type
0	No error, valid value
-1	Min error, value of sensor is inferior to minimum value
-2	Max error, value of sensor exceed maximal value
-256	Digital acquisition error

Table 1.10: Sensor types

`0x4022.3` ***Value***

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Current value output after filtering and conditioning.

3.4.2 0x4023 *a1_Torque_sensor_config* 0x4043 *a1_Velocity_sensor_config* 0x4063 *a1_Position_sensor_config*

Index	Name	Object type
0x4023	a1_Torque_sensor_config	RECORD

Subindex	Name	Data type
1	Sensor_select	UINT16
2	Frequency_divider	UINT16
3	Sensor_flags	UINT16
4	Param_0	INT16
5	Param_1	INT16
6	Param_2	INT16
7	Param_3	INT16

0x4023.1 *Sensor_select*

Data type	Acces	Default	Range	Unit
UINT16	RW	4352	-	-

Selects the sensor.

Value	Sensor type
0x0000	OFF
0x1100	TORQUE_FROM_MOTOR
0x1200	VELOCITY_FROM_MOTOR
0x1300	POSITION_FROM_MOTOR
0x2200	POSITION_FROM_VELOCITY
0x2300	VELOCITY_FROM_POSITION
0x3101	QEI_CH1
0x3102	QEI_CH2
0x3201	SSI_CH1
0x3202	SSI_CH2
0x4001	ANALOG_CH1
0x4002	ANALOG_CH2

Table 1.11: Sensor types

0x4023.2 *Frequency_divider*

Data type	Acces	Default	Range	Unit
UINT16	RW	1	-	-

Configure the frequency divider of sensor.

0x4023.3 *Sensor_flags*

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	-

Configuration flags.

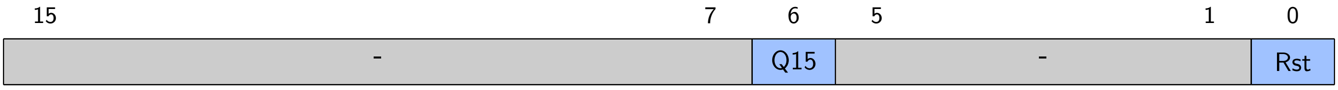


Figure 1.8: Sensor configuration flags

- b6 **Q15**: Q15.16 enable, shift sensor raw value by 16 bits right
- b0 **Rst**: Reset counter (for incremental sensors)

0x4023.4 *Param_0*

Data type	Acces	Default	Range	Unit
INT16	RW	0	-	-

Configure parameter 0, depends on the sensor.

0x4023.5 *Param_1*

Data type	Acces	Default	Range	Unit
INT16	RW	0	-	-

Configure parameter 1, depends on the sensor.

0x4023.6 *Param_2*

Data type	Acces	Default	Range	Unit
INT16	RW	0	-	-

Configure parameter 2, depends on the sensor.

0x4023.7 *Param_3*

Data type	Acces	Default	Range	Unit
INT16	RW	0	-	-

Configure parameter 3, depends on the sensor.

3.4.3 0x4024 *a1_Torque_sensor_filter* 0x4044 *a1_Velocity_sensor_filter* 0x4064 *a1_Position_sensor_filter*

Index	Name	Object type
0x4024	a1_Torque_sensor_filter	RECORD

Subindex	Name	Data type
1	Filter_select	UINT16
2	Param_0	Q15.16
3	Param_1	Q15.16
4	Param_2	Q15.16
5	Param_3	Q15.16

0x4024.1 *Filter_select*

Data type	Acces	Default	Range	Unit
UINT16	RW	8192	-	-

Filter selection.

Value	Threshold type	Description
0x0000	No filter	No filter active, direct raw value
0x2000	Averaging	Averaging value from 'N' samples

Table 1.12: Threshold types

0x4024.2 *Param_0*

Data type	Acces	Default	Range	Unit
Q15.16	RW	1048576	-	-

Configure parameter 0, depends on the filter.

0x4024.3 ***Param_1***

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure parameter 1, depends on the filter.

0x4024.4 ***Param_2***

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure parameter 2, depends on the filter.

0x4024.5 ***Param_3***

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure parameter 3, depends on the filter.

3.4.4 0x4025 *a1_Torque_sensor_conditioning* 0x4045 *a1_Velocity_sensor_conditioning* 0x4065 *a1_Position_sensor_conditioning*

Index	Name	Object type
0x4025	a1_Torque_sensor_conditioning	RECORD
Subindex	**Name**	**Data type**
1	Pre_offset	Q15.16
2	Scale	Q15.16
3	Post_offset	Q15.16
4	Error_min	Q15.16
5	Error_max	Q15.16
6	Threshold_min	Q15.16
7	Threshold_max	Q15.16
8	Threshold_mode	UINT8

0x4025.1 ***Pre_offset***

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure the pre-offset of sensor.

0x4025.2 ***Scale***

Data type	Acces	Default	Range	Unit
Q15.16	RW	65536	-	-

Configure the scale of sensor.

0x4025.3 ***Post_offset***

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure the post-offset of sensor.

0x4025.4 ***Error_min***

Data type	Acces	Default	Range	Unit
Q15.16	RW	-2147483648	-	-

Configure the minimal value of sensor that trig an error.

0x4025.5 *Error_max*

Data type	Acces	Default	Range	Unit
Q15.16	RW	2147483647	-	-

Configure the maximal value of sensor that trig an error.

0x4025.6 *Threshold_min*

Data type	Acces	Default	Range	Unit
Q15.16	RW	-2147483648	-	-

Configure the threshold min of sensor.

0x4025.7 *Threshold_max*

Data type	Acces	Default	Range	Unit
Q15.16	RW	2147483647	-	-

Configure the threshold max of sensor.

0x4025.8 *Threshold_mode*

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

Configure the threshold mode of sensor.

Value	Threshold mode	Description
0x0000	No threshold	No threshold active, direct value
0x0001	Min-max	Limit output range from min to max
0x0002	Modulo	Apply a modulo to keep value in range from min to max
0x0003	Last valid value	Keep the last valid value as output

Table 1.13: Threshold modes

3.5 Control loops

3.5.1 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

Objects in this area configure the control loop (PID) of Torque, Velocity and Position and also configure sensor. Some object indicates different status.

This index range of control loop is split as follows:

Index range	Description	Object index	Description
0x4020 to 0x403F	Torque control loop	0x4020	a1_Torque_PID_status
		0x4021	a1_Torque_PID_config
		0x4022	a1_Torque_sensor_status
		0x4023	a1_Torque_sensor_config
		0x4024	a1_Torque_sensor_filter
		0x4025	a1_Torque_sensor_conditioning
0x4040 to 0x405F	Velocity control loop	0x4040	a1_Velocity_PID_status
		0x4041	a1_Velocity_PID_config
		0x4042	a1_Velocity_sensor_status
		0x4043	a1_Velocity_sensor_config
		0x4044	a1_Velocity_sensor_filter
		0x4045	a1_Velocity_sensor_conditioning
0x4060 to 0x407F	Position control loop	0x4060	a1_Position_PID_status
		0x4061	a1_Position_PID_config
		0x4062	a1_Position_sensor_status
		0x4063	a1_Position_sensor_config
		0x4064	a1_Position_sensor_filter
		0x4065	a1_Position_sensor_conditioning

Table 1.14: Index range of control loop

3.5.2 0x4020 *a1_Torque_PID_status*
0x4040 *a1_Velocity_PID_status*
0x4060 *a1_Position_PID_status*

Index	Name	Object type
0x4020	a1_Torque_PID_status	RECORD

Subindex	Name	Data type
1	Input	Q15.16
2	Error	Q15.16
3	Integrator	Q15.16
4	Output	Q15.16
5	Target	Q15.16

0x4020.1 *Input*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Indicates input of control loop PID.

0x4020.2 *Error*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Indicates error of control loop PID.

0x4020.3 *Integrator*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Indicates integrator of control loop PID.

0x4020.4 *Output*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Indicates output of control loop PID.

3.5.3 0x4021 *a1_Torque_PID_config*
0x4041 *a1_Velocity_PID_config*
0x4061 *a1_Position_PID_config*

Index	Name	Object type
0x4021	a1_Torque_PID_config	RECORD

Subindex	Name	Data type
1	P	Q15.16
2	I	Q15.16
3	D	Q15.16
4	Min	Q15.16
5	Max	Q15.16
6	Threshold	Q15.16
7	Frequency_divider	UINT16
8	PID_flags	UINT16

0x4021.1 *P*

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure the proportional gain of control loop PID.

0x4021.2 *I*

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure the integral gain of control loop PID.

0x4021.3 *D*

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure the derivative gain of control loop PID.

0x4021.4 *Min*

Data type	Acces	Default	Range	Unit
Q15.16	RW	-2147483648	-	-

Configure the minimum output of control loop PID.

0x4021.5 *Max*

Data type	Acces	Default	Range	Unit
Q15.16	RW	2147483647	-	-

Configure the maximin output of control loop PID.

0x4021.6 *Threshold*

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configure the threshold of control loop PID.

0x4021.7 *Frequency_divider*

Data type	Acces	Default	Range	Unit
UINT16	RW	1	-	-

Configure the frequency divider of control loop PID.

0x4021.8 ***PID_flags***

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	-

Configuration flags for PID.

Value	Flags	Description
0x0001	ANTI_REVERSE	Prevent output with sign that differ with target

Table 1.15: PID flags

3.6 Axis synchronization mode

This mode allows to synchronized two axis. This synchronization is authorized on the UMC board with the control of 2 axis (axis 1 and 2).

The synchronization mode is configurable with object 0x2A01 S12_Synchro_config.

3.6.1 Mode activation

The activation of the axis synchronization mode is conditioned by several parameters:

- both axis are in same state : Operation Enabled
- both axis are in same operating mode (see 0x6060 a1_Modes_of_operation)
- the gap between two axis must be < 0x2A01.2 *Max_diff*

If theses conditions are not met and 0x2A01.1 *Mode_Synchro* != 0, the target sent is 0.

3.6.2 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

3.6.3 Input objects

This area indicates different parameter statuses.

Name	Description
0x2A01 S12_Synchro_config	

Table 1.16: Configuration objects

3.6.4 Output objects

Name	Description
0x2A00 S12_Synchro_status	

Table 1.17: Status objects

3.6.5 0x2A00 S12_Synchro_status

Index	Name	Object type
0x2A00	S12_Synchro_status	RECORD
Subindex	**Name**	**Data type**
1	Flag	INT16
2	Error	Q15.16
3	Corrector	Q15.16

Indicates synchro statuses.

0x2A00.1 *Flag*

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

Indicates the status of synchronization mode.

Flag	Description
0	SYNC OFF
1	SYNC INSIDE WINDOW
2	SYNC ACTIVE CORRECTOR
-1	SYNC ERROR MODE
-2	SYNC ERROR MAX

Table 1.18: Synchro status flags

0x2A00.2 *Error*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Indicates error between axis.

0x2A00.3 *Corrector*

Data type	Acces	Default	Range	Unit
Q15.16	RO,TPDO	-	-	-

Indicates applied corrector.

3.6.6 0x2A01 S12_Synchro_config

Index	Name	Object type
0x2A01	S12_Synchro_config	RECORD
Subindex	**Name**	**Data type**
1	Mode_Synchro	UINT8
2	Max_diff	Q15.16
3	Coeff	Q15.16
4	Window	Q15.16
5	Offset	Q15.16

Controls the synchronization of axis 2 on axis 1.

0x2A01.1 *Mode_Synchro*

Data type	Acces	Default	Range	Unit
UINT8	RW,RPDO	0	-	-

Defines the mode of synchronization.

Mode	Description
0x0000	Off
0x0001	On

Table 1.19: Modes of synchro

0x2A01.2 *Max_diff*

Data type	Acces	Default	Range	Unit
Q15.16	RW	5242880	-	-

Configures the max acceptable difference.

`0x2A01.3` ***Coeff***

Data type	Acces	Default	Range	Unit
Q15.16	RW	8388608	-	-

Configures the coefficient.

`0x2A01.4` ***Window***

Data type	Acces	Default	Range	Unit
Q15.16	RW	131072	-	-

Configures the synchronization acceptance window.

`0x2A01.5` ***Offset***

Data type	Acces	Default	Range	Unit
Q15.16	RW	0	-	-

Configures the offset.

Chapter 2

CANOpen motion controller, DS402

1 Drives and motion control

The CiA 402 or DS402 specifies the CANOpen interface for management of drive and motion controllers. This profile allows to control the axis via several operating modes:

- Manufacturer mode:
  - Duty Cycle (DTY)
- Torque mode:
  - Torque (TQ)
- Velocity mode:
  - Velocity (VL)
- Position modes:
  - Profile Position (PP)
  - Interpolated Position (IP)
  - Continuous Position (CP)

1.1 Multi-axis

In the index range of the standardized profile area (0x6000 to 0x9FFF), the device can manage up to 8 axis. In multi-axis devices, objects can be accessed by shifting as follows:

Axis	Index range
Axis 1	0x6000 - 0x67FF
Axis 2	0x6800 - 0x6FFF
Axis 3	0x7000 - 0x77FF
Axis 4	0x7800 - 0x7FFF
Axis 5	0x8000 - 0x87FF
Axis 6	0x8800 - 0x8FFF
Axis 7	0x9000 - 0x97FF
Axis 8	0x9800 - 0x9FFF

Table 2.1: Index range of standardized profile area

1.2 Power drive system (PDS)

The PDS FSA can be seen as a black box that control the application behavior when a control device is using the PDS. The PDS assure control even if the communication doesn't work properly. The PDS FSA depends on CANOpen status.

The controlword received via the network and local signals are used to manage the PDS. The statusword is produced by the drive device and used to send the state of the PDS. Error detection signals are used to drive the PDS FSA.

The status and the possible control sequence of the PDS are defined by the FSA of the PDS. Each special internal and external behavior is represented by a unique state. The commands accepted depend on the state of the PDS FSA.

1.2.1 State machine

The state machine of the power drive system controls the power electronics according to user parameters and events that occurred.

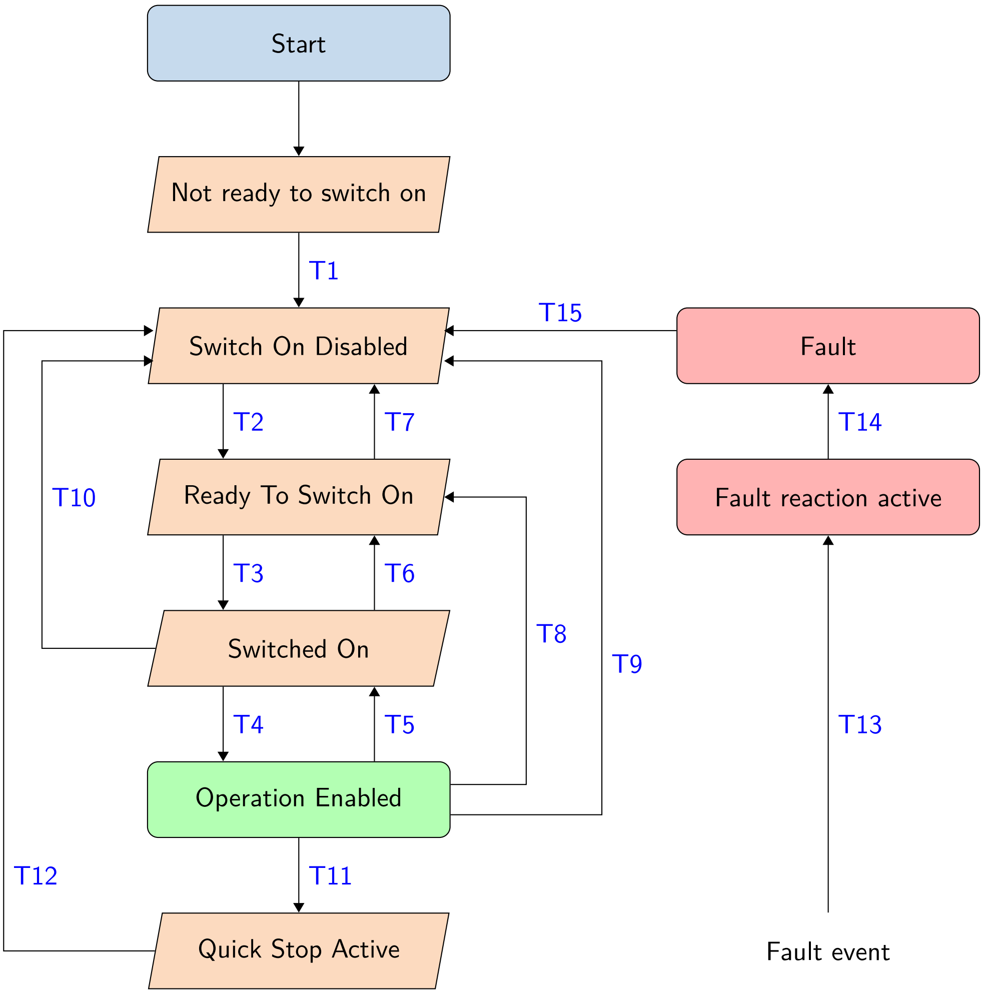


Figure 2.1: State machine of power drive system

When the system is started, it enters the start state and automatically switches to the Not ready to switch on state if NMT is in the Start state.

Transition	Event(s)	Action(s)	Command
T0	Automatic transition after power-on or reset application	Drive device self-test and/or self initialization shall be performed.	-
T1	Automatic transition	Communication shall be activated.	-
T2	Shutdown command from control device or local signal	None	Shutdown
T3	Switch on command received from control device or local signal	The high-level power shall be switched on, if possible.	Switch on
T4	Enable operation command received from control or local signal	The drive function shall be enabled and all internal set-points cleared.	Switch on + enable operation
T5	Disable operation command received from control device or local signal	The drive function shall be disabled.	Disable operation
T6	Shutdown command received from control device or local signal	The high-level power shall be switched off, if possible.	Enable operation
T7	Quick stop or disable voltage command from control device or local signal	None	Quick stop
T8	Shutdown command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.	Shutdown
T9	Disable voltage command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.	Disable voltage
T10	Disable voltage or quick stop command from control device or local signal	The high-level power shall be switched off, if possible.	Quick stop
T11	Quick stop command from control device or local signal	The quick stop function shall be started.	Quick stop
T12	Automatic transition when the quick stop function is completed and quick stop option code is 1, 2, 3 or 4, or disable voltage command received from control device (depends on the quick stop option code)	The drive function shall be disabled, and the high-level power shall be switched off, if possible.	Disable voltage
T13	Fault signal	The configured fault reaction function shall be executed.	-
T14	Automatic transition	The drive function shall be disabled; the high-level power shall be switched off, if possible.	-
T15	Fault reset command from control device or local signal	A reset of the fault condition is carried out, if no fault exists currently on the drive device; after leaving the fault state, the fault reset bit in the controlword shall be cleared by the control device.	Fault reset

Table 2.2: Transition options

Transition events and actions

Function	States							
	Not ready to switch on	Switch On Disabled	Ready To Switch On	Switched On	Operation Enabled	Quick Stop Active	Fault reaction active	Fault
Brake applied, if present	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes
Low-level power applied	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High-level power applied	Yes/No	Yes/No	Yes/No	Yes	Yes	Yes	Yes	Yes/No
Drive function Enabled	No	No	No	No	Yes	Yes	Yes	No
Configuration Allowed	Yes	Yes	Yes	Yes	Yes/No	Yes/No	Yes/No	Yes

Table 2.3: Supported functions

Supported functions When the motor is not connected to the high level power, it must rotate freely if it is not braking.

When the drive function is deactivated the motor must not receive power. All the control values such as torque, velocity or position should not be treated.

If the drive function is activated, energy can be provided to the motor and control values should be treated.

If there is a default detected by the drive device, it should reach the active fault state. When this state is reached a special fault reaction is executed.

After this reaction, the state of the drive device changes to the fault state.

When the default is no longer active, the fault state can be left after a reset command.

After a fatal error, it is necessary to switch-off the drive device instantly, as it can no longer control the motor.

Transition options The behavior of transitions can be modified with the following objects:

Transition	Object
Events	0x6007 a1_Abort_connection_option
T11	0x605A a1_Quick_stop_option
T8	0x605B a1_Shutdown_option
T5	0x605C a1_Disable_operation_option
T13	0x605E a1_Fault_reaction_option

Table 2.4: Transition options

1.3 Modes of operation

Several modes of operation are implemented and user selectable.

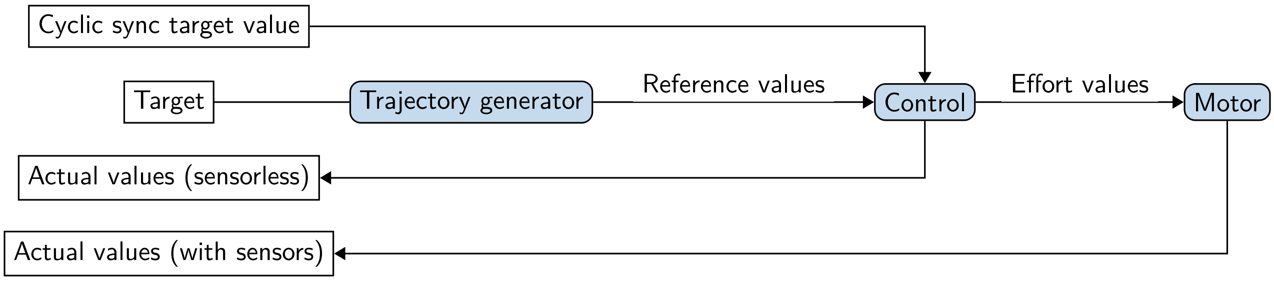


Figure 2.2: Relation between different value parameters

The 0x6060 a1_Modes_of_operation object enables mode change.

1.4 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

1.4.1 Input objects

Name	Description
0x6040 a1_Controlword	Control the power drive system state machine
0x6060 a1_Modes_of_operation	Modify the operation mode
0x6007 a1_Abort_connection_option	Modify action during an Abort connection
0x605A a1_Quick_stop_option	Modify action during a Quick stop
0x605B a1_Shutdown_option	Modify action during a Shutdown
0x605C a1_Disable_operation_option	Modify action during a Disable operation
0x605D a1_Halt_option	Modify action during a Halt
0x605E a1_Fault_reaction_option	Modify action during a Fault reaction

Table 2.5: Configuration objects

1.4.2 Output objects

Name	Description
0x6041 a1_Statusword	Status of the power drive system finite state
0x6061 a1_Modes_of_operation_display	Current operating mode

Table 2.6: Status objects

1.4.3 0x6040 a1_Controlword

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	0	-	-

This object allows to control the power drive system state machine. The start of any movement is specific to the operating mode.

15	10	9	8	7	6	4	3	2	1	0
-		OMS	H	FR	OMS		EO	QS	EV	SO

Figure 2.3: Controlword: Command coding

- b0 **SO**: Switch on
- b1 **EV**: Enable voltage
- b2 **QS**: Quick stop
- b3 **EO**: Enable operation
- b7 **FR**: Fault reset
- b8 **H**: Halt, behavior depending on the specific operating mode :
  Activate or deactivate the controlled movement, the behavior of Halt depends on the specific operating mode and is adjustable by 0x605D a1_Halt_option.
- b9, b6-4 **OMS**: Operation mode specific, mode depending

Command	Control Word					Transitions
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	T2, T6, T8
Switch on	0	0	1	1	1	T3
Switch on + enable operation	0	1	1	1	1	T3 + T4
Disable voltage	0	X	X	0	X	T7,9,T10,T12
Quick stop	0	X	0	1	X	T7,T10,T11
Disable operation	0	0	1	1	1	T5
Enable operation	0	1	1	1	1	T4,T16
Fault reset	1	X	X	X	X	T15

Figure 2.4: Frame of Controlword

1.4.4 `0x6041` a1_Statusword

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

This object provide the status of the power drive system finite state automaton. The current state of the state machine is provided by the bit combination of `0x6041` *a1_Statusword*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS		OMS		ILA	TR	RM	MS	W	SOD	QS	VE	F	OE	SO	RTSO

Table 2.7: Status word: status coding

- b0 **RTSO**: Ready to switch on
- b1 **SO**: Switched on
- b2 **OE**: Operation enabled
- b3 **F**: Fault
- b4 **VE**: Voltage enabled
- b5 **QS**: Quick stop
- b6 **SOD**: Switch on disabled
- b7 **W**: Warning
- b8 **MS**: Manufacturer-specific
- b9 **RM**: remote
- b10 **TR**: Target reached
- b11 **ILA**: Internal limit active, mode depending
- b12–13 **OMS**: Operation mode specific, mode depending
- b14–15 **MS**: Manufacturer-specific

Bit 6,5,3,2,1,0	State
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x0xx 1111	Fault reaction active
xxxx xxxx x0xx 1000	Fault

Table 2.8: Statusword: State coding

1.4.5 0x6060 a1_Modes_of_operation

Data type	Acces	Default	Range	Unit
INT8	RW,RPDO	-1	[-128 - 10]	-

This object allows to modify the operation mode. Please note that the mode change is not direct, you need to observe 0x6061 a1_Modes_of_operation_display object.

Value	Definition
-16	Continuous Position (CP)
-1	Duty Cycle (DTY)
0	No mode change/no mode assigned
1	Profile Position (PP)
2	Velocity (VL)
4	Torque (TQ)
7	Interpolated Position (IP)

Figure 2.5: Options of modes of operation

1.4.6 0x6061 a1_Modes_of_operation_display

Data type	Acces	Default	Range	Unit
INT8	RO,TPDO	-	[-128 - 10]	-

This object provides the actual operation mode. Values are the same as 0x6060 a1_Modes_of_operation in table 0x6060 a1_Modes_of_operation.

1.4.7 0x6502 a1_Supported_drive_modes

Data type	Acces	Default	Range	Unit
UINT32	RO	1115211	[0 - 0xFFFFFFFF]	-

This object provides the supported drive modes.

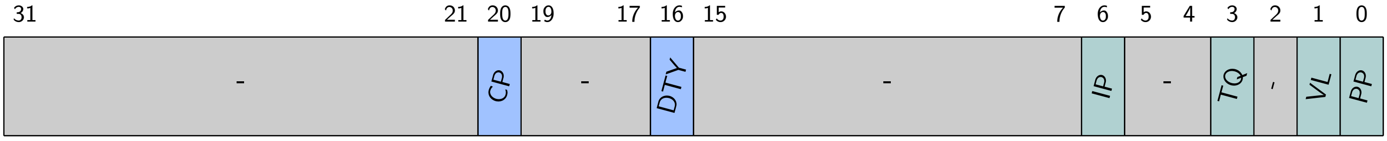


Figure 2.6: Frame of supported drive modes

- b0 **PP**: Profile Position (PP)
- b1 **VL**: Velocity (VL)
- b3 **TQ**: Torque (TQ)
- b6 **IP**: Interpolated Position (IP)
- b16 **DTY**: Duty Cycle (DTY)
- b20 **CP**: Continuous Position (CP)

Value	Description
0	Not supported
1	Supported

Table 2.9: Supported drive modes: coding

1.4.8 0x6007 a1_Abort_connection_option

Data type	Acces	Default	Range	Unit
INT16	RW	1	[0 - 3]	-

This object is used to configure the command to be executed when an event occurs:

- Bus-off: problem on the network
- Heartbeat, Node guarding
- Network management services (NMT) events: change of status (NMT Start; Stop, PreOp)
- Network management services (NMT) reset: reset application and reset communication

Value	Definition
0	No action
1	Fault signal
2	Disable voltage command
3	Quick stop command

Table 2.10: Abort connection options

Note: Each option can be configured with a corresponding object (0x605E a1_Fault_reaction_option, 0x605C a1_Disable_operation_option and 0x605A a1_Quick_stop_option

1.4.9 0x605A a1_Quick_stop_option

Data type	Acces	Default	Range	Unit
INT16	RW	2	[0 - 6]	-

This object is used to configure the action to be executed when a quick stop function occurs.

Value	Definition
0	Disable drive function
1	Slow down ramp (mode depending) and stay in Switch On Disabled
2	Quick stop ramp (mode depending) and stay in Switch On Disabled
5	Slow down ramp (mode depending) and stay in Quick Stop Active
6	Quick stop ramp (mode depending) and stay in Quick Stop Active

Table 2.11: Quick stop options

1.4.10 0x605B a1_Shutdown_option

Data type	Acces	Default	Range	Unit
INT16	RW	1	[0 - 1]	-

This object is used to configure the action to be executed when there is a transition from Operation Enabled state to Ready To Switch On state.

Value	Definition
0	Disable the drive function (switch-off the drive power stage)
1	Slow down ramp (mode depending) and deactivation of the drive function

Table 2.12: Shutdown options

1.4.11 0x605C a1_Disable_operation_option

Data type	Acces	Default	Range	Unit
INT16	RW	1	[0 - 1]	-

This object is used to configure the action to be executed when there is a transition from Operation Enabled state to Switched On state.

Value	Definition
0	Disable the drive function (switch-off the drive power stage)
1	Slow down ramp (mode depending) and deactivation of the drive function

Table 2.13: Disable operation options

1.4.12 0x605D a1_Halt_option

Data type	Acces	Default	Range	Unit
INT16	RW	1	[1 - 2]	-

This object is used to configure the action to be executed when a halt function occurs.

Value	Definition
1	Slow down ramp, mode depending
2	Quick stop ramp, mode depending

Table 2.14: Halt options

Note: Whatever the option, the state of the power drive system stays at Operation Enabled

1.4.13 0x605E a1_Fault_reaction_option

Data type	Acces	Default	Range	Unit
INT16	RW	2	[0 - 2]	-

This object is used to configure the action to be executed when a fault is detected.

Value	Definition
0	Disable drive function, motor is free to rotate
1	Slow down ramp, mode depending
2	Quick stop ramp, mode depending

Table 2.15: Option of fault reaction

1.5 Emergency error codes

Error codes	Meaning
0x2000	**Current generic error**
0x2100	Current, CANOpen device input side generic
0x2130	Short circuit (input)
0x2200	Current inside the CANOpen device generic
0x2213	Over-current in ramp function
0x2214	Over-current in the sequence
0x2220	Continuous over current (device internal)
0x2250	Short circuit (device internal)
0x2300	Current, CANOpen device output side generic
0x2310	Continuous over current (device output side)
0x2340	Short circuit (motor-side)

Error codes	Meaning
0x3000	**Voltage generic error**
0x3100	Mains voltage generic
0x3110	Mains over-voltage
0x3120	Mains under-voltage
0x3130	Phase failure
0x3134	Phase sequence
0x3200	Voltage inside the CANOpen device generic
0x3210	DC link over-voltage
0x3211	Over-voltage no. 1
0x3212	Over-voltage no. 2
0x3220	DC link under-voltage
0x3221	Under-voltage no. 1
0x3222	Under-voltage no. 2
0x3230	Load error
0x3300	Output voltage generic
0x3310	Output over-voltage
0x4000	**Temperature generic error**
0x4200	Device temperature generic
0x4210	Excess temperature device
0x4220	Too low temperature device
0x4300	Temperature drive
0x4310	Excess temperature drive
0x4320	Too low temperature drive
0x5000	**CANOpen device hardware generic error**
0x5200	Control device hardware
0x5210	Measurement circuit
0x5220	Computing circuit
0x5300	Operating unit
0x5400	Power section
0x5410	Output stages
0x5420	Chopper
0x5440	Contacts
0x5450	Fuses
0x5500	Hardware memory
0x6000	**CANOpen device software generic error**
0x6010	Software reset (watchdog)
0x6300	Data Set
0x6310	Loss of parameters
0x6320	Parameter error
0x7000	**Additional modules generic error**
0x7100	Power additional modules
0x7110	Brake chopper
0x7111	Failure brake chopper
0x7112	Over current brake chopper
0x7120	Motor
0x7121	Motor blocked
0x7122	Motor error or commutation malfunc.
0x7123	Motor tilted
0x7200	Measurement circuit
0x7300	Sensor
0x7301	Tacho fault
0x7305	Incremental sensor 1 fault
0x7310	Speed
0x7320	Position
0x7380	Torque
0x7400	Computation circuit
0x7500	Communication
0x7600	Data storage (external)
0x8000	**Monitoring generic error**

Error codes	Meaning
0x8300	Torque controller
0x8311	Excess torque
0x8312	Difficult start up
0x8313	Standstill torque
0x8321	Insufficient torque
0x8331	Torque fault
0x8400	Velocity controller
0x8500	Position controller
0x8600	Positioning controller
0x8611	Following error
0x8612	Reference limit
0x8613	Homing error
0x8700	Sync controller
0x8800	Winding controller

Table 2.16: Emergency error codes

2 Duty Cycle (DTY)

This mode allows to operate as a direct voltage control on power stage with or without ramp.

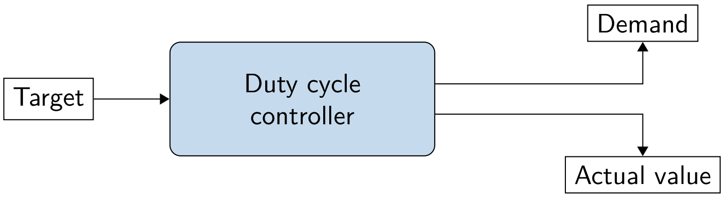


Figure 2.7: Duty cycle mode

2.1 Functional description

2.1.1 Duty cycle mode

Here is the functional diagram with all the parameters.

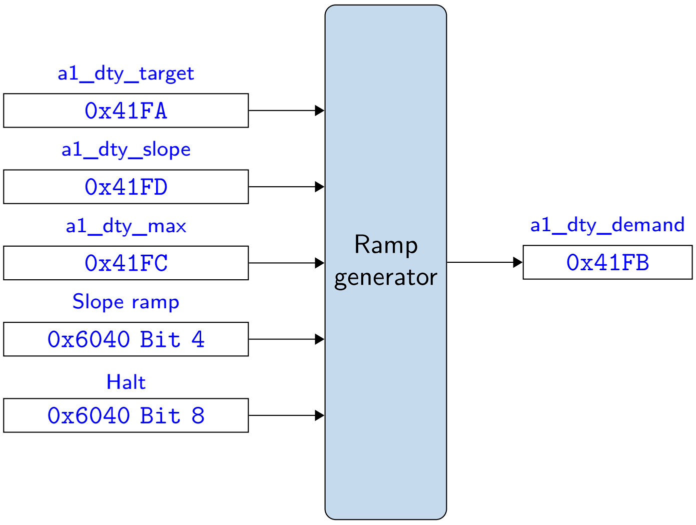


Figure 2.8: Functional diagram of Duty cycle mode

2.1.2 Ramp generator

The Duty cycle mode uses the 0x41FD a1_dty_slope to generate a ramp.

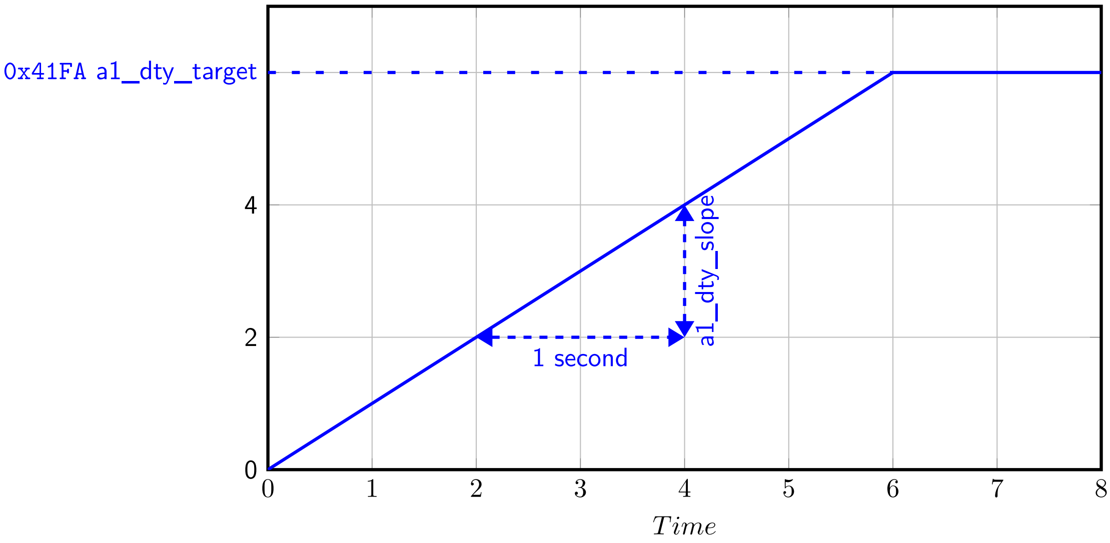


Figure 2.9: Duty cycle mode slope example

2.1.3 Mode activation

The 0x6060 a1_Modes_of_operation object enables speed mode by setting the value to **-1**.

2.2 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

2.2.1 Input objects

Name	Description
0x6060 a1_Modes_of_operation	Modes of operation
0x41FA a1_dty_target	Requested target
0x41FD a1_dty_slope	Slope of ramp
0x41FC a1_dty_max	Max

Table 2.17: Configuration objects

2.2.2 Output objects

Name	Description
0x41FB a1_dty_demand	Generated by the generator trajectory

Table 2.18: Status objects

2.2.3 0x6040 *a1_Controlword*

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	0	-	-

The Duty cycle mode uses bits 8 and 4 of 0x6040 a1_Controlword.

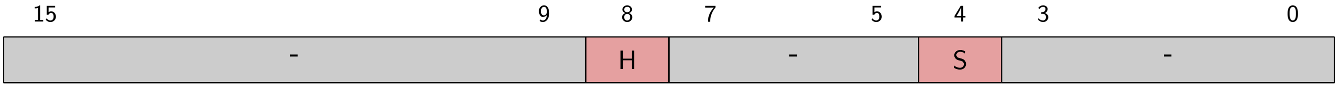


Figure 2.10: Controlword bits for Duty cycle mode

Name	Bit	Value	Definition
H: Halt	8	0	No command
		1	Motor stops according to the halt option code (see 0x605D a1_Halt_option)
S: Slope	4	0	Slope ramp disabled
		1	Slope ramp enabled, with 0x41FD a1_dty_slope

Table 2.19: Definition of controlword bits

2.2.4 0x41FA **a1_dty_target**

Data type	Acces	Default	Range	Unit
INT16	RW,RPDO	0	-	-

This object is the set point. The value shall be given per thousand of rated.

2.2.5 0x41FB **a1_dty_demand**

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

The output value of the trajectory generator. The value shall be given per thousand of rated.

2.2.6 0x41FC **a1_dty_max**

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	30000	-	-

The configured maximum permissible. The value shall be given per thousand of rated.

2.2.7 0x41FD **a1_dty_slope**

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	10000	-	-

The configured rate of change. The value shall be given in units of per thousand of rated per second.

3 Torque (TQ)

This mode allows operation at regulated torque via a ramp generator that can be configured through various settings : input torque adjustment, slope ramp adjustment and other optional parameters.

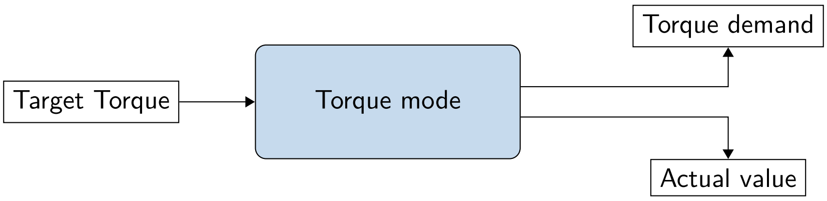


Figure 2.11: Torque mode

3.1 Functional description

3.1.1 Torque mode

Here is the functional diagram with all the parameters.

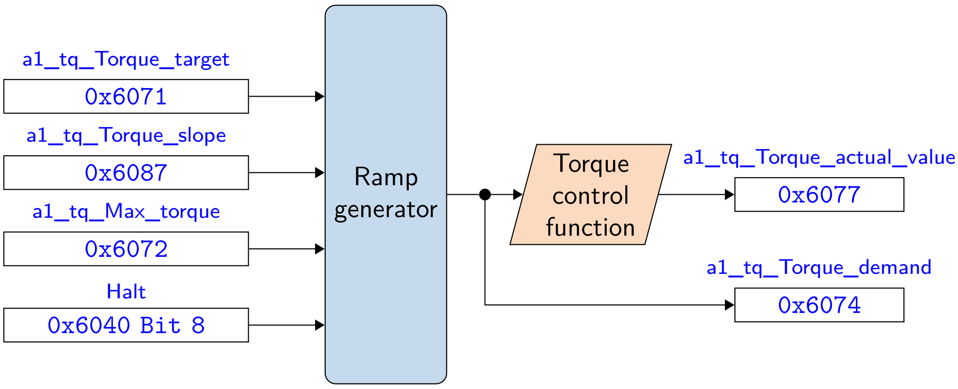


Figure 2.12: Functional diagram of Torque mode

3.1.2 Ramp generator

The Torque mode uses the 0x6087 a1_tq_Torque_slope to generate a ramp.

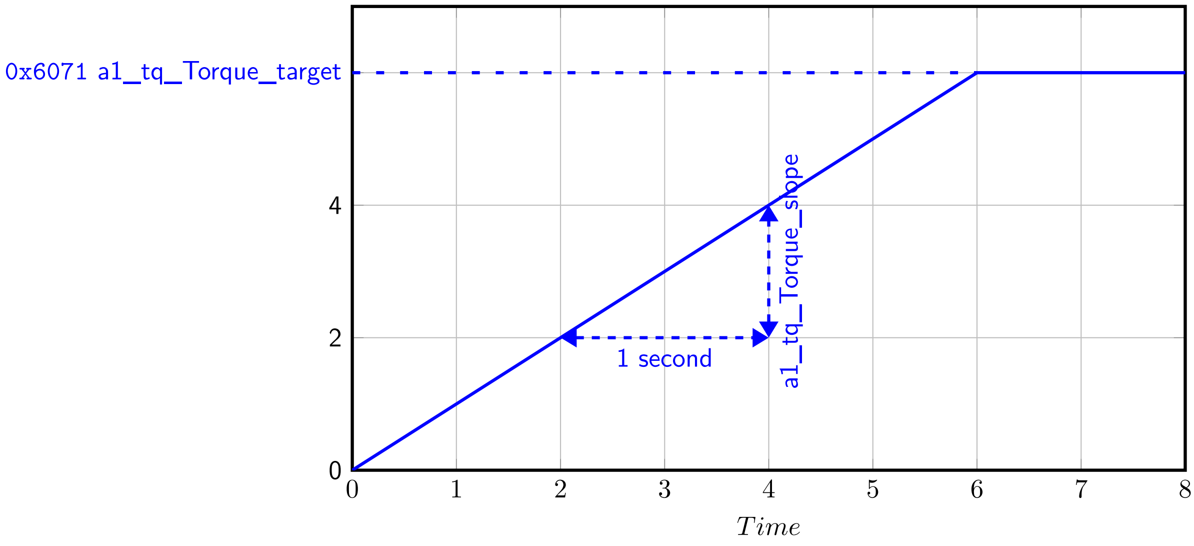


Figure 2.13: Feature of torque ramp

3.1.3 Mode activation

The 0x6060 a1_Modes_of_operation object enables speed mode by setting the value to 4.

3.2 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for ohter axis.

3.2.1 Input objects

Name	Description
0x6060 a1_Modes_of_operation	Modes of operation
0x6071 a1_tq_Torque_target	Requested target torque
0x6087 a1_tq_Torque_slope	Slope of ramp
0x6072 a1_tq_Max_torque	Max of torque

Table 2.20: Configuration objects

3.2.2 Output objects

Name	Description
0x6041 *a1_Statusword*	Status
0x6074 a1_tq_Torque_demand	Torque generated by the generator trajectory
0x6077 a1_tq_Torque_actual_value	Actual Torque value

Table 2.21: Status objects

3.2.3 0x6040 *a1_Controlword*

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	0	-	-

The Torque mode uses only bit 8 of 0x6040 a1_Controlword.

15 9	8	7 0
-	H	-

Name	Bit	Value	Definition
H: Halt	8	0	No command
		1	Motor stops according to the halt option code (see 0x605D a1_Halt_option)

Table 2.22: Definition of Controlword bits (bit 8)

3.2.4 0x6041 *a1_Statusword*

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

The Torque mode uses only bit 11 (Internal limit active) of 0x6041 a1_Statusword.

15 11	10	9 0
-	TR	-

Name	Bit	Value	Definition
TR	10	0	Halt = 0: Target torque not reached
			Halt = 1: Axis decelerates
		1	Halt = 0: Target torque reached
			Halt = 1: Velocity of axis is 0

Table 2.23: Definition of statusword bits (bit 10)

3.2.5 0x6071 a1_tq_Torque_target

Data type	Acces	Default	Range	Unit
INT16	RW,RPDO	0	-	-

This object is torque set point. The value shall be given per thousand of rated torque.

3.2.6 0x6072 a1_tq_Max_torque

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	30000	-	-

The configured maximum permissible torque in the motor. The value shall be given per thousand of rated torque.

3.2.7 0x6074 a1_tq_Torque_demand

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

The output value of the trajectory generator. The value shall be given per thousand of rated torque.

3.2.8 0x6077 a1_tq_Torque_actual_value

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

The actual value of the torque. It shall correspond to the instantaneous torque in the motor. The value shall be given per thousand of rated torque.

3.2.9 0x6078 a1_tq_Current_actual_value

Data type	Acces	Default	Range	Unit
INT16	RO	-	-	-

The actual value of the current. It shall correspond to the current in the motor. The value shall be given per thousand of rated current.

3.2.10 0x6087 a1_tq_Torque_slope

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	10000	-	-

The configured rate of change of torque. The value shall be given in units of per thousand of rated torque per second.

3.2.11 0x6088 a1_tq_Torque_profile_type

Data type	Acces	Default	Range	Unit
INT16	RW	0	-	-

The configured type of profile used to perform a torque change.

Value	Definition
0	Linear ramp

Table 2.24: Option of 0x6088 *a1_tq_Torque_profile_type*

4 Velocity (VL)

This mode allows operation at regulated speed via a ramp generator that can be configured through various settings : input speed adjustment, acceleration and deceleration ramp adjustment and other optional parameters.

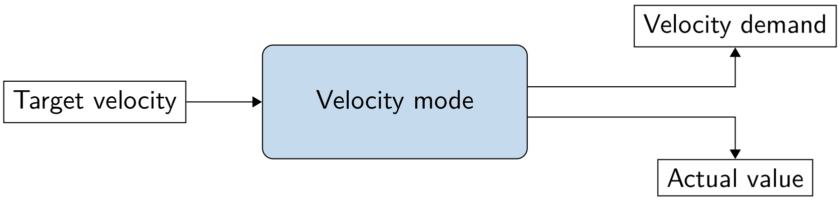
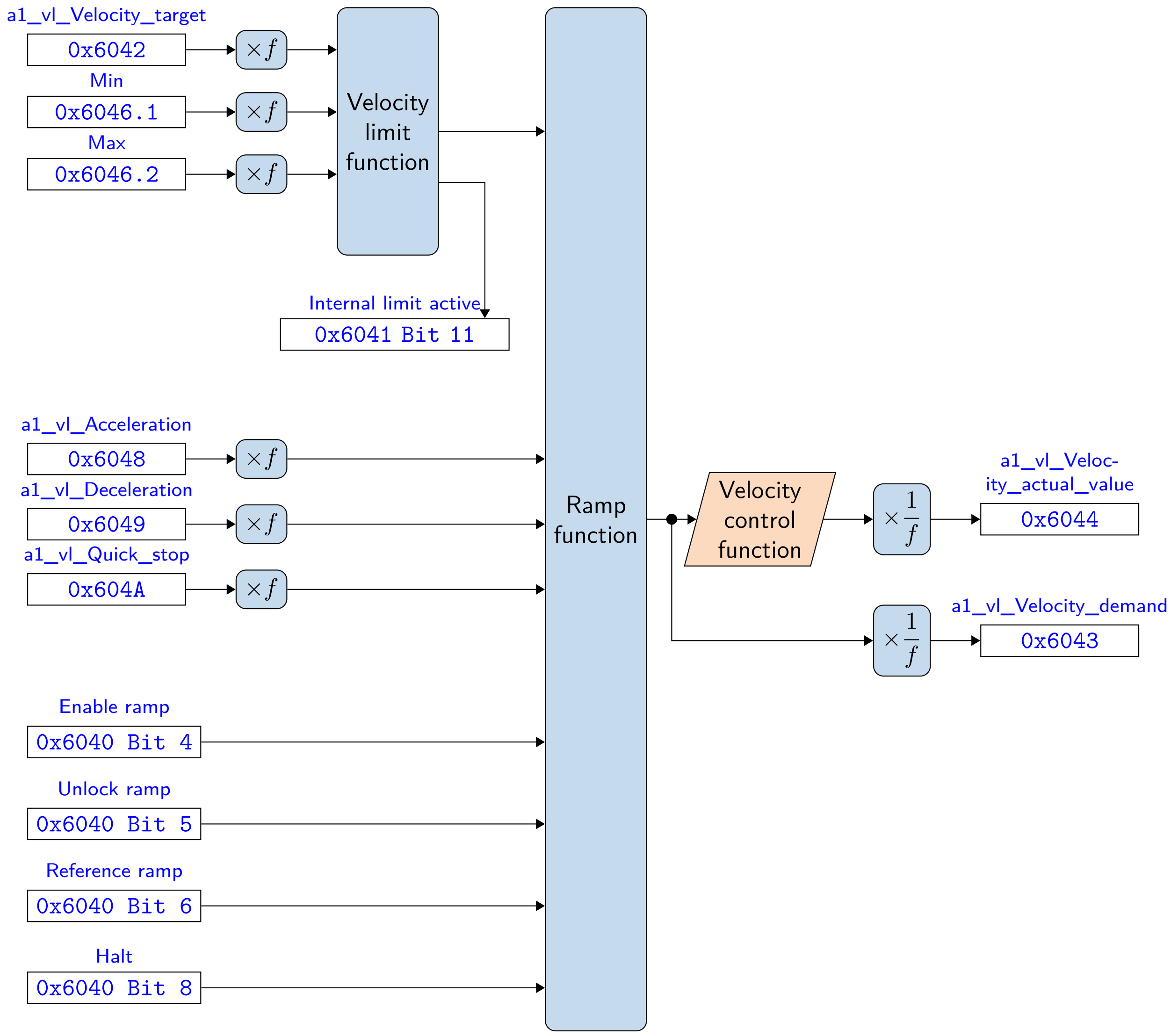


Figure 2.14: Velocity mode

4.1 Functional description

4.1.1 Velocity mode

Here is the functional diagram with all the parameters.



$$\boxed{f = \frac{\texttt{0x604B.1}}{\texttt{0x604B.2}} * \frac{\texttt{0x604C.1}}{\texttt{0x604C.2}}}$$

Figure 2.15: Functional diagram of Velocity mode

4.1.2 Ramp generator

The velocity profile is used to limit the increase or decrease in speed:

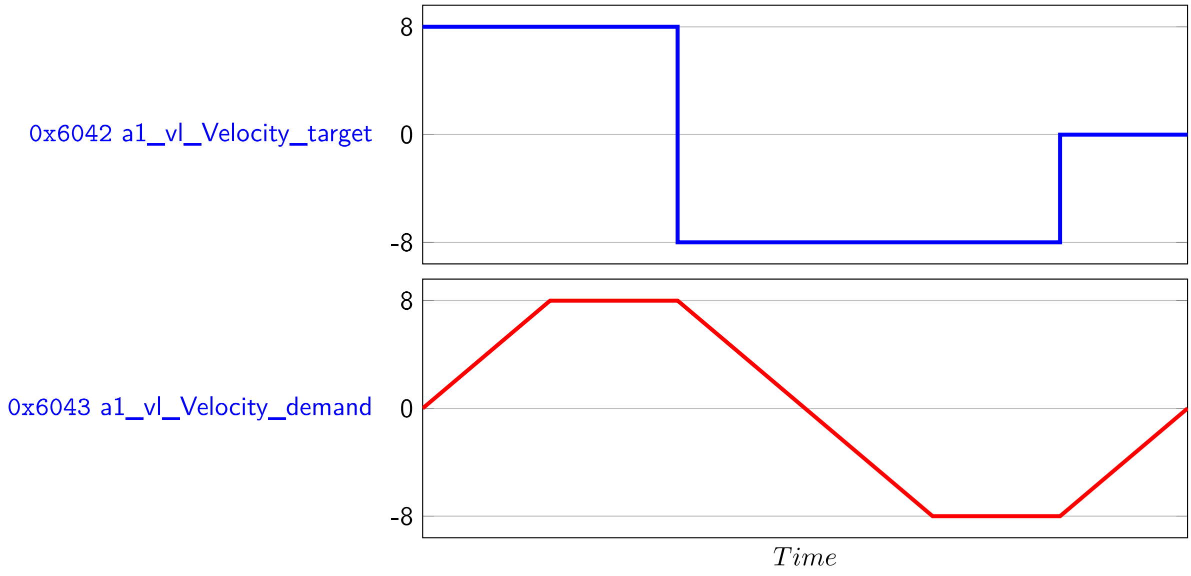


Figure 2.16: Velocity profile

4.1.3 Factor function

This function transforms user units into internal units.

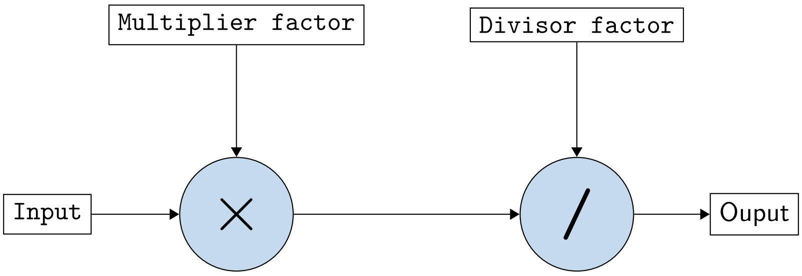


Figure 2.17: Factor function

This function transforms internal units to user units.

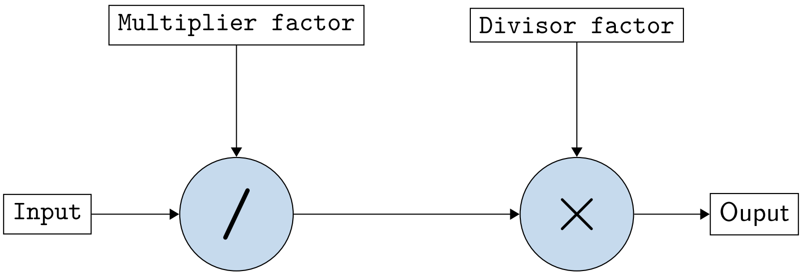


Figure 2.18: Reverse factor function

4.1.4 Mode activation

The 0x6060 a1_Modes_of_operation object enables speed mode by setting the value to 2.

4.2 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

4.2.1 Input objects

Name	Description
0x6060 a1_Modes_of_operation	Modes of operation
0x6040 *a1_Controlword*	Control
0x6042 a1_vl_Velocity_target	Requested velocity
0x6046 a1_vl_Min_max	Maximal and minimal velocity
0x6048 a1_vl_Acceleration	Maximal cruising acceleration
0x6049 a1_vl_Deceleration	Maximal cruising deceleration
0x604A a1_vl_Quick_stop	Maximal cruising deceleration of Quick Stop

Table 2.25: Configuration objects

4.2.2 Output objects

Name	Description
0x6041 *a1_Statusword*	Status
0x6043 a1_vl_Velocity_demand	Velocity generated by the generator trajectory
0x6044 a1_vl_Velocity_actual_value	Actual velocity value

Table 2.26: Status objects

4.2.3 0x6040 *a1_Controlword*

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	0	-	-

The Velocity mode uses bits 4,5,6 and 8 of 0x6040 a1_Controlword.

15	9	8	7	6	5	4	3	0
-		H	-	RR	UR	ER	-	

Name	Bit	Value	Definition
ER	4	0	Disable ramp
		1	Enable ramp
UR	5	0	Ramp output value is locked
		1	Ramp output value = Ramp input value
RR	6	0	Ramp input set to zero
		1	Ramp input used ramp reference
H	8	0	No command
		1	Axis stopped accordingly to halt option code (see 0x605D a1_Halt_option

Table 2.27: Definition of controlword bits (bit 4, bit 5, bit 6 and bit 8)

Halt H	Enable ramp ER	Unlock ramp UR	Reference ramp RR	Priority
0	0	X	X	Disable ramp
0	1	0	X	Lock ramp
0	1	1	0	Ramp set to zero with normal ramp
0	1	1	1	Normal ramp Reference
1	X	X	X	Ramp set to zero with: see Halt option 0x605D

Table 2.28: Priority of controlword bits (Halt/Enable/Unlock/Reference)

4.2.4 0x6041 *a1_Statusword*

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

The Velocity mode uses only bit 11 of 0x6041 a1_Statusword.

15 12	11	10 0
-	ILA	-

Name	Bit	Value	Definition
ILA	11	0	No internal limit
		1	Internal limit is active

Table 2.29: Definition of statusword bits (bit 11)

4.2.5 0x6042 a1_vl_Velocity_target

Data type	Acces	Default	Range	Unit
INT16	RW,RPDO	0	-	-

This object defines the target speed of the system. Positive values mean forward direction and negative values mean reverse direction.

The object 0x604B a1_vl_Set_point_factor and the 0x604C a1_vl_Dimension_factor allow to modify the units or the resolution.

4.2.6 0x6043 a1_vl_Velocity_demand

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

This object provides the instantaneous velocity generated by the ramp function. Positive values mean forward direction and negative values mean reverse direction.

4.2.7 0x6044 a1_vl_Velocity_actual_value

Data type	Acces	Default	Range	Unit
INT16	RO,TPDO	-	-	-

This object provide the velocity at the motor spindle. Positive values mean forward direction and negative values mean reverse direction.

4.2.8 0x6046 a1_vl_Min_max

Index	Name		Object type
0x6046	a1_vl_Min_max		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT32	RW,RPDO	[0 - 0xFFFFFFFF]	-

Subindex	Name
1	Min
2	Max

This object defines the minimum and maximum velocity. The sub-object 0x6046 a1_vl_Min_max indicates a negative or positive minimum velocity. The sub-object 0x6046 a1_vl_Min_max indicates a negative or positive maximum velocity.

The object 0x604B a1_vl_Set_point_factor and the 0x604C a1_vl_Dimension_factor allow to modify the units or the resolution.

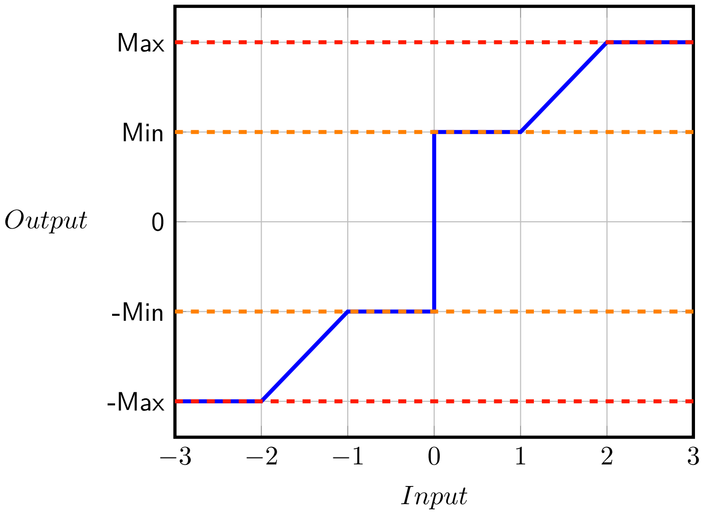


Figure 2.19: Minimum and maximum velocity characteristic

Note: Special case when going from "Switched On" state to "Operation Enabled" state: if the speed is equal to 0 when passing, then the demand remains at zero.

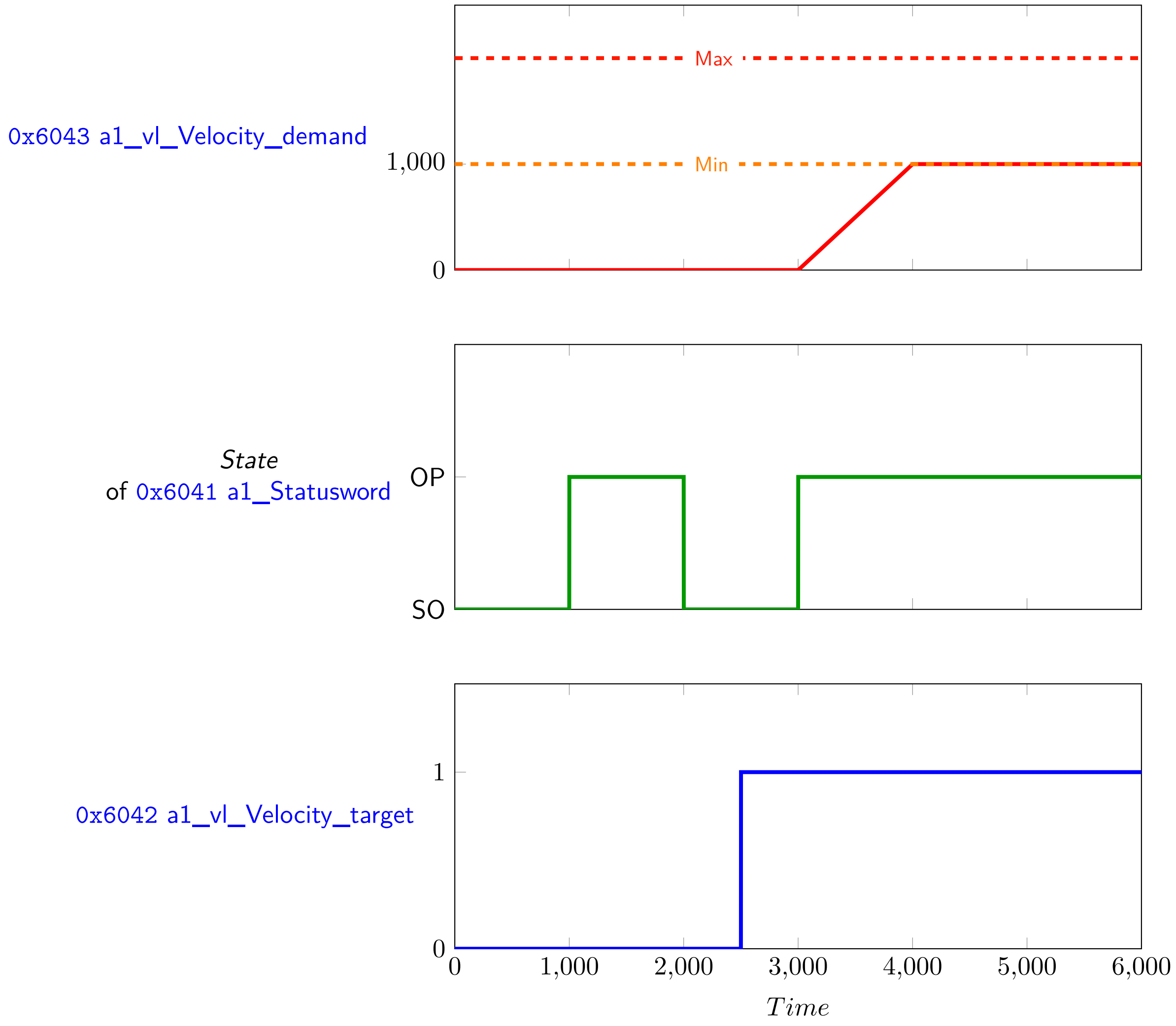


Figure 2.20: Special case when going from "Switched On (SO)" state to "Operation Enabled (OP)" state

4.2.9 0x6048 a1_vl_Acceleration

Index	Name	Object type
0x6048	a1_vl_Acceleration	RECORD

Subindex	Name	Data type
1	Delta_speed	UINT32
2	Delta_time	UINT16

This object defines the acceleration ramp, configurable with delta speed and delta time, as shown by the following equation and graphic:

$$Acceleration = \frac{\texttt{0x6048.1}}{\texttt{0x6048.2}} = \frac{Delta\_speed}{Delta\_time} \quad (2.1)$$

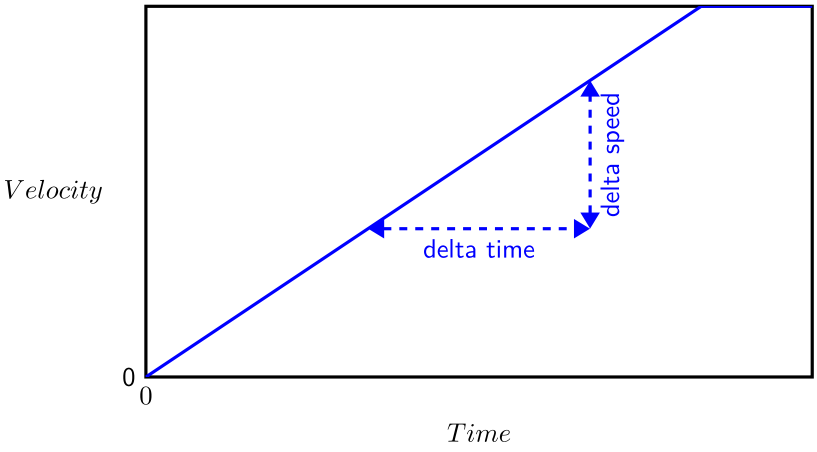


Figure 2.21: Feature of acceleration ramp

The object 0x604B a1_vl_Set_point_factor and the 0x604C a1_vl_Dimension_factor allow to modify the units or the resolution.

The value of delta speed cannot be equal to zero. The value of delta time shall be given in s and cannot be equal to zero.

0x6048.1 ***Delta_speed***

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	[0x1 - 0xFFFFFFFF]	-

0x6048.2 ***Delta_time***

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	1	[0x1 - 0xFFFF]	ms

4.2.10 0x6049 a1_vl_Deceleration

Index	Name	Object type
0x6049	a1_vl_Deceleration	RECORD

Subindex	Name	Data type
1	Delta_speed	UINT32
2	Delta_time	UINT16

This object defines the deceleration ramp, configurable with delta speed and delta time, as shown by the following equation and graphic:

$$Deceleration = \frac{\texttt{0x6049.1}}{\texttt{0x6049.2}} = \frac{Delta\_speed}{Delta\_time} \tag{2.2}$$

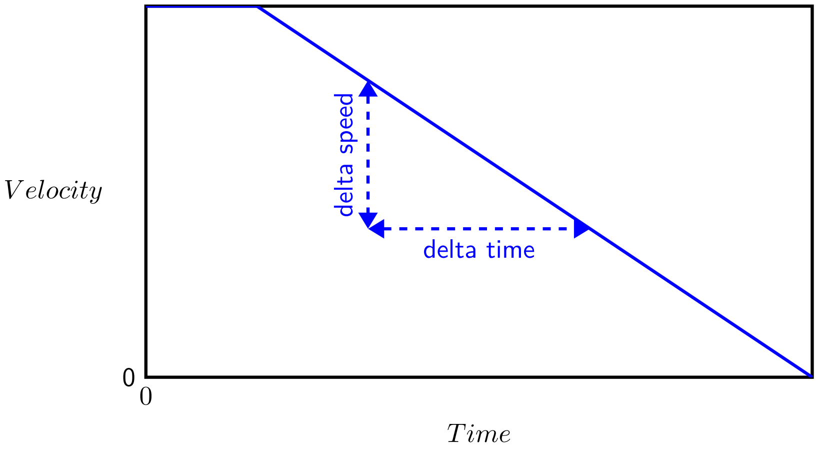


Figure 2.22: Feature of deceleration ramp

The object 0x604B a1_vl_Set_point_factor and the 0x604C a1_vl_Dimension_factor allow to modify the units or the resolution.

The value of delta speed cannot be equal to zero. The value of delta time shall be given in s and cannot be equal to zero.

Note: If delta speed is equal to 0 or if the object is not implemented, the values of 0x6048 a1_vl_Acceleration will be used

`0x6049.1` ***Delta_speed***

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	[0x1 - 0xFFFFFFFF]	-

`0x6049.2` ***Delta_time***

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	1	[0x1 - 0xFFFF]	ms

4.2.11 `0x604A` a1_vl_Quick_stop

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	[0x1 - 0xFFFFFFFF]	-

This object defines the deceleration ramp for a quick stop, configurable with delta speed and delta time, as shown by the following equation and graphic:

$$Deceleration = \frac{\texttt{0x604A.1}}{\texttt{0x604A.2}} = \frac{Delta\_speed}{Delta\_time} \tag{2.3}$$

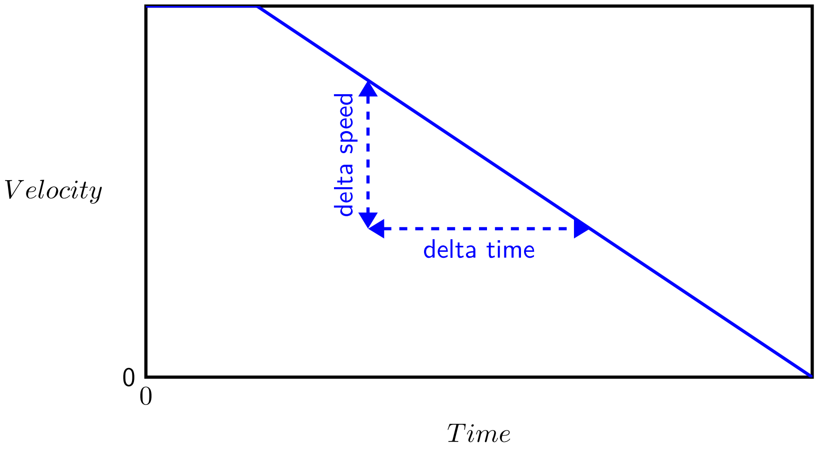


Figure 2.23: Feature of deceleration ramp for quick stop

The object 0x604B a1_vl_Set_point_factor and the 0x604C a1_vl_Dimension_factor allow to modify the units or the resolution.

The value of delta speed cannot be equal to zero. The value of delta time shall be given in millisecond and cannot be equal to zero.

Note: If delta speed is equal to 0 or if the object is not implemented, the values of 0x6048 a1_vl_Acceleration will be used

`0x604A.1` ***Delta_speed***

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	[0x1 - 0xFFFFFFFF]	-

`0x604A.2` ***Delta_time***

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	1	[0x1 - 0xFFFF]	ms

4.2.12 0x604B a1_vl_Set_point_factor

Index	Name		Object type
0x604B	a1_vl_Set_point_factor		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT16	RW,RPDO	[0x1 - 0x7FFF]	-

Subindex	Name
1	Numerator
2	Denominator

The resolution or directing range of the velocity set point can be modified with numerator and denominator of this object. This factor has no influence on the limit function and the ramp function.

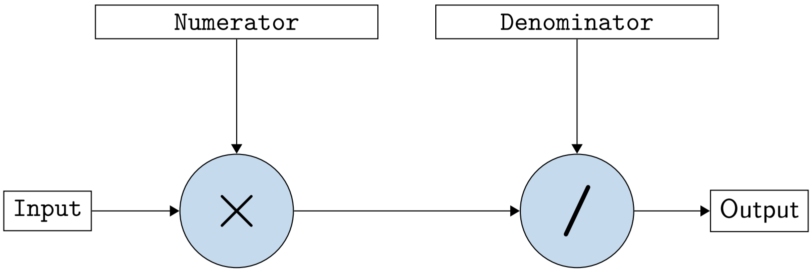


Figure 2.24: Factor function

4.2.13 0x604C a1_vl_Dimension_factor

Index	Name		Object type
0x604C	a1_vl_Dimension_factor		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT32	RW,RPDO	[0x1 - 0x7FFF]	-

Subindex	Name
1	Numerator
2	Denominator

This object serves to change unit or to scale specific units of the user. This object convert the specific velocity units to the LSB per millisecond or LSB per millisecond to specific velocity units.

$$Velocity_{internal}[LSB/ms] = Velocity[user\ units] \times \texttt{0x604C}a1\_vl\_Dimension\_factor[LSB/ms/user\ units] \quad (2.4)$$

5 Profile Position (PP)

This mode allows positional movement by applying a target position to the generator trajectory. It generates a position request for the position control loop.

The trajectory generator takes into account the limit values of speed, acceleration, deceleration during the movement.

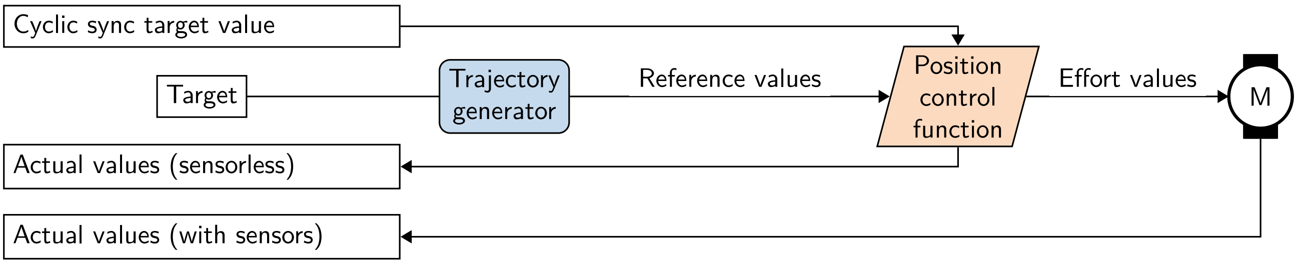


Figure 2.25: Profile Position Mode

5.1 Functional description

The setting of set points is controlled by 0x6040 *a1_Controlword*: *New set point, Change set-immediately, Absolute or relative* value as well as status of 0x6041 *a1_Statusword* : *Set point acknowledge* and *Target reached*

This profile position mode works with a single set point, so only one set point is expected by the device.

5.1.1 Profile position mode

The general structure of this mode is shown in the following diagram with all objects involved for trajectory generator.

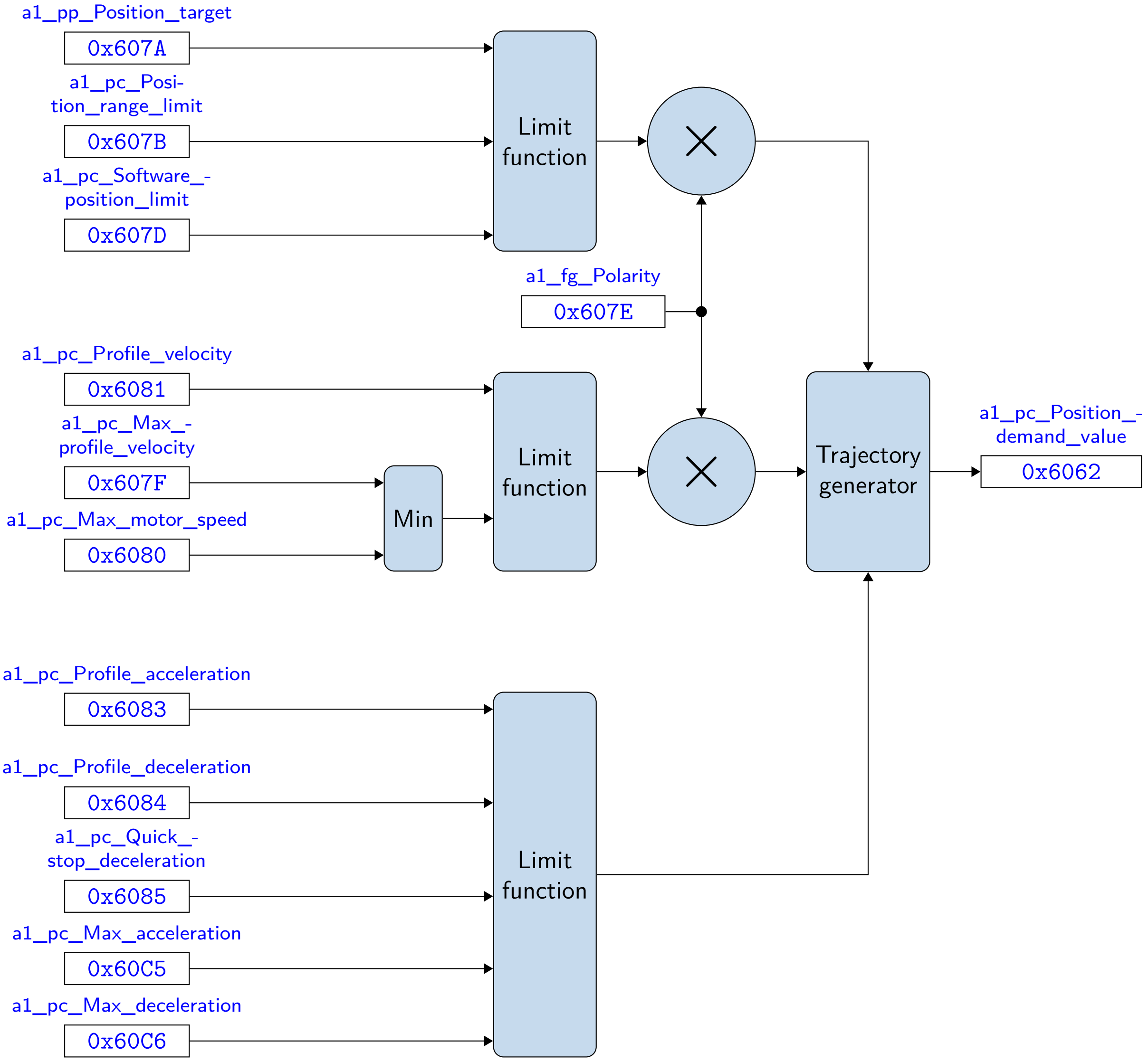


Figure 2.26: Functional diagram of Profile Position Mode

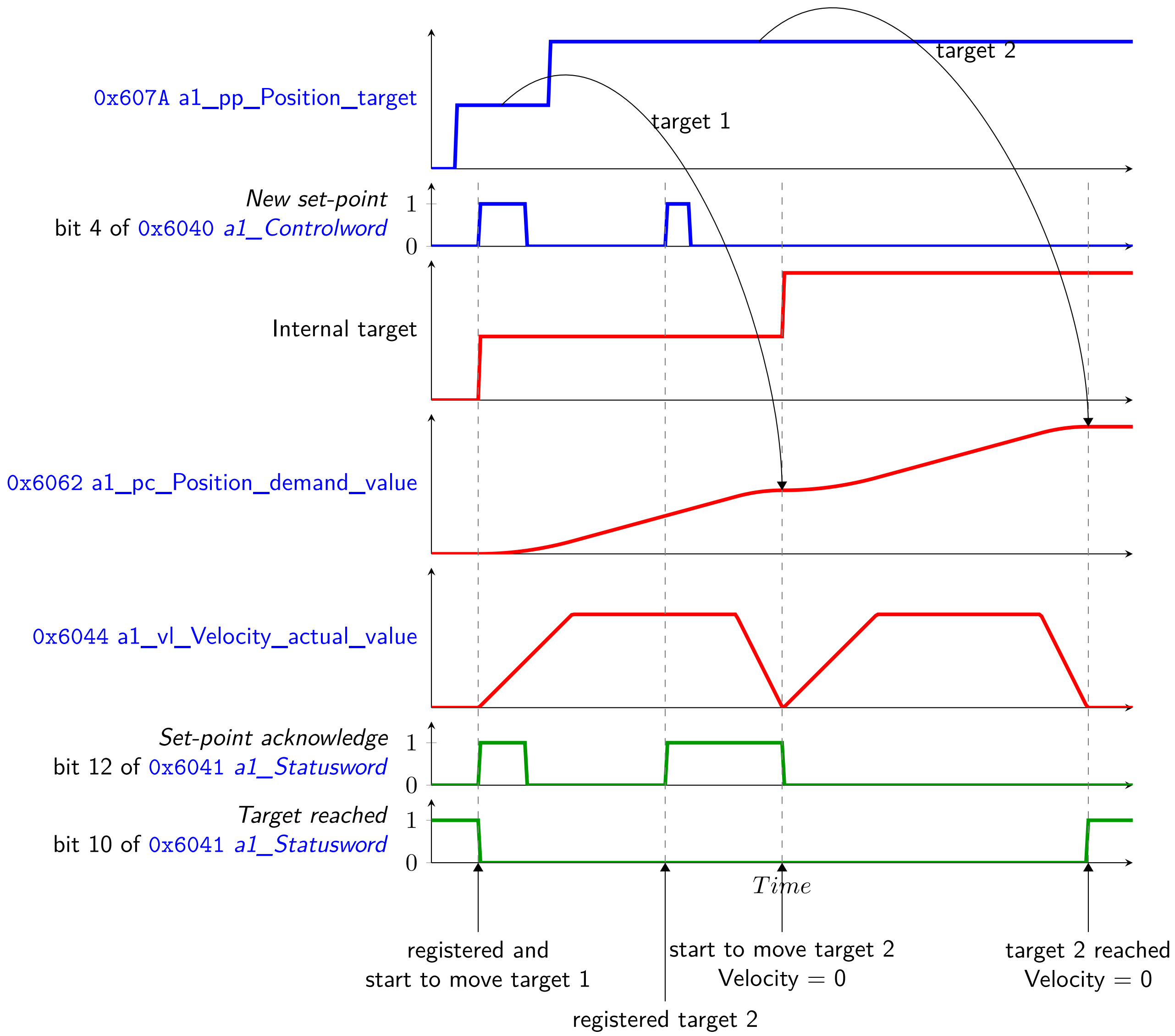
5.1.2 Processing of a new setpoint

A new set point received by the device will apply on a rising edge of the *New set point bit* of the 0x6040 *a1_Controlword*. *Change set-immediately and Absolute or relative* to modify processing of a new set point.

A new 0x6081 a1_pc_Profile_velocity, 0x6083 a1_pc_Profile_acceleration and 0x6084 a1_pc_Profile_deceleration as well as control their respective limits are taken into account when executing the new set point in accordance with the two methods described below.

- Two methods are available : Set of set-points and Single set-point according to *Change set immediately bit*, when a set-point is being executed and the *New set-point bit* is activated

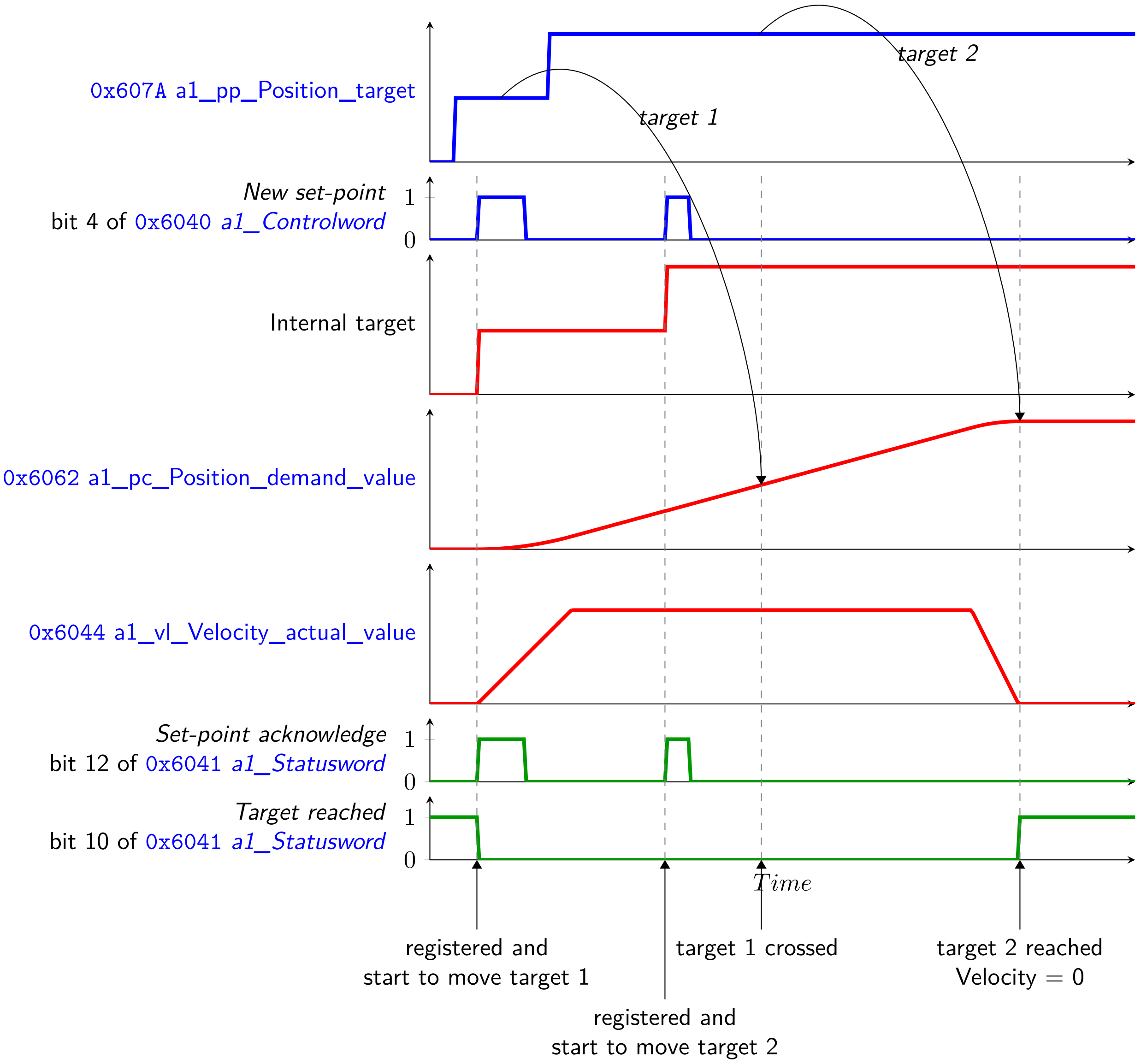
  - **Set of set-points**: *Change set immediately bit* = 0.
    The new set point will not be processed until the previous has been reached and set the *Set-point acknowledgment bit* in the 0x6041 *a1_Statusword* to 1 until the previous set-point has been reached.



All graphics are generated from real device.

Figure 2.27: Set of set points method

- **Single set point**: *Change set immediately bit* = 1:
  The new set point will be processed immediately and set the *Set-point acknowledgment bit* in the 0x6041 *a1_Statusword* to 1 until the *New set point bit* is set to 0.



All graphics are generated from real device.

Figure 2.28: Single set-point method

- *Absolute or relative* bit of 0x6040 a1_Controlword is defined if the *New set-point bit* is:
  - absolute value, position reached = 0x607A a1_pp_Position_target
  - relative value, position reached = 0x607A a1_pp_Position_target + 0x6062 a1_pc_Position_demand_value.

5.1.3 Trajectory generator

The trajectory generator uses the following motion profile: Linear ramp (trapezoidal profile). Below are the motion profiles of acceleration, velocity and position:

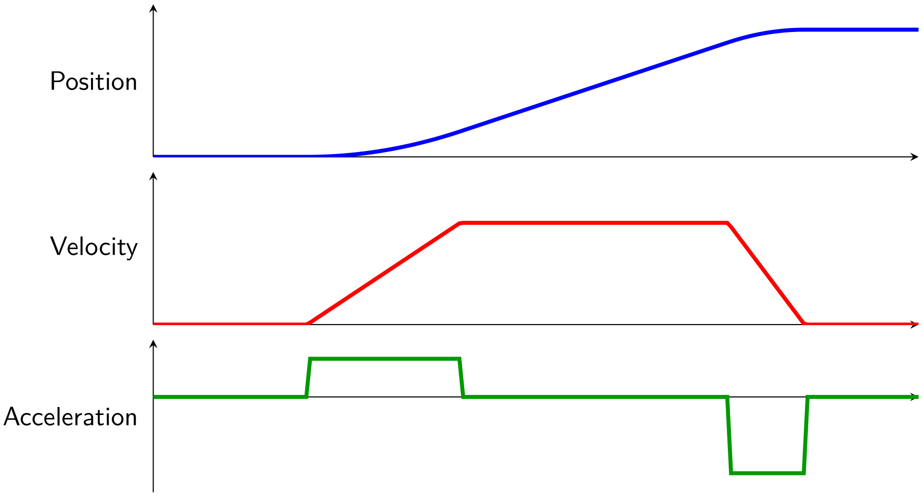


Figure 2.29: Linear ramp (trapezoidal profile)

5.1.4 Mode activation

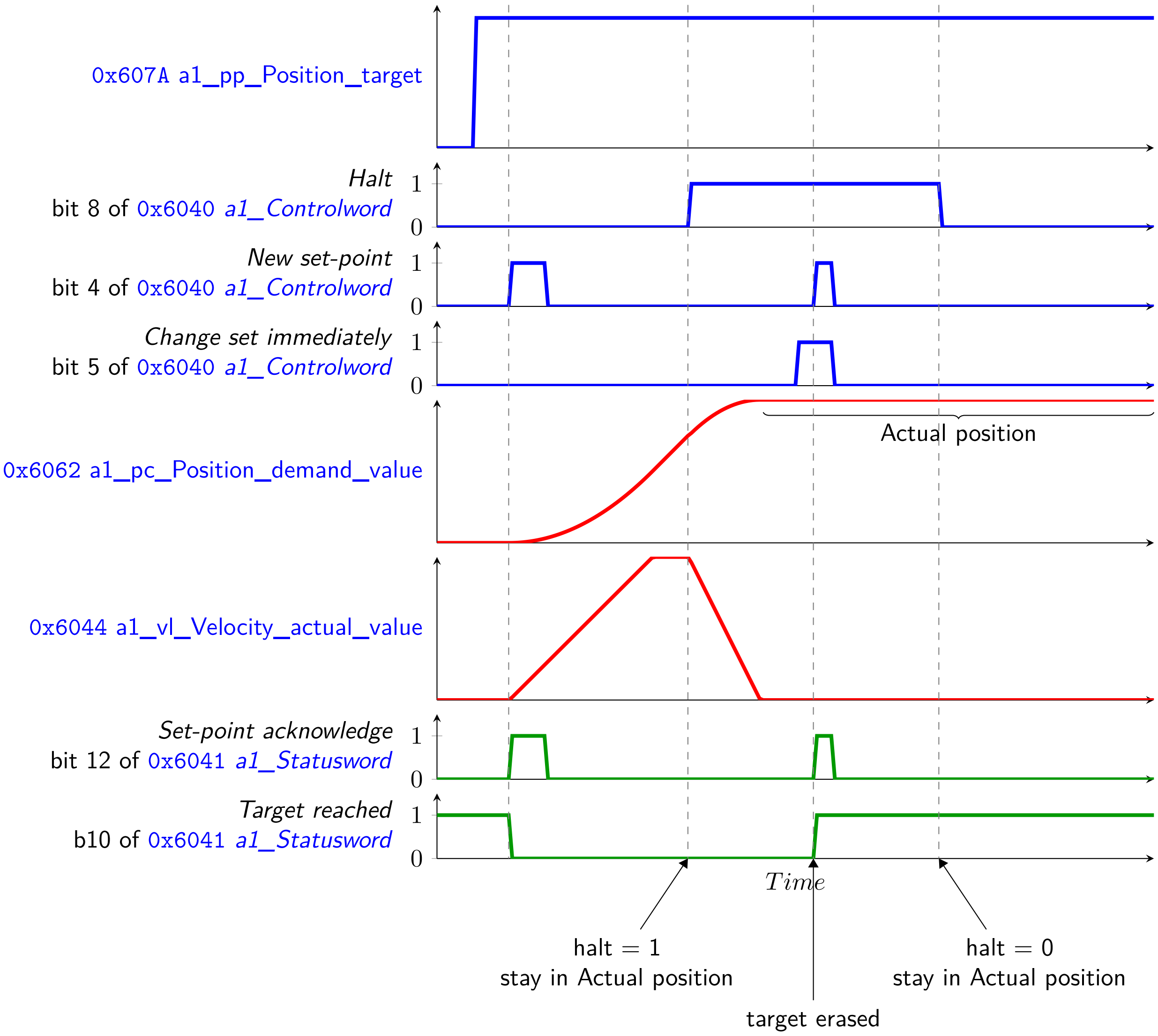
The 0x6060 a1_Modes_of_operation object enables speed mode by setting the value to 1.

5.1.5 Specific usage

Halt bit of Controlword The stop bit is used to stop movement by decelerating the axis, so that the speed approaches zero. The behavior of halt is in agreement to 0x605D a1_Halt_option. Two specific uses are described below :

- **Halt bit in conjunction with new set point bit:**
  A new set point can be defined by activating the *new set point bit* and setting the new set point during the active shutdown. After releasing the stop bit, the actual set point processing continues with the new set point set during the shutdown.

- **Erase processed set point during Halt:**
  This procedure allows to erase processed set point and sets the actual position as a new target. After releasing the *halt bit* the position remains at the actual position.



All graphics are generated from real device.

Figure 2.30: Erase processed set point

Inter state transitions The behavior when transitioning from the Operation Enabled state to another state is to stop the movement by decelerating the axis, so that the speed approaches zero. In agreement with the Transition options.

Inter mode transitions These transitions are possible from the Operation Enabled state. The behavior consists in stopping the movement by decelerating the axis, so that the speed approaches zero. The behavior is in agreement to 0x605D a1_Halt_option. Once the velocity is equal to zero, the mode change is realized.

5.2 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

5.2.1 Input objects

Name	Description
0x6060 a1_Modes_of_operation	Modes of operation
0x6040 *a1_Controlword*	Control
0x607A a1_pp_Position_target	Requested target position
0x607B a1_pc_Position_range_limit	Maximal and minimal position range limits
0x607D a1_pc_Software_position_limit	Maximal and minimal software position limits
0x607F a1_pc_Max_profile_velocity	Maximal speed limit
0x6080 a1_pc_Max_motor_speed	Maximal speed limit for the motor
0x6081 a1_pc_Profile_velocity	Maximal cruising speed
0x6083 a1_pc_Profile_acceleration	Maximal cruising acceleration
0x6084 a1_pc_Profile_deceleration	Maximal cruising deceleration
0x60C5 a1_pc_Max_acceleration	Maximal acceleration
0x60C6 a1_pc_Max_deceleration	Maximal deceleration

Table 2.30: Configuration objects

5.2.2 Output objects

Name	Description
0x6041 *a1_Statusword*	Status
0x6062 a1_pc_Position_demand_value	Position generated by the generator trajectory
0x6064 a1_pc_Position_actual_value	Actual position value

Table 2.31: Status objects

5.2.3 0x6040 *a1_Controlword*

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	0	-	-

The bit 4, bit 5, bit 6, and bit 8 of 0x6040 a1_Controlword are used in Profile Position Mode.

15	9	8	7	6	5	4	3	0
-		H	-	abs/rel	CSI	NSP	-	

Name	Bit	Value	Definition
NSP	4	0	Target not applied
		0 → 1	Target applied
CSI	5	0	Wait until the actual target position is complete before applying the new position.
		1	New target position is applied to immediately.
abs/rel	6	0	Target position is in absolute value
		1	Target position is in relative value
H	8	0	No command
		1	Axis stopped accordingly to halt option code (see 0x605D a1_Halt_option)

Table 2.32: Definition of controlword bits (bit 4, bit 5, bit 6, and bit 8)

5.2.4 0x6041 *a1_Statusword*

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

The bit 10, 12, 13 of 0x6041 a1_Statusword is used in Profile Position Mode.

15	14	13	12	11	10	9 … 0
-		FO	SPA	-	TA	-

Name	Bit	Value	Definition
TA	10	0	Halt = 0 : Target position not reached Halt = 1 : Axis decelerates
		1	Halt = 0 : Target position reached Halt = 1 : Velocity of axis is 0
SPA	12	0	Previous target already processed, waiting for a new target
		1	Target in process
FO	13	0	No following error
		1	Following error

Table 2.33: Definition of statusword bits (bit 11)

5.2.5 0x607A a1_pp_Position_target

Data type	Acces	Default	Range	Unit
INT32	RW,RPDO	0	-	-

Target position requested from the position generator to which it should go using the current settings of the motion control parameters. This value is given in user-defined position units.

5.2.6 0x6062 a1_pc_Position_demand_value

Data type	Acces	Default	Range	Unit
INT32	RO,TPDO	-	-	-

Position generated by the generator trajectory and used as input to the position control. This value is given in user-defined position units.

5.2.7 0x6064 a1_pc_Position_actual_value

Data type	Acces	Default	Range	Unit
INT32	RO,TPDO	-	-	-

Actual position value of measuring device. This value is given in user-defined position units.

5.2.8 0x607B a1_pc_Position_range_limit

Index	Name		Object type
0x607B	a1_pc_Position_range_limit		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT32	RW,RPDO	-	-

Subindex	Name
1	Min
2	Max

Defines maximal and minimal position range limits, allowing the numerical range of the target value to be limited. The value automatically wraps to the other end of the range when the target value reaches or exceeds these limits. To prevent overflow of the input value, set the software position limits. To disable the position range limits, the min and max must be set to 0. These values are given in user-defined position units.

5.2.9 0x607D a1_pc_Software_position_limit

Index	Name		Object type
0x607D	a1_pc_Software_position_limit		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT32	RW,RPDO	-	-

Subindex	Name
1	Min
2	Max

Defines maximal and minimal software position limits, defines the absolute position limits for the position demand value and the position actual value. To disable the software position limits, the min and max be set to 0. These values are given in user-defined position units.

5.2.10 0x607F a1_pc_Max_profile_velocity

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	16384	-	-

Defines the maximum speed limit. This value is given in user-defined units.

5.2.11 0x6080 a1_pc_Max_motor_speed

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	16384	-	-

Defines the maximum speed limit for the motor. This value is given in user-defined units.

5.2.12 0x6081 a1_pc_Profile_velocity

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	16384	-	-

Defines the maximum cruising speed. This value is given in user-defined units.

5.2.13 0x6083 a1_pc_Profile_acceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	-	-

Defines the maximum cruising acceleration. This value is given in user-defined units.

5.2.14 0x6084 a1_pc_Profile_deceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	-	-

Defines the maximum cruising deceleration. This value is given in user-defined units.

5.2.15 0x6086 a1_pp_Motion_profile_type

Data type	Acces	Default	Range	Unit
INT16	RW,RPDO	0	-	-

Defines the type of ramp of the motion profile used.

Value	Description
0	Trapezoidal profile

Table 2.34: 0x6086 *a1_pp_Motion_profile_type*options

5.2.16 0x60C5 a1_pc_Max_acceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	100000	-	-

Defines the maximum acceleration. This value is given in user-defined units.

5.2.17 0x60C6 a1_pc_Max_deceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	100000	-	-

Defines the maximum deceleration. This value is given in user-defined units.

6 Interpolated Position (IP)

This mode allows the axis to be moved according to several target positions over a defined period. For each period, the axis moves to the next memorized position by calculating the intermediate points at an appropriate speed.

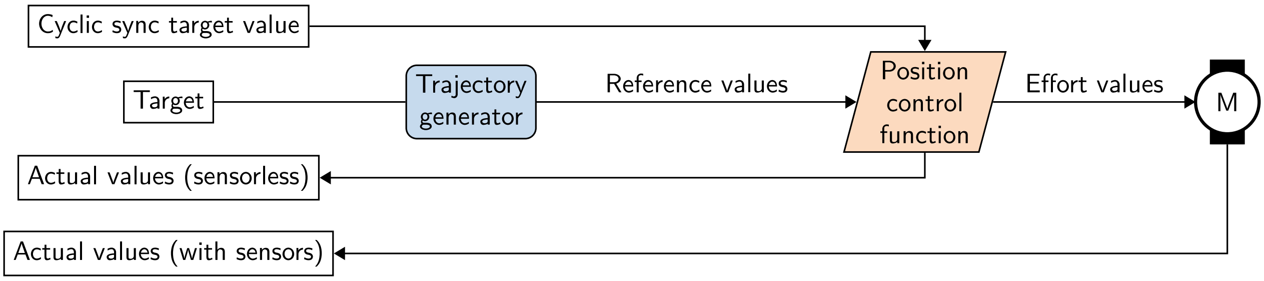


Figure 2.31: Interpolated Position mode

6.1 Functional description

Here is the functional diagram with all the parameters.

6.1.1 Interpolated Position mode

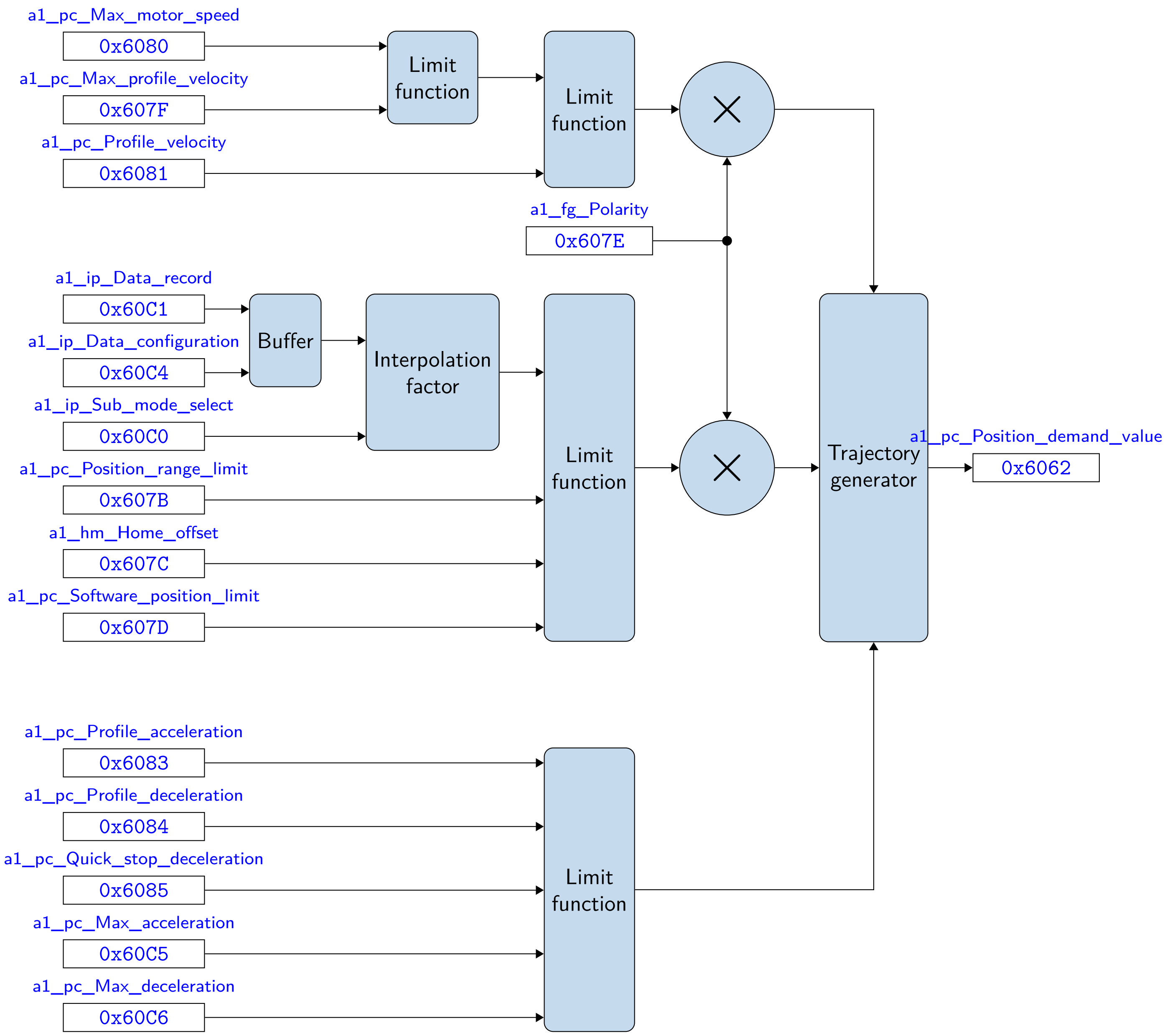


Figure 2.32: Functional diagram of Interpolated Position mode

6.1.2 State machine of Interpolated position mode

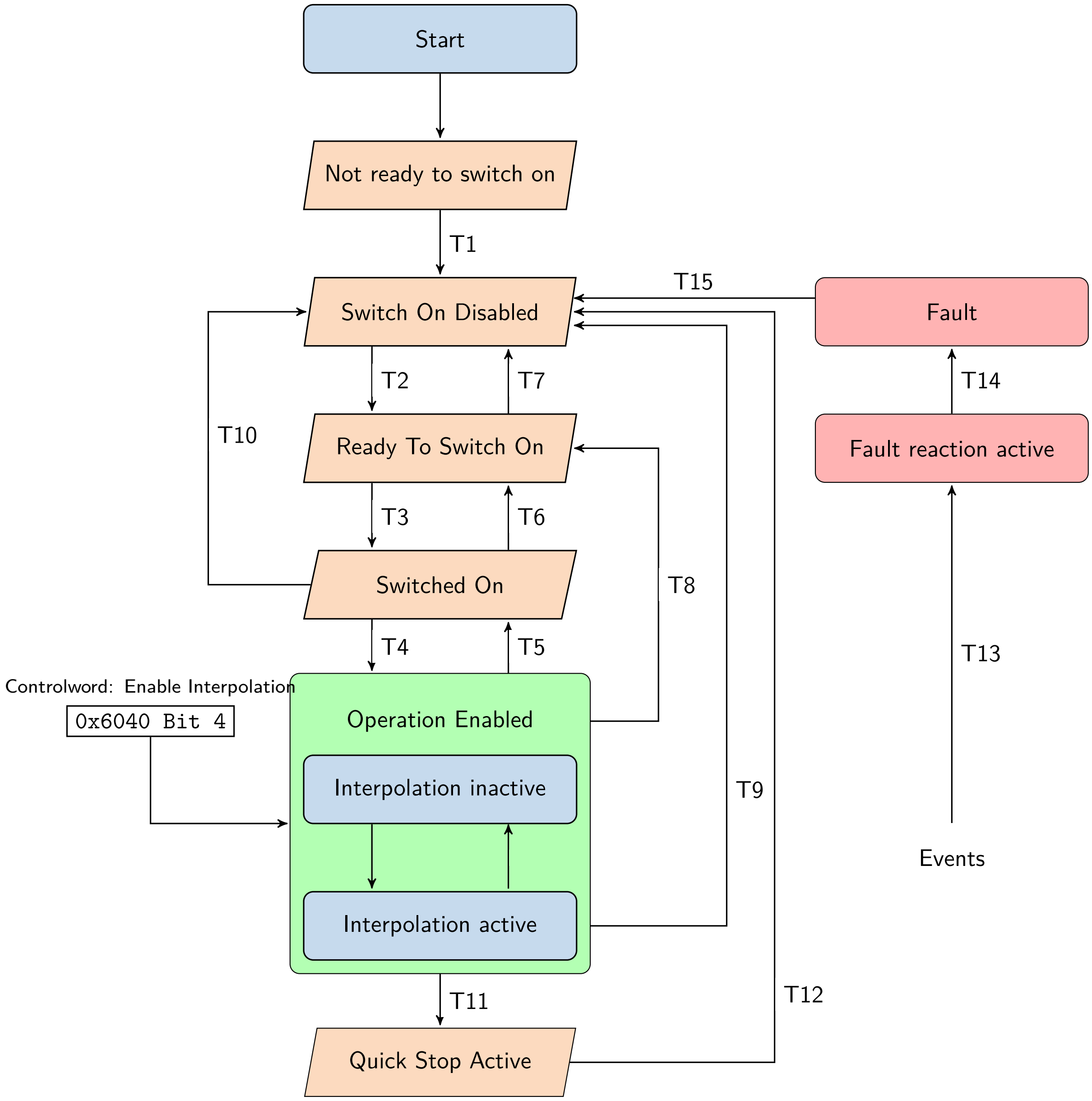


Figure 2.33: State machine of Interpolated position mode

6.1.3 State and transition of Interpolated position mode

Each state supports different functions described below:

State	Function
Interpolation inactive	The axis doesn't move. Input data is accepted. Buffer cleared on transition.
Interpolation active	The axis moves. Input data is accepted.

Figure 2.34: Supported functions of state

The change of state is carried out by bit 4 of the object 0x6040 *a1_Controlword*

6.1.4 Linear interpolation

The interpolated position mode allows to calculate the intermediate position points between two positions set points. At each period defined by object 0x60C2 a1_ip_Time_period, the mode go to next set point. The mode calculates the appropriate velocity to reach the next position.

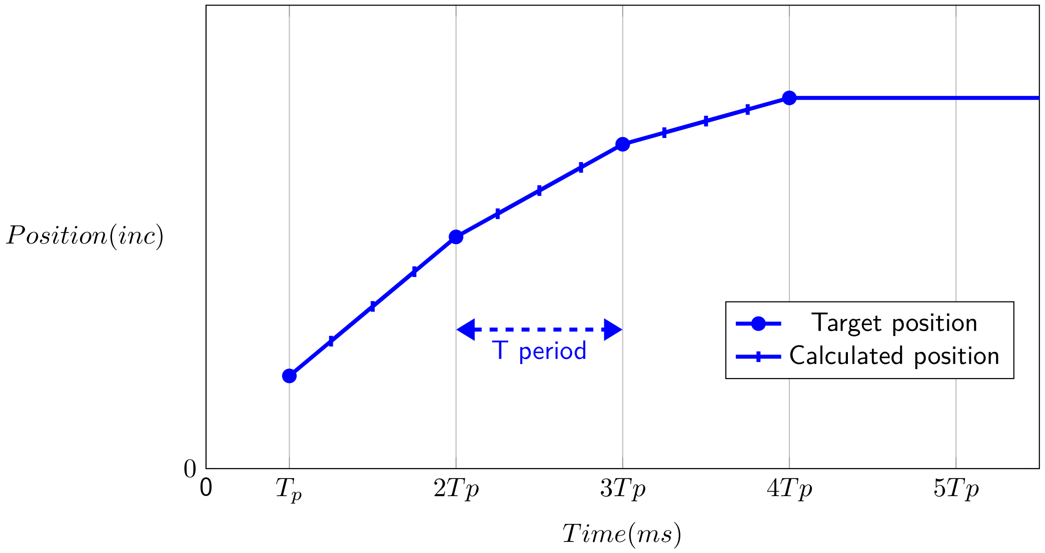


Figure 2.35: Linear interpolation

The axis continues its movement as long as there are position set points in the buffer.

6.1.5 Buffer

The buffer is organized as FIFO (First In, First Out). When the device receives a new position, it places it at the end of the queue and at each period, the device takes the set point at the top of the queue. The buffer size is 32 items. When the buffer is empty, the device keeps the last set point. The object 0x60C4 a1_ip_Data_configuration allows to have information on the buffer. When the buffer is full, the new positions are rejected.

6.1.6 Mode activation

The 0x6060 a1_Modes_of_operation object enables speed mode by setting the value to 7.

6.1.7 Specific usage

Halt command of controlword The Halt command of 0x6040 *a1_Controlword* allows the axis to be stopped at any time when bit 8 is set to 1. When bit 8 is to 0, the normal cycle resumes by continuing the execution of the ramp.

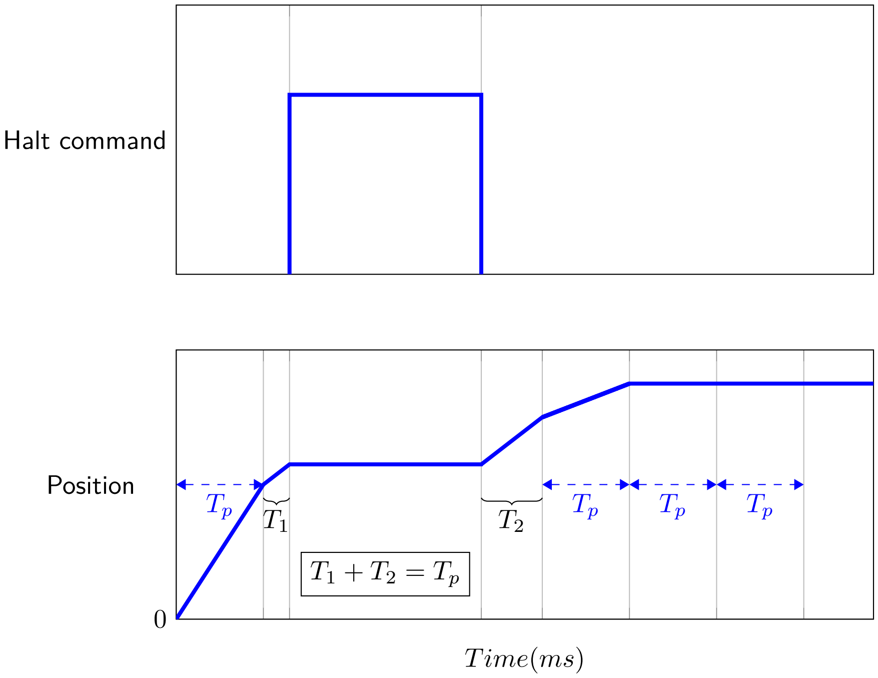


Figure 2.36: Halt Command Operation Graph

6.2 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

6.2.1 Input objects

Name	Description
0x6060 a1_Modes_of_operation	Modes of operation
0x6040 *a1_Controlword*	Control
0x60C1 a1_ip_Data_record	Requested target position
0x60C2 a1_ip_Time_period	Configuration interpolation cycle time
0x60C4 a1_ip_Data_configuration	Configuration buffer
0x60C0 a1_ip_Sub_mode_select	Type of interpolation
0x607B a1_pc_Position_range_limit	Maximal and minimal position range limits
0x607D a1_pc_Software_position_limit	Maximal and minimal software position limits
0x607F a1_pc_Max_profile_velocity	Maximal speed limit
0x6080 a1_pc_Max_motor_speed	Maximal speed limit for the motor
0x6081 a1_pc_Profile_velocity	Maximal cruising speed
0x6083 a1_pc_Profile_acceleration	Maximal cruising acceleration
0x6084 a1_pc_Profile_deceleration	Maximal cruising deceleration
0x60C5 a1_pc_Max_acceleration	Maximal acceleration
0x60C6 a1_pc_Max_deceleration	Maximal deceleration

Table 2.35: Configuration objects

6.2.2 Output objects

Name	Description
0x6041 *a1_Statusword*	Status
0x6062 a1_pc_Position_demand_value	Position generated by the generator trajectory

Table 2.36: Status objects

6.2.3 0x6040 *a1_Controlword*

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	0	-	-

The bit 4 and bit 8 of 0x6040 a1_Controlword are used for Interpolated Position mode.

15 9	8	7 5	4	3 0
-	H	-	EI	-

Name	Bit	Value	Definition
EI	4	0	Disable interpolation
		1	Enable interpolation
H	8	0	Execute instruction of bit 4
		1	Axis stopped accordingly to halt option code (see 0x605D a1_Halt_option)

Table 2.37: Definition specific bits of Controlword for Interpolated Position mode

6.2.4 0x6041 *a1_Statusword*

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

The bit 11, bit 10, bit 12 and bit 13 of 0x6041 a1_Statusword are used for Interpolated Position mode.

15	14	13	12	11	10	9	0
-		FE	IMA	ILA	TR	-	

Name	Bit	Value	Definition
TR	10	0	Target position not (yet) reached
		1	Target position reached
ILA	11	0	No internal limit
		1	Internal limit is active
IMA	12	0	Interpolation inactive
		1	Interpolation active
FE	13	0	No following error
		1	Following error

Table 2.38: Definition specific bits of Statusword for Interpolated Position mode

6.2.5 0x60C0 a1_ip_Sub_mode_select

Data type	Acces	Default	Range	Unit
INT16	RW,RPDO	0	[-32768 - 0]	-

This object define the interpolation mode. The device only provides a linear interpolation algorithm.

Value	Definition
0	Linear interpolation

Figure 2.37: Option of Interpolation sub mode

6.2.6 0x60C1 a1_ip_Data_record

This object adds a new set point position to the interpolation algorithm. Only one interpolation data record can be sent to the device at a time. On reception, the interpolation data record is placed in a FIFO buffer (subsubsection 6.1.5), defining a new position set point.

As a reminder: The buffer can contain a maximum of 32 interpolation data record.

Warning: The position delta must not exceed 0x7FFFFFFF.

6.2.7 0x60C2 a1_ip_Time_period

Index	Name	Object type
0x60C2	a1_ip_Time_period	RECORD
Subindex	**Name**	**Data type**
1	Time_units	UINT8
2	Time_index	INT8

This object defines interpolation cycle time or period in scientific notation in seconds. The 0x60C2.1 *Time_units* indicates the coefficient and the 0x60C2.2 *Time_index* indicates the exponent.

So the cycle time is given by:

$$Interpolation\,cycle\,time = unit * 10^{index} \tag{2.5}$$

0x60C2.1 *Time_units*

Data type	Acces	Default	Range	Unit
UINT8	RW,RPDO	4	[0x00 - 0xFF]	-

0x60C2.2 ***Time_index***

Data type	Acces	Default	Range	Unit
INT8	RW,RPDO	-3	[-3 - 63]	-

Note: The change of duration or cycle period will be taken into account in the next transition from Interpolation inactive state to Interpolation active state.

6.2.8 0x60C4 a1_ip_Data_configuration

Index	Name	Object type
0x60C4	a1_ip_Data_configuration	RECORD

Subindex	Name	Data type
1	Maximum_buffer_size	UINT32
2	Actual_buffer_size	UINT32
3	Buffer_organization	UINT8
4	Buffer_position	UINT16
5	Size_of_data_record	UINT8
6	Buffer_clear	UINT8

This object provides the maximum buffer size, the actual buffer length and a clear buffer command.

0x60C4.1 ***Maximum_buffer_size***

Data type	Acces	Default	Range	Unit
UINT32	RO	32	[0x00000000 - 0xFFFFFFFF]	-

This object gives the maximum number of interpolated positions the buffer can store.

0x60C4.2 ***Actual_buffer_size***

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	0	[0x00000000 - 0xFFFFFFFF]	-

This object gives the current number of interpolated positions in the buffer.

0x60C4.6 ***Buffer_clear***

Data type	Acces	Default	Range	Unit
UINT8	WO,RPDO	0	[0x00 - 0x01]	-

This object allows to clear the buffer and delete all recorded positions. This object reacts to a write event.

Value	Definition
0	Clear buffer
!= 0	No clear

Table 2.39: Buffer clear options

7 Continuous Position (CP)

This mode allows positional movement by applying a target position to the generator trajectory. It generates a position request for the position control loop.

The generator of trajectory takes into account the limit values of speed, acceleration, deceleration during the movement.

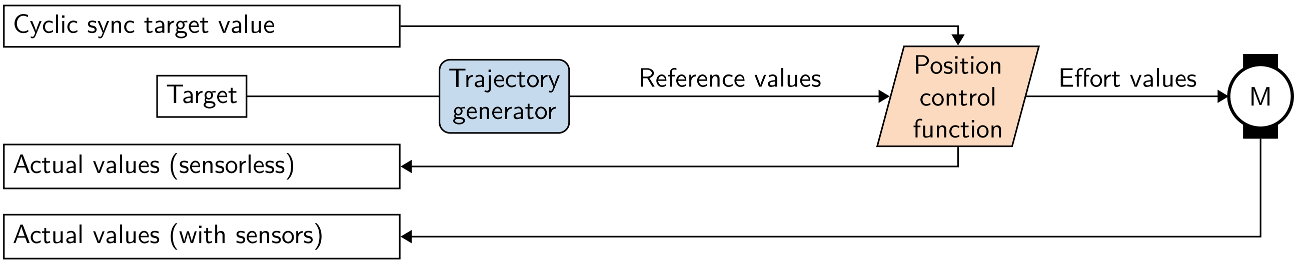


Figure 2.38: Continuous Position mode

7.1 Functional description

The setting of set points is controlled by 0x6040 *a1_Controlword*: *Absolute or relative* value as well as status of 0x6041 *a1_Statusword* : *Set-point acknowledge.*
This profile position mode works with a single set point, so only one set point is expected by the device.

7.1.1 Continuous Position

The general structure of this mode is presented in the following diagram with all the objects involved for trajectory generator.

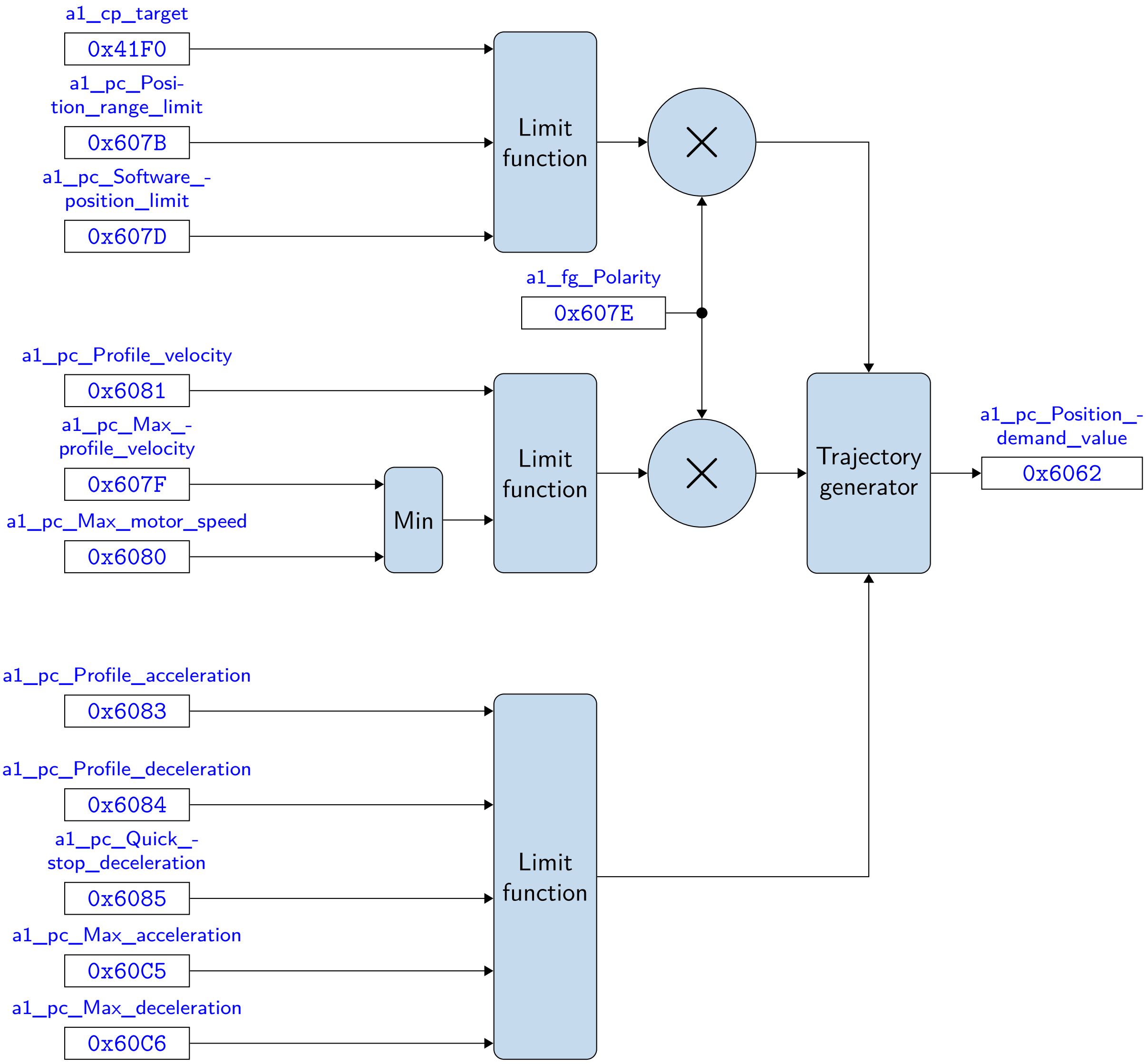


Figure 2.39: Functional diagram of Continuous Position

7.1.2 Processing of a new set point

- The new set point will be processed immediately and set the *Set-point acknowledgment bit* in the 0x6041 *a1_Statusword*

  A new 0x6081 a1_pc_Profile_velocity, 0x6083 a1_pc_Profile_acceleration and 0x6084 a1_pc_Profile_deceleration as well as control their respective limits are taken into account when executing the new set-point.

- *Absolute or relative* bit of 0x6040 a1_Controlword define if the *New set-point bit* is:
  - absolute value, position reached = 0x607A a1_pp_Position_target
  - relative value, position reached = 0x607A a1_pp_Position_target + 0x6062 a1_pc_Position_demand_value.

7.1.3 Trajectory generator

The trajectory generator uses the motion profile: Linear ramp (trapezoidal profile). Below, motion profiles of acceleration, velocity and position:

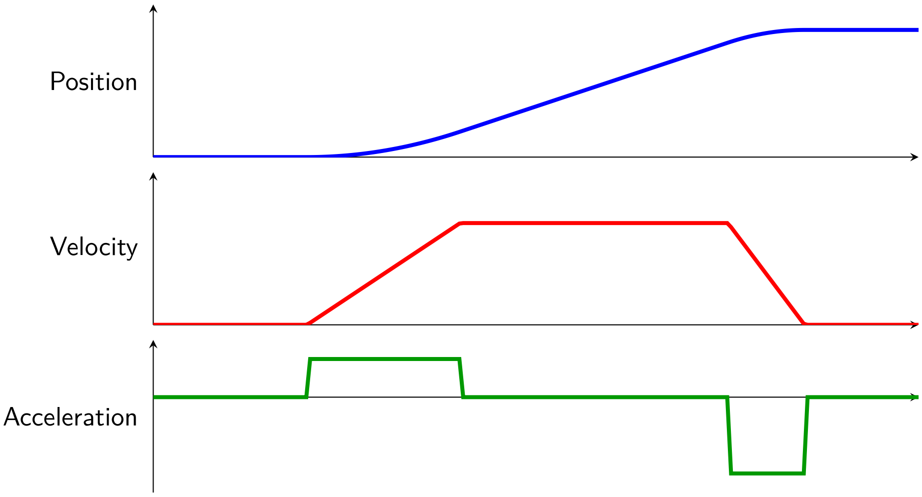


Figure 2.40: Linear ramp (trapezoidal profile)

7.1.4 Mode activation

The 0x6060 a1_Modes_of_operation object enables speed mode by setting the value to **-16**.

7.1.5 Specific usage

Halt bit of Controlword The stop bit is used to stop movement by decelerating the axis, so that the speed approaches zero. The behavior of halt is in agreement to 0x605D a1_Halt_option.

Inter state transitions The behavior when transitioning from the Operation Enabled state to another state is to stop the movement by decelerating the axis, so that the speed approaches zero. In agreement with the Transition options.

Inter mode transitions These transitions are possible from the Operation Enabled state. The behavior consists in stopping the movement by decelerating the axis, so that the speed approaches zero. The behavior is in agreement to 0x605D a1_Halt_option. Once the velocity is equal to zero, the mode change is realized.

7.2 Definition of parameters

All the objects described below correspond to axis 1, see Multi-axis for other axis.

7.2.1 Input objects

Name	Description
0x6060 a1_Modes_of_operation	Modes of operation
0x6040 *a1_Controlword*	Control
0x41F0 a1_cp_target	Requested target position
0x607B a1_pc_Position_range_limit	Maximal and minimal position range limits
0x607D a1_pc_Software_position_limit	Maximal and minimal software position limits
0x607F a1_pc_Max_profile_velocity	Maximal speed limit
0x6080 a1_pc_Max_motor_speed	Maximal speed limit for the motor
0x6081 a1_pc_Profile_velocity	Maximal cruising speed
0x6083 a1_pc_Profile_acceleration	Maximal cruising acceleration
0x6084 a1_pc_Profile_deceleration	Maximal cruising deceleration
0x60C5 a1_pc_Max_acceleration	Maximal acceleration
0x60C6 a1_pc_Max_deceleration	Maximal deceleration

Table 2.40: Configuration objects

7.2.2 Output objects

Name	Description
0x6041 *a1_Statusword*	Status
0x6062 a1_pc_Position_demand_value	Position generated by the generator trajectory
0x6064 a1_pc_Position_actual_value	Actual position value

Table 2.41: Status objects

7.2.3 0x6040 ***a1_Controlword***

Data type	Acces	Default	Range	Unit
UINT16	RW,RPDO	0	-	-

The bit 4, bit 5, bit 6, and bit 8 of 0x6040 a1_Controlword are used in Continuous Position.

15 9	8	7	6	5 0
-	H	-	abs/rel	-

Name	Bit	Value	Definition
abs/rel	6	0	Target position is in absolute value
		1	Target position is in relative value
H	8	0	No command
		1	Axis stopped accordingly to halt option code (see 0x605D a1_Halt_option)

Figure 2.41: Definition of specific bits of Controlword for Continuous Position

7.2.4 0x6041 ***a1_Statusword***

Data type	Acces	Default	Range	Unit
UINT16	RO,TPDO	-	-	-

The bit 10, 12, 13 of 0x6041 a1_Statusword is used in Continuous Position.

15 14	13	12	11	10	9 0
-	FO	SPA	-	TA	-

Name	Bit	Value	Definition
TA	10	0	Halt = 0 : Target position not reached Halt = 1 : Axis decelerates
		1	Halt = 0 : Target position reached Halt = 1 : Velocity of axis is 0
SPA	12	0	Target reached
		1	Target in process
FO	13	0	No following error
		1	Following error

Figure 2.42: Definition of specific bits of Statusword for Continuous Position

7.2.5 `0x41F0` a1_cp_target

Data type	Acces	Default	Range	Unit
INT32	RW,RPDO	0	-	-

Target position requested from the position generator to which it should go using the current settings of motion control parameters. This value is given in user-defined position units.

7.2.6 `0x6062` a1_pc_Position_demand_value

Data type	Acces	Default	Range	Unit
INT32	RO,TPDO	-	-	-

Position generated by the generator trajectory and used at the position control input. This value is given in user-defined position units.

7.2.7 `0x6064` a1_pc_Position_actual_value

Data type	Acces	Default	Range	Unit
INT32	RO,TPDO	-	-	-

Actual position value of measuring device. This value is given in user-defined position units.

7.2.8 `0x607B` a1_pc_Position_range_limit

Index	Name		Object type
0x607B	a1_pc_Position_range_limit		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT32	RW,RPDO	-	-

Subindex	Name
1	Min
2	Max

Index	Name		Object type
0x607B	a1_pc_Position_range_limit		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT32	RW,RPDO	-	-

Subindex	Name
1	Min
2	Max

Defines maximal and minimal position range limits, allowing the numerical range of the target value to be limited. The value automatically wraps to the other end of the range when the target value reaches or exceeds these limits. To prevent overflow of the input value, set the software position limits. To disable the position range limits, the min and max must be set to 0. These values are given in user-defined position units.

7.2.9 0x607D a1_pc_Software_position_limit

Index	Name		Object type
0x607D	a1_pc_Software_position_limit		ARRAY
Data type	**Acces**	**Range**	**Unit**
INT32	RW,RPDO	-	-

Subindex	Name
1	Min
2	Max

Defines maximal and minimal software position limits, defines the absolute position limits for the position demand value and the position actual value. To disable the software position limits, the min and max be set to 0. These values are given in user-defined position units

7.2.10 0x607F a1_pc_Max_profile_velocity

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	16384	-	-

Defines the maximum speed limit. This value is given in user-defined units.

7.2.11 0x6080 a1_pc_Max_motor_speed

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	16384	-	-

Defines the maximum speed limit for the motor. This value is given in user-defined units.

7.2.12 0x6081 a1_pc_Profile_velocity

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	16384	-	-

Defines the maximum cruising speed. This value is given in user-defined units.

7.2.13 0x6083 a1_pc_Profile_acceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	-	-

Defines the maximum cruising acceleration. This value is given in user-defined units.

7.2.14 0x6084 a1_pc_Profile_deceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	-	-

Defines the maximum cruising deceleration. This value is given in user-defined units.

7.2.15 0x6085 a1_pc_Quick_stop_deceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	1	-	-

Defines the maximum cruising deceleration when Quick Stop active. This value is given in user-defined units.

7.2.16 0x6086 a1_pp_Motion_profile_type

Data type	Acces	Default	Range	Unit
INT16	RW,RPDO	0	-	-

Defines the type of ramp of the motion profile used.

Value	Description
0	Trapezoidal profile

Figure 2.43: Option of 0x6086 *a1_pp_Motion_profile_type*

7.2.17 0x60C5 a1_pc_Max_acceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	100000	-	-

Defines the maximum acceleration. This value is given in user-defined units.

7.2.18 0x60C6 a1_pc_Max_deceleration

Data type	Acces	Default	Range	Unit
UINT32	RW,RPDO	100000	-	-

Defines the maximum deceleration. This value is given in user-defined units.

7.2.19 0x607E a1_fg_Polarity

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

7.2.20 0x607C a1_hm_Home_offset

Data type	Acces	Default	Range	Unit
INT32	RW,RPDO	0	-	-

Chapter 3

CANOpen protocol, DS301

1 Introduction

CANOpen is based on bus CAN (Controller Area Network) which is ISO-11898 standardized. CANOpen protocol is standardized by the CAN In Automation (CIA) under the name of CiA301.

1.1 CAN for CANOpen

CANOpen uses standard CAN frames to communicate on the bus. The frames are identified with an Id : CAN-ID. It is coded on 11 bits. The CAN-ID with the lowest value has priority if multiple frames are sent at the same time from different devices.

The simplified CANOpen frame:

11 bit	0	1	2	3	4	5	6	7
CAN-ID	Data							

Table 3.1: CAN frame

Note: A CAN message can contain a maximum of 8 bytes of useful data.

Note: The data is always sent on the bus in **little-endian format**.

1.2 CANOpen node

Each device, also called Node, is identified by a unique identifier in the network called: Node-Id. The Node-Id can take a value in the range [1-127].

All descriptions and functionalities of a Node are described in the EDS (Electronic Data Sheet) file.

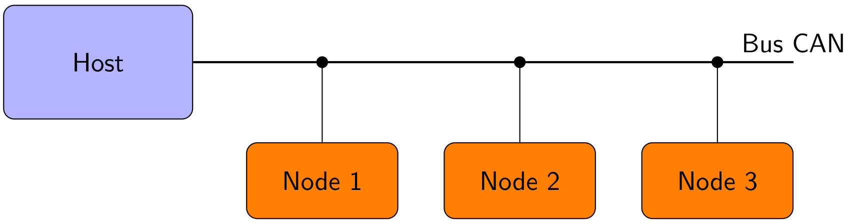


Table 3.2: Bus CAN

1.3 Communication

CANOpen protocol defines three communication protocol sequences :

- Master/Slave protocol
- Client/Server protocol

- Producer/Consumer protocol

1.3.1 Master / slave protocol

This protocol works on a network where only one CANOpen master is present for a specific functionality. And therefore the other CANOpen devices are slaves. The master sends a request and the slave responds.

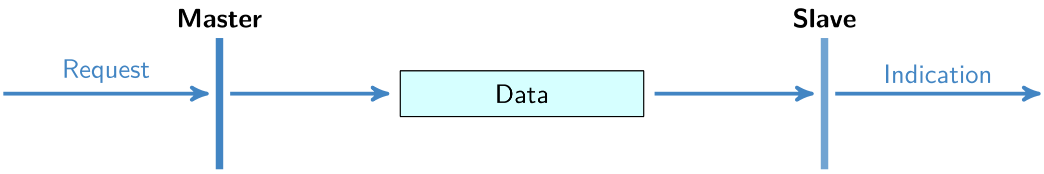


Figure 3.1: Unconfirmed master / slave protocol

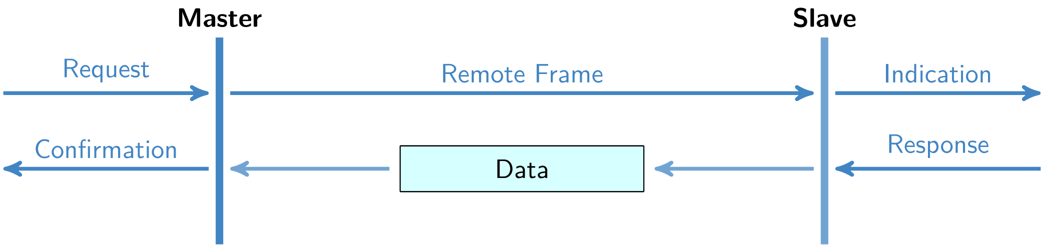


Figure 3.2: Confirmed master / slave protocol

1.3.2 Client / server protocol

This protocol is used between a client and a server. When the client makes a request (download / upload), the server triggers the processing of the request. The server responds to the request when the task is completed.

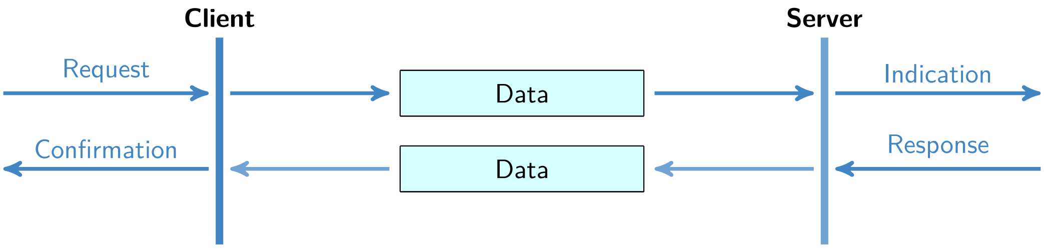


Figure 3.3: Client / server protocol

1.3.3 Producer / consumer protocol

This protocol works with a producer that sends a message that can be received by one or more devices on the network. The producer does not receive confirmation.

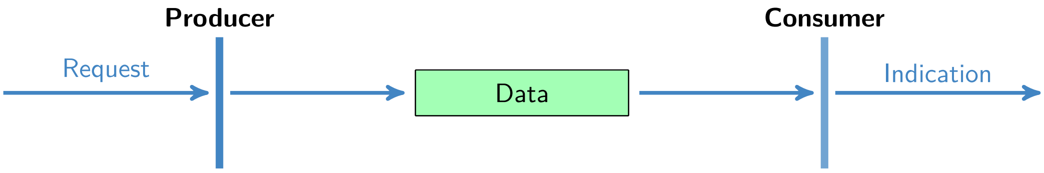


Figure 3.4: Push producer / consumer protocol

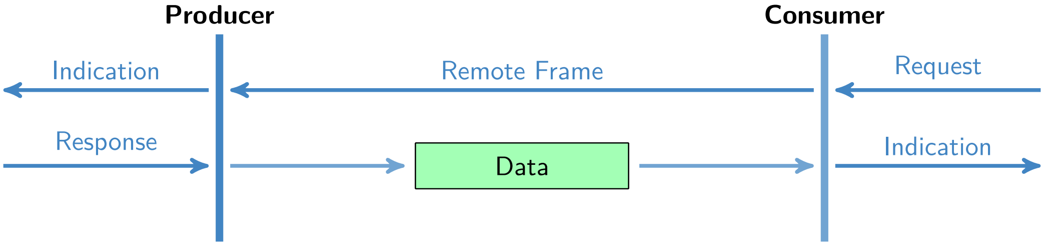


Figure 3.5: Pull producer / consumer protocol

2 CANOpen Services

Services provided by the CANOpen stack allow a standard communication between the devices on the network. These services allow communication object exchanges. There are 5 types of services:

- Network management services
  - Network management services (NMT)
  - Node guarding, network equipment monitoring
  - Heartbeat, network equipment monitoring
  - Boot-up
- Service Data Object (SDO), providing read and write access to the dictionary of objects
- Process Data Object (PDO), allowing to transmit data in real time process:
  - TPDO Transmit-PDO for use in data transmission
  - RPDO Receive-PDO for use in data reception
- SYNC, synchronization object used by PDO.
- Emergency (EMCY), emergency

Each type of message is defined by the allocation of the following CAN-IDs:

Services	Default CAN-ID	CAN-ID configurable
NMT	0x000	
SYNC	0x080	yes
EMCY	0x080 + Node-Id	yes
TPDO1	0x180 + Node-Id	yes
RPDO1	0x200 + Node-Id	yes
TPDO2	0x280 + Node-Id	yes
RPDO2	0x300 + Node-Id	yes
TPDO3	0x380 + Node-Id	yes
RPDO3	0x400 + Node-Id	yes
TPDO4	0x480 + Node-Id	yes
RPDO4	0x500 + Node-Id	yes
TSDO	0x580 + Node-Id	
RSDO	0x600 + Node-Id	
Boot-Up	0x700 + Node-Id	
Node-guarding and Heartbeat	0x700 + Node-Id	

Table 3.3: Default CAN-ID services

Some services can have a CAN-ID configurable by a communication object:

Services	Object
SYNC	0x1005
EMCY	0x1014
TPDOX	0x180X
RPDOX	0x140X

Table 3.4: Index of CAN-ID services

But the changing CAN-ID should not interfere with the following reserved CAN-IDs:

CAN-ID COB-ID	Description
0x000	NMT
0x000 0x07F	reserved
0x101 0x180	reserved
0x581 0x5FF	default SDO (tx)
0x601 0x67F	default SDO (rx)
0x6E0 0x6FF	reserved
0x701 0x77F	NMT Error Control
0x780 0x7FF	reserved

Table 3.5: Restricted CAN-ID

3 Object Dictionary (OD)

The object dictionary is a collection of all the data items which have an influence on the behavior of the application objects, the communication objects and the state machine used on the device. Each device on the network has its own object dictionary.

The object dictionary is divided into several areas:

Index range	Description
0x0000	Reserved
0x0001 to 0x025F	Data types
0x0260 to 0x0FFF	Reserved
0x1000 to 0x1FFF	Communication profile area
0x2000 to 0x5FFF	Manufacturer-specific profile area
0x6000 to 0x9FFF	Standardized profile area

Table 3.6: Object dictionary area

The **Communication profile area** contain the communication specific parameters. These objects are common to all CANOpen devices.

The **Standardized profile area** contain all data objects common to a profiles of CANOpen devices that may be read or written via the network. The objects from 6000 h to 9FFF h describe parameters and functionality.

The **Manufacturer-specific** profile area contains the objects for specific UniSwarm features.

3.1 Description of the object dictionary

The objects of the dictionary are described by several parameters. This description is materialized by an EDS file: Electronic Data Sheet. ASCII format respecting a strict syntax that can be used by the bus configuration software.

3.1.1 Index and sub-index

These form the unique identifier of an object in the objects dictionary in hexadecimal notation.

3.1.2 Object code

The object code denotes what kind of object is at a particular index within the objects dictionary. They can be one of the following:

Object name	Description	Code
NULL	An object with no data fields	0x00
DOMAIN	A large variable amount of data	0x02
DEFTYPE	A type definition for simple data type such as a Boolean, Unsigned16	0x05
DEFSTRUCT	Defines a new record type	0x06
VAR	A single value	0x07
ARRAY	A data area in which each entry is of the same data type.	0x08
RECORD	A data area that contains entries that are a combination of simple data types.	0x09

Table 3.7: Object code

3.1.3 Data type

The data type information indicates the data type of the object.

Type	Value	Size in byte
Boolean	0x0001	1
Integer8	0x0002	1
Integer16	0x0003	2
Integer24	0x0010	3
Integer32	0x0004	4
Integer40	0x0012	5
Integer48	0x0013	6
Integer56	0x0014	7
Integer64	0x0015	8
Unsigned8	0x0005	1
Unsigned16	0x0006	2
Unsigned32	0x0007	4
Unsigned24	0x0016	3
Unsigned40	0x0018	5
Unsigned48	0x0019	6
Unsigned56	0x001A	7
Unsigned64	0x001B	8
Real32	0x0008	4
Real64	0x0011	8
VISIBLE STRING	0x0009	Variable
OCTET STRING	0x000A	Variable
UNICODE STRING	0x000B	Variable
TIME OF DAY	0x000C	Variable
TIME DIFFERENCE	0x000D	Variable
Domain	0x000F	Variable

Table 3.8: Data types

Note: Data type with indices from 0x0001 to 0x0007, 0x0010, from 0x0012 to 0x0016 , and from 0x0018 to 0x001B may be mapped in order to define the appropriate space in the RPDO.

Note: Data type with indices from 0x0008 to 0x000F, 0x0011, from 0x0020 to 0x0023 shall not be mapped into RPDOs

3.1.4 Access usage

Access object:

- rw: read and write access
- wo: write only access
- ro: read only access
- const: read only access, value is constant

In addition, there are the access attributes for the PDOs:

- rww: read and write access and can be mapped on RPDO
- rwr: read and write access and can be mapped on TPDO

4 Network management services (NMT)

Network management (NMT) follows a master-slave structure. All devices are NMT slaves but the network must have a device master (device master, computer or other).

The service provides a tool to initiate, start, monitor, reset or stop the devices. Monitoring is made with the Node guarding and Heartbeat functionalities.

4.1 NMT Network management

The NMT master controls the state of each NMT slave. The state can be chosen among the following ones: Stopped, Pre-operational, Started, Initialization.

4.1.1 NMT state machine

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

- state **Initialization**
  - **Initializing**: the device enters this state after a power-on or an hardware/software reset.
  - **Application reset**: The object dictionary in Manufacturer-specific profile area, index range `0x2000` to `0xFFFF` is reset.
  - **Communication reset**: The object dictionary in Communication profile area, index range `0x1000` to `0x1FFF` is reset.
- state **Pre-operational**
- state **Started**
- state **Stopped**

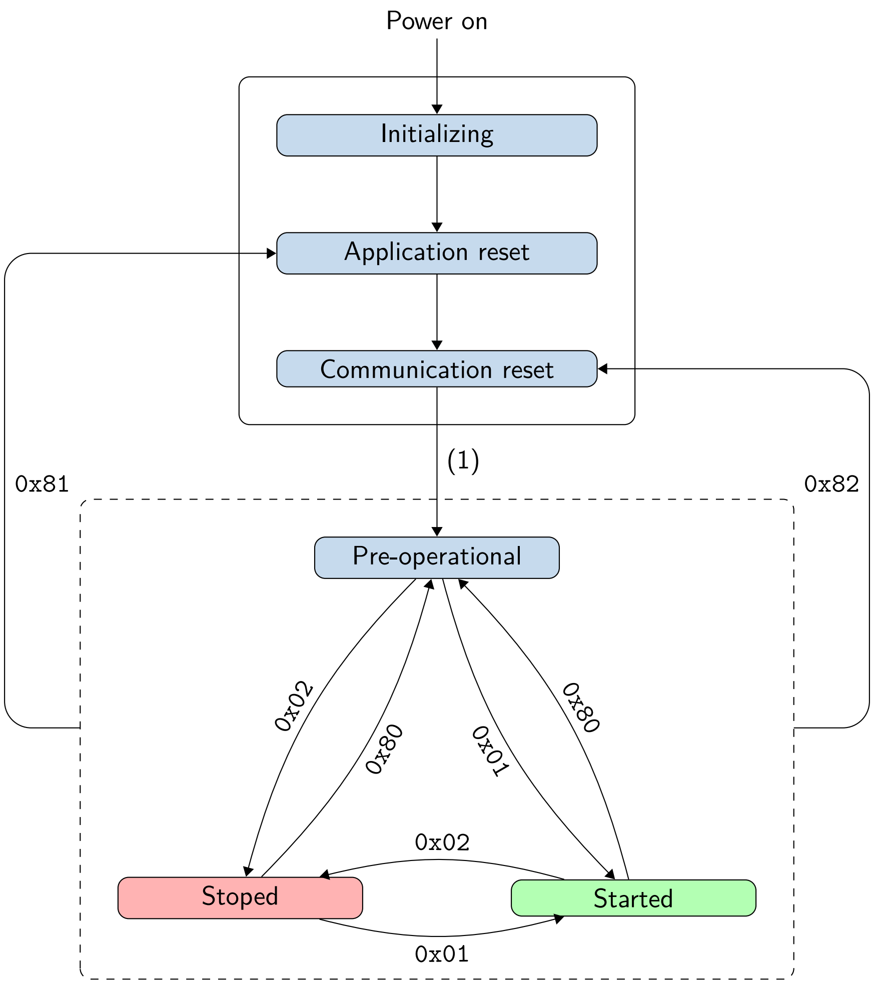
The following figure show the state diagram:

CAN-ID	Byte[0]	Byte[1]
0x000	Mode	Node-Id

Figure 3.7: NMT frame

Mode value	Description
0x01	Start node
0x02	Stop node
0x80	Enter in Pre-operational mode
0x81	Application reset
0x82	Communications reset

Figure 3.8: MODE values in NMT frame



(1) Automatic switch to Pre-operational and send Boot-up message

Figure 3.6: Different status of a CAN Open node

4.1.2 NMT frame

The NMT frame allow to change the state of device. An NMT frame have the CAN-ID 0x000 and a payload of two bytes. The first one is the mode and the second one the node id.

4.1.3 NMT States and Services

Authorized services according to the NMT state:

	Pre-operational	Started	Stopped
PDO		X	
SDO	X	X	
SYNC	X	X	
EMCY	X	X	
Node guarding	X	X	X
Heartbeat	X	X	X

Figure 3.9: NMT states and authorized services

4.2 Node Guarding - Heartbeat

Two services are available to detect an error on the CAN network: the **Node guarding** service and the **Heartbeat** service.

- **Node guarding**: the master sends a message periodically and each slave has to respond within a time limit.
- **Heartbeat**: each slave sends a message with his state without prior request from the master.

Note: The Heartbeat service has priority over the Node guarding service. Activation of the Heartbeat service results in the deactivation of the Node Guarding service.

4.2.1 Node guarding

This service monitors the status of devices on the bus and makes it possible to detect remote errors on the network.
The master and the slave monitor each other: the master cyclically requests the NMT status of the slave. In each response from the slave, the Toggle-bit (bit 7) is toggled.

Two monitoring functions are implemented:

- **Node guarding**: The master can react accordingly if these responses are not sent or if the slave always responds with the same bit Toggle.
- **Life guarding**: The slave monitors the reception of messages from the master, if the message is not sent within the allotted time, the Life Time, the slave triggers an Emergency (EMCY) message (with ode 0x8130) see table **??**.
  The Life time : *Life Time* = *0x100C Guard_time* x *0x100D Life_time_factor*.

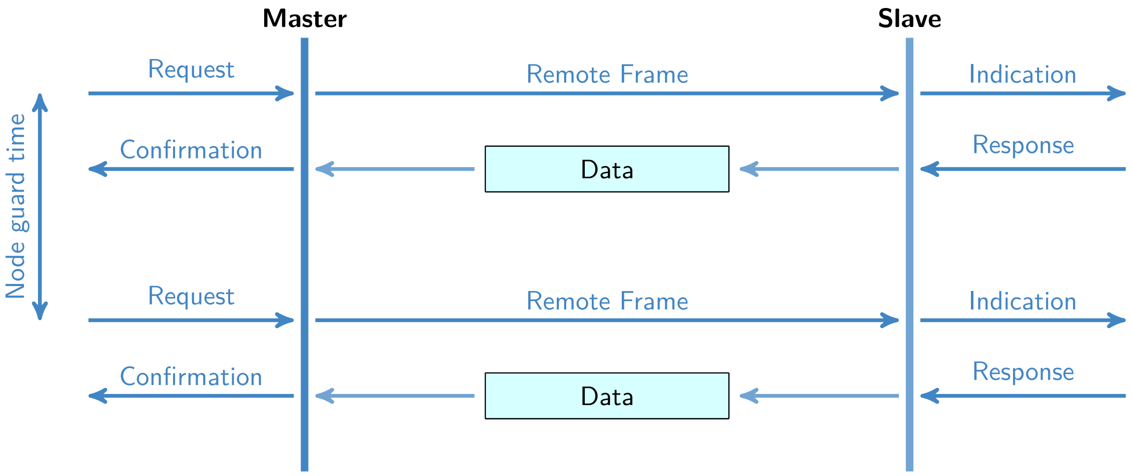
Note: This service is activated by setting value in object 0x100C Guard_time and 0x100D Life_time_factor other than zero.

Node guarding frame The node guarding service follows a master / slave architecture: the NMT master sends an RTR (Remote Transmission Request) message with CAN-ID 700 + node Id to a slave and the slave responds with an 8-bit message.

The response message is built with a Toggle-bit (bit 7) and the current NMT state of the slave in bits 6 to 0. For the first response and after a NMT reset, the toggle bit should be 0.

CAN-ID	Byte[0]
0x700 + Node-Id	0x00

Figure 3.10: Node guarding frame



- t: Toggle Bit
- Node state:
  - 4: Stopped
  - 5: Started
  - 27: Pre-operational

Figure 3.11: Node guarding protocol

4.2.2 Heartbeat

The Heartbeat works in Producer/Consumer mode with one producer and 0 minimum consumer.

The producer heartbeat send a heartbeat message periodically, with the time between two messages defined by the "Producer heartbeat time" object.

The consumer check if it received a message in the time defined by the object "Consumer heartbeat time".

Note: This service is activated by setting the producer heartbeat time object in object 0x1017 Producer_heartbeat_time to a value other than zero.

CAN-ID	Byte[0]
0x700 + Node-Id	0x00

Figure 3.12: Heartbeat frame

CAN-ID	Byte[0]
0x700 + Node-Id	0x00

Figure 3.14: Boot-up frame

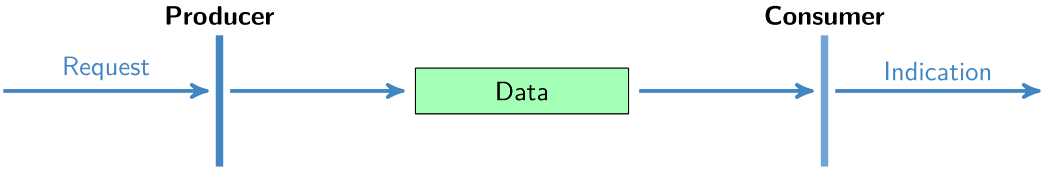
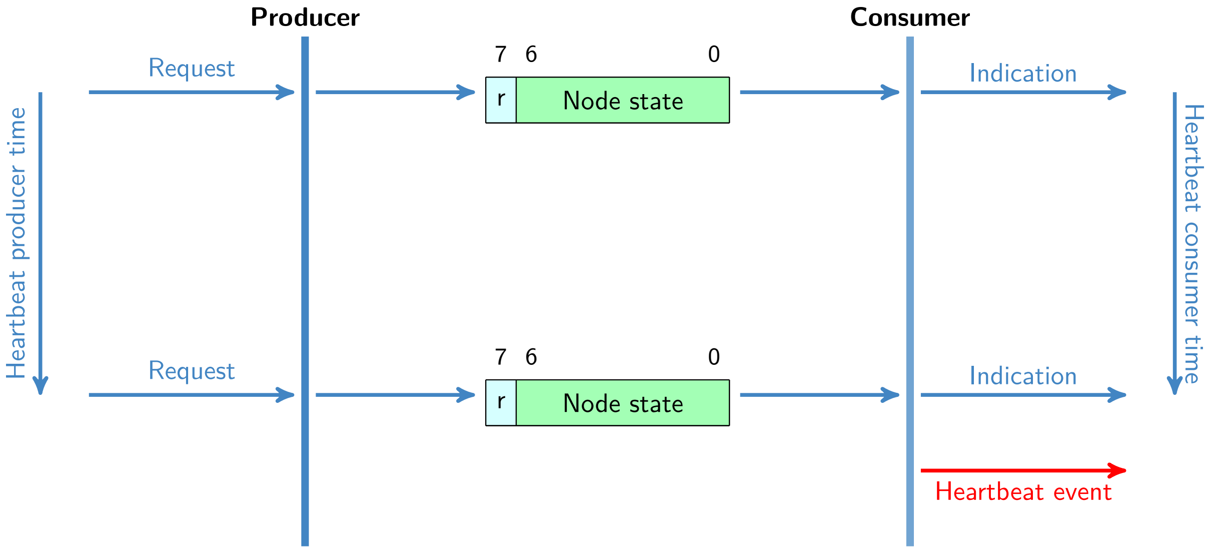


Figure 3.15: Boot-up protocol



- r: reserved (always 0)
- Node state:
  - 0: Boot-Up
  - 4: Stopped
  - 5: Operational
  - 127: Pre-operational

Figure 3.13: Heartbeat protocol

4.3 Boot-up

After power-up, the slave sends a Boot-up message to indicate that the Initializing phase is complete.

5 Emergency (EMCY)

The EMCY service is used to transmit application faults associated with each station. When a fault is detected the device send a EMCY message with Emergency error code, Error register and error code (optional).

CAN-ID	Byte[0]
0x080 + Node-Id	...

Figure 3.16: EMCY frame

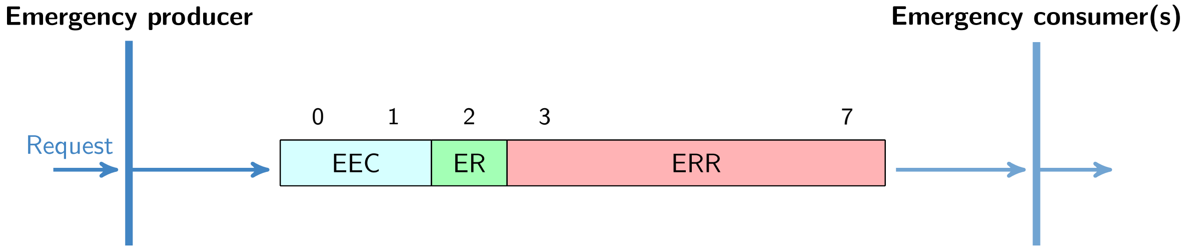


Figure 3.17: Emergency protocol with byte[0]

- EEC: Emergency error code
- ER: Error register (see object 0x1001 *Error_register*)
- ERR: Manufacturer-specific error code (Optional)

EEC	Description
0x0000	**Error reset or no error**
0x1000	**Generic error**
0x2000	**Current generic error**
0x2100	Current, CANOpen device input side generic
0x2200	Current inside the CANOpen device generic
0x2300	Current, CANOpen device output side generic
0x3000	**Voltage generic error**
0x3100	Mains voltage generic
0x3200	Voltage inside the CANOpen device generic
0x3300	Output voltage generic
0x4000	**Temperature generic error**
0x4100	Ambient temperature generic
0x4200	Device temperature generic
0x5000	**CANOpen device hardware generic error**
0x6000	**CANOpen device software generic error**
0x6100	Internal software generic
0x6200	User software generic
0x6300	Data set generic
0x7000	**Additional modules generic error**
0x8000	**Monitoring generic error**
0x8100	Communication generic
0x8110	CAN overrun (objects lost)
0x8120	CAN in error passive mode
0x8130	Life guard error or heartbeat error
0x8140	Recovered from bus off
0x8150	CAN-ID collision
0x8200	Protocol error - generic
0x8210	PDO not processed due to length error
0x8220	PDO length exceeded
0x8230	DAM MPDO not processed, destination object not available
0x8240	Unexpected SYNC data length
0x8250	RPDO timeout
0x9000	**External error generic error**
0xF000	**Additional functions generic error**
0xFF00	**Device specific generic error**

Table 3.9: General emergency error codes
Other specific EEC codes are defined in the operating modes

6 Error codes and error behavior

6.1 Definition of parameters

6.1.1 0x1001 Error_register

Data type	Acces	Default	Range	Unit
UINT8	RO,TPDO	-	-	-

Bit	Description
0	Generic error
1	Current
2	Voltage
3	Temperature
4	Communication error (overrun, error state)
5	Device profile specific
6	reserved (always 0 b)
7	manufacturer-specific

Table 3.10: Error code register of object 0x1001

6.1.2 0x1029 Error_behaviour

Index	Name		Object type
0x1029	Error_behaviour		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT8	RW	[0 - 2]	-

Subindex	Name
1	Communication_error

Sets the behavior in the event of a serious device failure in the Operational NMT state. By default, the device automatically enter the Pre-operational NMT state.

Failures include the following communication errors:

- CAN interface bus stop conditions
- NodeGuarding "time out"
- Heartbeat "time out"
- Serious errors can also be caused by internal device failures.

Bit	Description
0	Change to NMT state Pre-operational (default)
1	No change of the NMT state
2	Change to NMT state Stopped

Table 3.11: Error class values of 0x1029.1

7 Service Data Object (SDO)

This service provides access to the device object dictionary by an Index and Sub-index without time constraints in writing or reading.

Each network device is an SDO server, the one that holds the OD. The client refers to the node requesting to read or write an object value in the server's object dictionary.

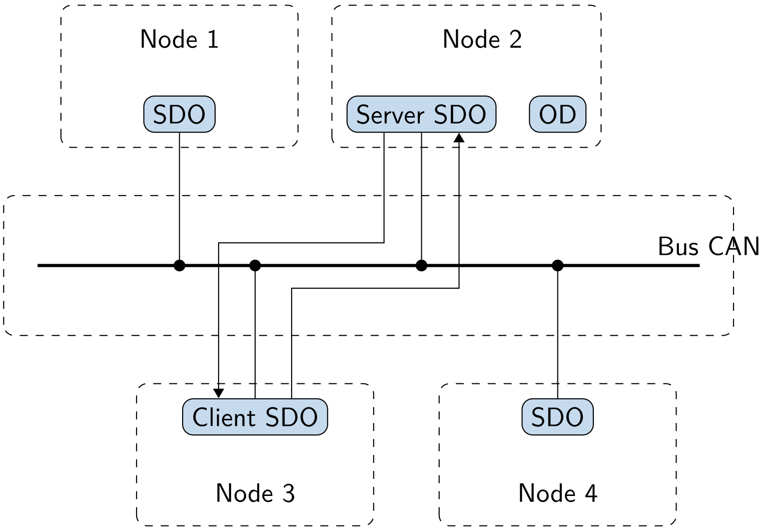


Figure 3.18: Server / client SDO

7.1 SDO message

CAN-ID	**Byte[0-7]**
CAN-ID+Node-Id	...

Figure 3.19: SDO frame

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			

- Cmd: Command
- Index: Index of object
- Subindex: Subindex of object

Figure 3.20: Expedited SDO upload frame

7.2 Expedited transfer

This mode of communication is used to write or read data in object. The object size must be inferior or equal to 4 bytes. An answer is expected after each request, either with data, with a confirmation or with an error message.

7.2.1 SDO reading

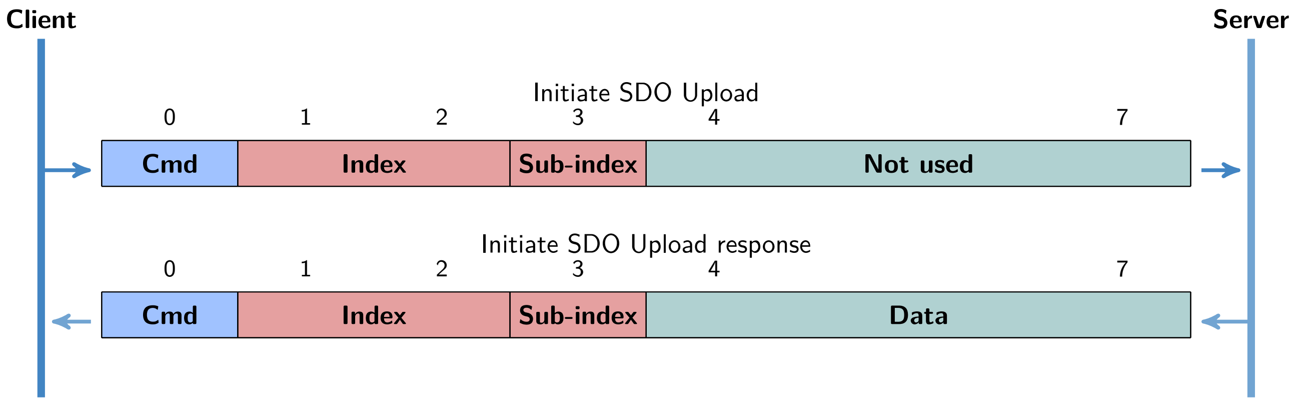


Figure 3.21: Expedited SDO Upload protocol

Cmd	Description
0x40	Upload/Reading request
0x4F	Upload/Reading response for data of size of 1 byte
0x4B	Upload/Reading response for data of size of 2 byte
0x47	Upload/Reading response for data of size of 3 byte
0x43	Upload/Reading response for data of size of 4 byte
0x2F	Download/Writing for data of size of 1 byte
0x2B	Download/Writing for data of size of 2 byte
0x27	Download/Writing for data of size of 3 byte
0x23	Download/Writing for data of size of 4 byte
0x60	Download/Writing response

Figure 3.22: List of commands (cmd)

Initiate SDO Upload/Reading request

0	1	2	3	4			7
Cmd	Index		Sub-index	Data			
0x40	lsb	msb	Sub-index	0	0	0	0

Initiate SDO Upload/Reading response

Read response for a 1 byte data:

0	1	2	3	4			7
Cmd	Index		Sub-index	Data			
0x4F	lsb	msb	Sub-index	data	0	0	0

Read response for a 2 bytes data:

0	1	2	3	4			7
Cmd	Index		Sub-index	Data			
0x4B	lsb	msb	Sub-index	lsb	msb	0	0

Read response for a 3 bytes data:

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			
0x47	lsb	msb	Sub-index	lsb	···	msb	0

Read response for a 4 bytes data:

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			
0x43	lsb	msb	Sub-index	lsb	···	···	msb

7.2.2 SDO Download/Writing

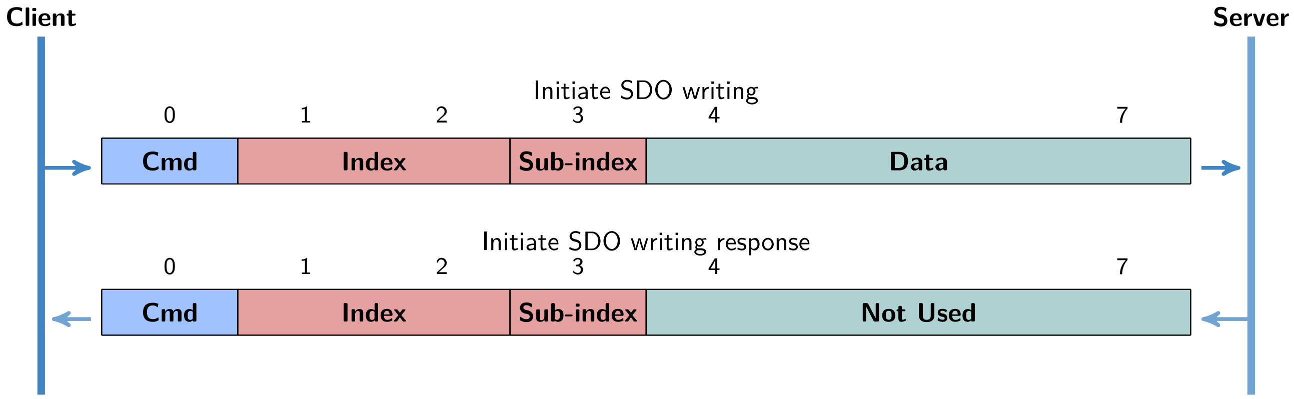


Figure 3.23: Expedited SDO Download protocol

Initiate SDO download / writing

Write request for 1 byte data:

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			
0x2F	lsb	msb	Sub-index	data	0	0	0

Write request for 2 bytes data:

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			
0x2B	lsb	msb	Sub-index	lsb	msb	0	0

Write request for 3 bytes data:

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			
0x27	lsb	msb	Sub-index	lsb	···	msb	0

Write request for 4 bytes data:

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			
0x23	lsb	msb	Sub-index	lsb	···	···	msb

Initiate SDO download / writing response

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**Data**			
0x60	lsb	msb	Sub-index	0	0	0	0

7.3 SDO abort transfer

Error response:

0	1	2	3	4			7
Cmd	**Index**		**Sub-index**	**SDO abort codes**			
0x80	lsb	msb	Sub-index	lsb	⋯	⋯	msb

7.4 SDO abort codes

Error codes	Description
0x05030000	Toggle bit not alternated
0x05040000	SDO protocol timed out
0x05040001	Client/server command specifier not valid or unknown
0x05040002	Invalid block size (block mode only)
0x05040003	Invalid sequence number (block mode only)
0x05040004	CRC error (block mode only)
0x05040005	Out of memory
0x06010000	Unsupported access to an object
0x06010001	Attempt to read a write only object
0x06010002	Attempt to write a read only object
0x06020000	Object does not exist in the object dictionary
0x06040041	Object cannot be mapped to the PDO
0x06040042	The number and length of the objects to be mapped would exceed PDO length
0x06040043	General parameter incompatibility reason
0x06040047	General internal incompatibility in the device
0x06060000	Access failed due to an hardware error
0x06070010	Data type does not match, length of service parameter does not match
0x06070012	Data type does not match, length of service parameter too high
0x06070013	Data type does not match, length of service parameter too low
0x06090011	Sub-index does not exist
0x06090030	Invalid value for parameter (download only)
0x06090031	Value of parameter written too high (download only)
0x06090032	Value of parameter written too low (download only)
0x06090036	Maximum value is less than minimum value
0x060A0023	Resource not available: SDO connection
0x08000000	General error
0x08000020	Data cannot be transferred or stored to the application
0x08000021	Data cannot be transferred or stored to the application because of local control
0x08000022	Data cannot be transferred or stored to the application because of the present device state
0x08000023	Object dictionary dynamic generation fails or no object dictionary is present
0x08000024	No data available

Table 3.12: SDO error codes

8 Process Data Object (PDO)

The purpose of "Process Data Objects (PDO)" is to provide a transfer of data in real time during the operation of controller. This service is performed without protocol overload or confirmation. The size of a PDO frame is variable and

RPDO	CAN-ID COB-ID	Object Index	
		Index RPDO communication	Index RPDO mapping
RPDO1	0x200 + Node-Id	0x1400	0x1600
RPDO2	0x300 + Node-Id	0x1401	0x1601
RPDO3	0x400 + Node-Id	0x1402	0x1602
RPDO4	0x500 + Node-Id	0x1403	0x1603

Table 3.13: Index used of RPDOs

depends on the size of object.

The PDO provides an interface to the application objects in the object dictionary. Data type and mapping of object is determined by the corresponding PDO mapping structure in the object dictionary.

The PDO configuration process (PDO Mapping) allows to configure the number of objects in a PDO. This process uses the SDO service.

There are two types of PDO, the Transmit-PDO (TPDO) for use in data transmission and the Receive-PDO (RPDO) for use in data reception. The data transmission via the PDO service operates according to a producer / consumer relationship: RPDOs are frames received from the master or others nodes. TPDOs are frames transmitted to others.

PDOs are described by the PDO communication parameter and the PDO mapping parameter.

Note: A node can have a maximum of four TPDOs and four RPDOs.

There is an index couple for each PDO: the columns "Index RPDO communication" and "Index RPDO mapping" provide the indexes of specials objects used to read or modify the parameters of communication objects via an SDO object:

TPDO	CAN-ID COB-ID	Object Index	
		Index TPDO communication	Index TPDO mapping
TPDO1	0x180 + Node-Id	0x1800	0x1A00
TPDO2	0x280 + Node-Id	0x1801	0x1A01
TPDO3	0x380 + Node-Id	0x1802	0x1A02
TPDO4	0x480 + Node-Id	0x1803	0x1A03

Table 3.14: Index used of TPDOs

8.1 PDO message

There is two ways to transmit a PDO message:

- Synchronous transmission: the object are synchronized on SYNC.
- Event-driven transmission

This PDO is only activated if the status of CANOpen is "Started". It is necessary to activate the PDOs, for that the "valid" bit of CAN-ID must be to set to 0x0.

Example:
to activate or deactivate the TPDO1, the object with the index 0x1800 and sub-index 0x1 is used:

- Deactivate TPDO1: set sub-index 1 to 0x80000181
- Activate TPDO1: set sub-index 1 to 0x00000181

The value 0x181 is the id of TPDO1.

After each SYNC, two things happen in this order:

- for TPDOs: the slaves samples and copy the data into TPDOs which are then sent on the bus.
- for RPDOs: the previous received RPDO data from master is copied in the objects database and made available to the application.

8.2 SYNC

The master in a CAN Open network sends a unique sync frame for all the nodes. At reception, the nodes transmit theirs TPDOs and apply their RPDOs. A SYNC frame has the ID 0x080 and does not contain a payload.

CAN-ID	Data
0x080	-

Figure 3.24: SYNC frame

The frequency of SYNC frame can be variable, but is always sent by the master.

8.3 PDO dynamic mapping

General procedure:

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

1. Deactivate PDO: setting the valid bit to 0x1 of sub-index 0x01 of the PDO communication parameter.
2. Disable mapping: setting sub-index 0x00 to 0.
3. Modify mapping: changing the values of corresponding sub-indexes.
4. Enable mapping: setting sub-index 0x00 to the number of mapped objects.
5. Activate PDO: setting the valid bit to 0 of sub-index 0x01 of PDO communication parameter.

8.4 PDO parameter objects

8.4.1 0x1400 *RPDO_parameter_1*
0x1401 *RPDO_parameter_2*
0x1402 *RPDO_parameter_3*
0x1403 *RPDO_parameter_4*

Index	Name	Object type
0x1400	RPDO_parameter_1	RECORD

Subindex	Name	Data type
1	COB_ID	UINT32
2	Transmission_type	UINT8
3	Inhibit_time	UINT16
5	Event_timer	UINT16

The PDO parameter describes the communication abilities of the PDO.

0x1400.1 ***COB_ID***

Data type	Acces	Default	Range	Unit
UINT32	RW	512	[0x00000080 - 0xFFFFFFFF]	-

Contains the CAN-ID (COB-ID) of RPDO

31		10	0
v		COB ID	

Table 3.15: COB ID

- **v**: The bit **valid** select which RPDOs are used in the NMT state Operational. PDOs can be fully configured but not used, and therefore set to "not valid" .

Value	Description
0x00	PDO exists / is valid
0x01	PDO does not exist / is not valid

Figure 3.25: Description of bit 31: v

- **COB ID CAN-ID** of RPDO

0x1400.2 ***Transmission_type***

Data type	Acces	Default	Range	Unit
UINT8	RW	1	-	-

The transmission of PDO depends on the configuration of Transmission Types parameters which can be:

Value	Description
0x00 to 0xF0	Synchronous
0xFE	Event

Table 3.16: Description of RPDO transmission type

- **Synchronous**: the synchronous transmission type means that the data is transmitted immediately, but it is applied when SYNC is received. The SYNC service provides a data synchronization signal over the network.
- **Event**: the Event-driven transmission type means that the PDO may be received at any time and that the data is applied immediately after reception.

0x1400.3 ***Inhibit_time***

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	0.1 ms

The value is defined as multiple of 100 ţs. The value of 0 shall disable the inhibit time. The value shall not be changed while the PDO exists.

0x1400.5 ***Event_timer***

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	ms

The value is defined as multiple of 1 ms. The value of 0 shall disable the event-timer. The RPDO use the time for deadline monitoring. The deadline monitoring is activated within the next reception of an RPDO after configuring the event-timer. A timeout results in an indication to the local application.

8.4.2 0x1600 ***RPDO_mapping_1***
0x1601 ***RPDO_mapping_2***
0x1602 ***RPDO_mapping_3***
0x1603 ***RPDO_mapping_4***

Index	Name	Object type
0x1600	RPDO_mapping_1	RECORD

Subindex	Name	Data type
0	Number_of_entries	UINT8
1	PDO_mapping_entry_1	UINT32
2	PDO_mapping_entry_2	UINT32
3	PDO_mapping_entry_3	UINT32
4	PDO_mapping_entry_4	UINT32
5	PDO_mapping_entry_5	UINT32
6	PDO_mapping_entry_6	UINT32
7	PDO_mapping_entry_7	UINT32
8	PDO_mapping_entry_8	UINT32

The PDO mapping parameter contains information about the content of PDO.

0x1600.0 *Number_of_entries*

Data type	Acces	Default	Range	Unit
UINT8	RW	1	[0 - 8]	-

The number of valid object entries within the mapping record. If it is equal to 0, the mapping is disabled.

0x1600.1 *PDO_mapping_entry_1*

Data type	Acces	Default	Range	Unit
UINT32	RW	1614807056	-	-

The information of mapped application objects. The object describes content of PDO by index, sub-index and length of the mapped object.

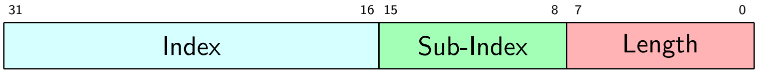


Figure 3.26: Structure of PDO mapping

Note: The MSB is first.

Note: The length is the length of object in bits.

8.4.3 0x1800 *TPDO_parameter_1* 0x1801 *TPDO_parameter_2* 0x1802 *TPDO_parameter_3* 0x1803 *TPDO_parameter_4*

Index	Name	Object type
0x1800	TPDO_parameter_1	RECORD

Subindex	Name	Data type
1	COB_ID	UINT32
2	Transmission_type	UINT8
3	Inhibit_time	UINT16
5	Event_timer	UINT16
6	SYNC_start_value	UINT8

The PDO parameter describes the communication abilities of the PDO.

0x1800.1 *COB_ID*

Data type	Acces	Default	Range	Unit
UINT32	RW	384	[0x00000080 - 0xFFFFFFFF]	-

Contains the CAN-ID (COB-ID) of the TPDO

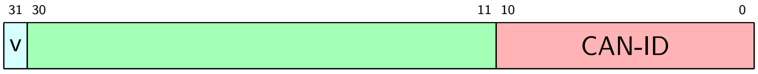


Table 3.17: COB ID

- **v**: The bit **valid** select which TPDOs are used in the NMT state Operational. PDOs can be fully configured but not used, and therefore set to "not valid" .

Value	Description
0x00	PDO exists / is valid
0x01	PDO does not exist / is not valid

Table 3.18: Description of bit 31: v

- **COB ID CAN-ID** of TPDO

0x1800.2 ***Transmission_type***

Data type	Acces	Default	Range	Unit
UINT8	RW	1	-	-

The transmission of the PDO depends on the configuration of Transmission Types parameters which can be:

Value	Description
0x00	Synchronous (acyclic)
0x01 to 0xF0	Synchronous cyclic every N SYNC
0xFC	RTR-only (synchronous)
0xFD	RTR-only (event-driven)
0xFE	Event-driven

Table 3.19: Description of TPDO transmission type

- **Synchronous acyclic**: On internal event, the sampling will start and will transmit after the next SYNC.
- **Synchronous**: the TPDO is transmitted after each SYNC received. The sampling of the data will start and will transmit on reception of each the SYNC received.
- **RTR-only (synchronous)**: On RTR (Remote Transmission Request) received, the sampling will start and will transmit after the next SYNC.
- **RTR-only (event-driven)**: On RTR (Remote Transmission Request) received, the sampling start and transmit immediately.
- **Event-driven**: sampling may be transmitted at any time when sampling is different of previous.

0x1800.3 ***Inhibit_time***

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	0.1 ms

It's a minimum interval time for PDO transmission. This parameter is available only for the transmission type 0xFE. The value is defined as multiple of 100 μs. The value of 0 shall disable the inhibit time. The value shall not be changed while the PDO exists.

0x1800.5 ***Event_timer***

Data type	Acces	Default	Range	Unit
UINT16	RW	0	-	ms

TPDO frame sending timer. Independent from SYNC. The value is defined as multiple of 1 ms. The value of 0 shall disable the event-timer.

0x1800.6 ***SYNC_start_value***

Data type	Acces	Default	Range	Unit
UINT8	RW	0	-	-

The SYNC message of which the counter value equals the SYNC Start value is be regarded as the first received SYNC message. The value of 0 shall disable the SYNC Start value.

8.4.4 0x1A00 ***TPDO_mapping_1***
0x1A01 ***TPDO_mapping_2***
0x1A02 ***TPDO_mapping_3***
0x1A03 ***TPDO_mapping_4***

Index	Name	Object type
0x1A00	TPDO_mapping_1	RECORD

Subindex	Name	Data type
0	Number_of_entries	UINT8
1	PDO_mapping_entry_1	UINT32
2	PDO_mapping_entry_2	UINT32
3	PDO_mapping_entry_3	UINT32
4	PDO_mapping_entry_4	UINT32
5	PDO_mapping_entry_5	UINT32
6	PDO_mapping_entry_6	UINT32
7	PDO_mapping_entry_7	UINT32
8	PDO_mapping_entry_8	UINT32

The PDO mapping parameter contains information about the content of PDO.

`0x1A00.0` ***Number_of_entries***

Data type	Acces	Default	Range	Unit
UINT8	RW	1	[0 - 8]	-

The number of valid object entries within the mapping record. If it is equal to 0, the mapping is disabled

`0x1A00.1` ***PDO_mapping_entry_1***

Data type	Acces	Default	Range	Unit
UINT32	RW	1614872592	-	-

The information of mapped application objects. The object describes content of PDO by the index, sub-index and length of the mapped object.

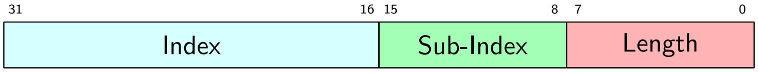


Figure 3.27: Structure of PDO mapping

Note: The MSB is first.

Note: The length is the length of object in bits.

9 Object description

9.1 Communication Profile Area object

9.1.1 `0x1000` Device_type

Data type	Acces	Default	Range	Unit
UINT32	RO	402	-	-

This object provides information about the device type. It is composed of two fields, the device profile and additional information, both on 16 bits.

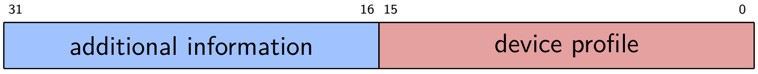


Table 3.20: Frame of Device Type

Value	Device profile	Description
0x191	CiA 401 profile	Inputs / outputs
0x192	CiA 402 profile	Motion control

Table 3.21: Description of device type

9.1.2 0x1008 Manufacturer_device_name

Data type	Acces	Default	Range	Unit
VSTRING	RO	UMC1BDS32 motion controller	-	-

Provides the name of device.

9.1.3 0x1009 Manufacturer_hardware_version

Data type	Acces	Default	Range	Unit
USTRING	RO	vXX.XX.XXX	-	-

Provides the manufacturer hardware version description.

9.1.4 0x100A Manufacturer_software_version

Data type	Acces	Default	Range	Unit
VSTRING	RO	-	-	-

Provides the manufacturer software version description.

9.1.5 0x100C Guard_time

Data type	Acces	Default	Range	Unit
UINT16	RW	0	[0 - 65535]	ms

The life time factor multiplied with the guard time gives the life time for the life guarding protocol. The value of 0x00 disable the life guarding.

9.1.6 0x100D Life_time_factor

Data type	Acces	Default	Range	Unit
UINT8	RW	0	[0 - 255]	-

The life time factor multiplied with the guard time gives the life time for the life guarding protocol. The value of 0x00 disable the life guarding.

9.1.7 0x1017 Producer_heartbeat_time

Data type	Acces	Default	Range	Unit
UINT16	RW	0	[0 - 65535]	ms

The producer heartbeat time indicate the configured cycle time of heartbeat.

9.1.8 0x1010 Store_parameters

Index	Name		Object type
0x1010	Store_parameters		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT32	RW	-	-

Subindex	Name
1	Save_all_parameters
2	Save_communication_parameters
3	Save_standardized_parameters
4	Save_manufacturer_parameters

This object stores the value of the parameters according to the communication profile, manufacturer profile and standardized profile or all profiles. The profile areas are defined to Object Dictionary (OD).

This functionality can only work in state PreOp if this is not the case, the device respond with the SDO abort transfer service (SDO abort code: 0x08000021).

Signature To start backup of parameters, a specific signature is required to avoiding wrong manipulation. Specific signature is written in appropriate subindex.

The signature is "save":

3	2	1	0
e	v	a	s
0x65	0x76	0x61	0x73

Figure 3.28: Signature of store

Upon receipt of the correct signature in the appropriate subindex, the device restores the default settings and then confirms the SDO transmission (SDO download initiation response).

If an erroneous signature is written, the device refuses to store the defaults and responds with the SDO abort transfer service (SDO abort code: 0x08000020).

9.1.9 0x1011 Restore_default_parameters

Index	Name		Object type
0x1011	Restore_default_parameters		ARRAY
Data type	**Acces**	**Range**	**Unit**
UINT32	RW	-	-

Subindex	Name
1	Restore_all_factory_parameters
2	Restore_factory_communication_parameters
3	Restore_factory_standardized_parameters
4	Restore_factory_manufacturer_parameters
5	Restore_all_saved_parameters
6	Restore_saved_communication_parameters
7	Restore_saved_standardized_parameters
8	Restore_saved_manufacturer_parameters

This object restores the factory or saved values of the parameters according to the communication profile, manufacturer profile and standardized profile or all profiles. Profile areas are defined to Object Dictionary (OD).

This functionality can only work in State PreOp if this is not the case, the device respond with the SDO abort transfer service (SDO abort code: 0x08000021).

Signature To start the restoration of parameters, a specific signature is required to avoid wrong manipulation. Specific signature is written in appropriate subindex.

The signature is "load":

3	2	1	0
d	a	o	l
0x64	0x61	0x6F	0x6C

Figure 3.29: Signature of restore

Upon receipt of the correct signature in the appropriate subindex, the device restore the default settings and then confirm the SDO transmission (SDO download initiation response).

If an erroneous signature is written, the device refuse to restore the defaults and respond with the SDO abort transfer service (SDO abort code: 0x08000020).

Automatic restore This feature determines an automatic restore after a NMT service reset node, NMT service reset communication or power cycled. This functionality is configured with a command (table below) written in the appropriate subindex.

Value	Description
0x00000000	Device does not restore settings automatically
0x00000001	Device restores settings automatically

Table 3.22: Automatic restore values

9.1.10 `0x1018` Identity_object

Index	Name	Object type
0x1018	Identity_object	RECORD
Subindex	**Name**	**Data type**
1	Vendor_ID	UINT32
2	Product_code	UINT32
3	Revision_number	UINT32
4	Serial_number	UINT32

Provide general identification information of the CANOpen device.

`0x1018.1` *Vendor_ID*

Data type	Acces	Default	Range	Unit
UINT32	RO	1186	-	-

Unique vendor value of a CANOpen device.

`0x1018.2` *Product_code*

Data type	Acces	Default	Range	Unit
UINT32	RO	4097	-	-

The unique value that identifies a specific type of CANOpen devices.

`0x1018.3` *Revision_number*

Data type	Acces	Default	Range	Unit
UINT32	RO	16777219	-	-

The major revision number and the minor revision number of the revision of the CANOpen device.

31 16	15 0
Major revision number	Minor revision number

Table 3.23: Revision number

`0x1018.4` *Serial_number*

Data type	Acces	Default	Range	Unit
UINT32	RO	0	-	-

The serial number identify a CANOpen device within a product group and a specific revision.

Appendix A

Firmware version history

Version	Date	Change
1.0.0	2020/12/14	Initial public version
1.0.1	2021/01/21	Added support for UMC1BDS32 v1.2.0
1.0.2	2022/03/09	Added auto detection of board version Improved BLDC speed estimator from hall sensors
1.0.3	2022/08/31	Improved global performance Improved BLDC maximal speed Reduced minimal BLDC measurable speed Fixed over current detection Added brake timings Added fault diagnostic objects Fixed SSI position sensor mode
1.0.4	2023/06/01	Updated compiler to a major release Updated SDK to a major release Added Digital Inputs read object Adding limits switch

Appendix B

Datasheet revision history

Revision	Date	Change
A	2020/03/20	Initial public revision
B	2021/11/08	Reviewed structure of document
C	2022/08/31	Reviewed CANOpen objects tables Added firmware revision 1.0.2 and 1.0.3 Added description of all manufacturer objects
D	2023/06/01	Adding missing objects Added firmware revision 1.0.4